

要求仕様に対する形態素ベースレビューの提案

柏原 一雄
株式会社デンソークリエイト
kazuo.kashiwabara.j3a@jpgr.denso.com

鈴木 淳
株式会社デンソークリエイト
atsushi.suzuki.j3h@jpgr.denso.com

新留 光治
株式会社デンソークリエイト
mitsuharu.shintome.j8u@jpgr.denso.com

松原 潤弥
株式会社デンソークリエイト
junya.matsubara.j4v@jpgr.denso.com

李 路雲
株式会社デンソークリエイト
uyun.li.j4z@jpgr.denso.com

不破 慎之介
株式会社デンソークリエイト
shinnosuke.fuwa.j8h@jpgr.denso.com

要旨

要求仕様が自然言語のような曖昧さを含む記法で記述されている場合、曖昧に表現された文が不具合を誘発させることがある。実際に我々の組織では、「修飾語」または「同義語」を使用している文が不具合を誘発していた。要求仕様書から修飾語・同義語が使用されている曖昧な文を検出するために、「形態素ベースレビュー」を考案した。実開発に形態素ベースレビューを導入し、効果を確認した。

1. はじめに

ソフトウェア開発において、仕様は関係者間の情報ハブ^[1]となり、仕様定義者は多数の関係者と仕様によりコミュニケーションをする。文献[1]では、形式的な品質の高い仕様は、定義内容の検索や変換、そして抽出や機械的な検証を行うことが容易になるため、プロジェクトのさまざまな局面での活用が可能になると述べられている。仕様の品質を向上するためには、曖昧さのない文を書くことが大切になる。文献[2]では、文の曖昧さは、表 1 に示すように多義性・修飾性・類義性の 3 つに分類されると述べられている。

実際に、我々の組織では、要求仕様の修飾性と類義性により、不具合が引き起こされた事例があった。具体的には、「修飾語」または「同義語」を使用していた文が、不具合を誘発していた。要求仕様に対するレビューを実施していても、修飾性と類義性の問題を見逃していた。

表 1 文の曖昧さの分類

文の曖昧さの分類	説明
多義性	一つの単語で異なる意味を表現する。
修飾性	修飾関係が一意に決まらない。
類義性	異なる単語で同じ意味を表現する。

本研究では、欠陥混入のリスクを低減することを目的とし、自然言語で記述された要求仕様書から、修飾語・同義語が使用されている曖昧な文を検出する手法を考案した。曖昧表現^[3]すべてを解決の対象とするのではなく、実際に不具合に繋がった曖昧表現のみを解決の対象とする。また、考案手法は、稼働中のプロジェクトにも少ないコストで導入できるようにした。

考案手法は、「形態素ベースレビュー」と呼ぶ。形態素ベースレビューは、要求仕様に対して形態素解析を実施した結果を入力とするリーディング技法^[4]である。

実験では、実開発で作成された要求仕様書を対象に形態素ベースレビューを実施した。評価の結果、形態素ベースレビューが以下の要求を満たすことを確認できた。

- ・ アドホックレビューでは見逃した曖昧な文を検出できる
- ・ 不具合を誘発した曖昧な文を検出できる
- ・ 稼働中のプロジェクトに対して導入しやすい

実開発において、形態素ベースレビューを活用することで、要求仕様の表現を起因とした不具合を減らす効果が期待できる。

これ以降の本稿の構成は次のとおりである。2章で現状分析の結果と研究の課題を示す。3章では、課題解決の参考とした先行研究を示す。4章では、考案した解決策を提案する。5章では、提案手法の評価結果と考察を示す。6章では、まとめと今後の進め方を示す。

2. 課題設定

2.1. 現状分析

(1) 不具合を誘発した曖昧な文

我々の組織では、要求仕様の表現を起因とした不具合が発生していた。不具合の事例を分析すると、不具合を誘発した曖昧な文では、「抽象的な語」「修飾語」「同義語」を使用していた。この3種類の単語が使われている文で発生する問題を表2に示す。

表2 不具合を誘発した単語の種類

単語の種類	関連する曖昧性	発生する問題
抽象的な語	多義性	仕様の誤解釈
修飾語	修飾性	仕様の誤解釈
同義語	類義性	検索での関連仕様の抽出漏れ

「抽象的な語」が誘発する問題に対しては、文献[5]で解決策の一つとなる手法「Domain Word Modeling」が提案されている。Domain Word Modelingは、“抽象的に表現された用語”を“不足している前提知識”で誤解釈することにより発生していた手戻りを防止する。

本研究では、「修飾語」または「同義語」を使用している文が誘発する不具合を防止することを目的とする。

(2) 修飾語

修飾語を使用している文は、仕様の誤解釈を誘発しやすい。修飾語を付けて対象とすること・ものを説明している場合に、その絞り込んでいる対象範囲を誤解釈しやすい。例えば、以下のような文が問題を引き起こした。

例:

- “使用可能な”モード以外には遷移しない。
- “不要となる”処理を実行しないように変更する。
- 初期化が“必要な”データを更新する。

上記の例の場合、誤解釈を防止するために、「“使用可能な”モード」、「“不要となる”処理」「初期化が“必要な”データ」が、他の文で具体的に定義されている必要がある。

修飾語となる品詞^[6]には、形容詞・副詞・形容動詞・副詞可能等がある。これらの品詞が、誤解釈を引き起こしやすいと考えた。

(3) 同義語

同義語^{[7][8]}を使用している文は、検索で関連する仕様の抽出漏れを誘発する。例えば、以下のような同義語が問題を引き起こした。

例:

- OffCounter / OFF カウンタ / 消去判定カウンタ
- 合計額 / 合計金額 / 投入金額

“ドメイン用語^[5]であり”かつ“表記ゆれでない”同義語が、特に問題を誘発しやすいことがわかった。ドメイン用語は、一般的な用語に比べて検索対象となることが多く、問題を引き起こす可能性が高い。また、文章作成ソフト等の表記ゆれチェック機能では検出できない同義語は見逃しやすい。例えば、「売切ランプ」と「売り切れランプ」のような似ている単語は、ツールで表記ゆれとして検出できるが、例のような似ていない単語は表記ゆれとして検出できない。表記ゆれでない単語は、ツールでは同義語として検出しにくく、問題を引き起こす可能性が高い。

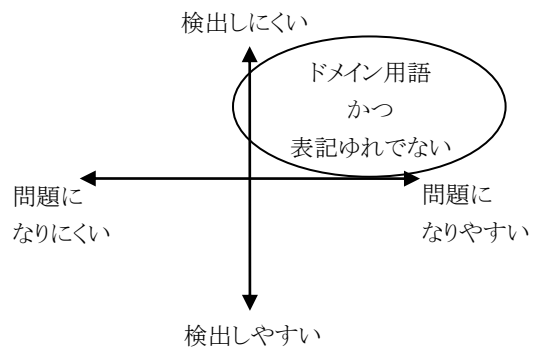


図1 問題を誘発しやすい同義語

(4) 要求仕様のレビュー

我々の組織で実施されている要求仕様に対するアドホックレビューでは、上から順に文を読む。このとき、文脈と短期記憶に依存して読むため、曖昧な文の検出漏れが起きやすい。

文脈に依存して読むため、レビューで修飾語を使用している文の曖昧さを指摘することが難しい。文献[9]で、「わかった状態」になるメカニズムが説明されている。「人にとって「わかった」状態は安定状態であり、「わかった」状態になりたがる。人は、文章を読むとき、「わかった」状態にするために、文脈を使う」と説明されている。つまり、人は文脈をもとに読むと、厳密な記述でなくても、「わかった」状態になってしまう。「わかった」ことを「わからないかもしれない」と、レビューで指摘することは難しい。更に、書き手と読み手がそれぞれ使用している文脈はレビュー対象の文書には表現されない。表現されておらず見えない書き手と読み手の文脈の違いを、レビューで指摘することは難しい。

短期記憶に依存して読むため、レビューで同義語を使用している文を検出することが難しい。同義語を抽出するためには、他の文で使用されている単語を記憶しておく必要がある。しかし、人は短期的には 7 ± 2 程度のものしか覚えられないと考えられている^[10]。そのため、レビュー対象の文の量が多くなればなるほど、単語を網羅的に比較し、同義語を抽出することは難しくなる。

「修飾語」または「同義語」を使用している曖昧な文を検出するためには、「上から順に文を読む」ではないリーディング技法が必要となる。

2.2. 課題提起

本研究では、要求仕様書から、「修飾語」「同義語」が使用されている曖昧な文を検出する手法を考案することを課題とする。

考案手法は、不具合を防ぐため、以下の問題を防止することを目的とする。

- ・ 修飾語の使用箇所で起きる仕様の誤解釈
- ・ 同義語の使用箇所で起きる検索での関連仕様の抽出漏れ

考案する手法は、早く実開発に導入し、確実に効果を得るために、以下の3つの要求を満たすものとする。

- ・ “上から順に文を読む”アドホックレビューで見逃した要求仕様の曖昧な文を検出できる

- ・ 実際に不具合を誘発した要求仕様の曖昧な文を検出できる
- ・ 要求仕様が自然言語で定義されている稼働中のプロジェクトに対して導入しやすい

考案手法を導入しやすくするために、「成果物を追加・変更するコスト」「手法を習得するコスト」「ツールを用意するコスト」を下げる。「成果物を追加・変更するコスト」を下げるため、既に存在している自然言語で記述されている要求仕様書のみを入力に実行できる手法とする。「手法を習得するコスト」を下げるため、考案手法を未経験の状態からでも、手順の説明を受けただけで実行できる手法とする。「ツールを用意するコスト」を下げるため、フリーソフトウェアで実行できる手法とする。

本研究では、無駄に曖昧表現を検出・修正する工数を増やさないために、すべての曖昧表現を解決の対象とするのではなく、実際に不具合につながった曖昧表現のみを解決の対象とした。また、通常の“上から順に文を読む”リーディング技法では検出が難しい曖昧表現のみを解決の対象とした。

3. 先行研究

3.1. キーワードベースドレビュー

文献[11]で、ドキュメントのあいまいさや不備を検出する手法として、キーワードベースドレビューが提案されている。キーワードベースドレビューは、ドキュメントのあいまいさや不備につながるキーワードを整理し、キーワードが使われている箇所を確認するリーディング技法である。キーワードは観点毎に整理されている。観点とキーワードの例の一部を表 3に示す。

表 3 観点とキーワードの例(文献[11]から引用)

観点	具体的記述の例
数値的表現	から, ~, 以上, 以下
時間的表現	あとで, 先に, 事前に, 後程, 常に, いつも, 常時, 逐次, 一定
条件指定表現	とき, でないとき, 以外, 以内, の際, ではないとき, している(する)最中
不明瞭な表現	ぐらい, とか, みたいな, 的には, おそらく, ほぼ, ような, できるだけ
形容表現	多, 少, 速, 遅
指示表現	これ, あれ, それ, どれ, その, この

キーワードベースレビューは、曖昧な文となり得る修飾語を使用している文を検出するために、有効な手法と考えられる。ただし、キーワードベースレビューを実施するためには、検索するキーワードを列挙したキーワードリストを、事前に作成する必要がある。

考案手法では、キーワードではなく品詞をもとに、修飾語を使用している曖昧な文を検出する方針とする。観点が同じキーワードには、同じ品詞を関連づけられる可能性がある。例えば、時間的表現のキーワードの多くは「副詞」、形容表現のキーワードは「形容詞」というように、キーワードに品詞を関連づけることができる。あいまいさにつながるキーワードで検索するのではなく、あいまいさにつながる品詞で検索をすることができれば、キーワードを列挙したキーワードリストを作成する必要はなくなると考えた。

3.2. 省略と修飾パターンを用いた用語不一致検証

文献[12]では、要求仕様の「一貫性」に着目し、要求仕様の一貫性検証支援ツールを実現している。ツールには、検証ルールと辞書が組み込まれている。仕様書内で設計要素の用語名が一貫しない状態を、省略パターン、修飾パターン、表記ゆれの3つに分類し、それらを検証するための知識が、用語不一致検証ルールとして定義されている。3つの用語不一致となる概念の説明を表4に示す。また、用語不一致検証を行うために、設計要素別の識別辞書・表記ゆれ辞書を用意している。

表4 用語不一致となる概念(文献[12]から引用)

概念	説明
省略	要求仕様書中に定義された設計要素から、形容詞や名詞を省略して、要求仕様書中で利用している場合に、用語不一致とみなす。
修飾	要求仕様書中に定義された設計要素に、形容詞や名詞を修飾して、要求仕様書中で利用している場合に、用語不一致とみなす。
表記ゆれ辞書一致	あらかじめ定義した表記ゆれ辞書に定義された表記ゆれ関係にある用語が、要求仕様書中に利用されている場合に、用語不一致とみなす。

検証ルールと辞書を利用した用語不一致検証は、同義語を抽出するために、有効な手法と考えられる。ただし、検証を実施するためには、ツールに組み込む辞書を、事前に作成する必要がある。また、ツールや辞書の制約により、ツールによる検証結果には、不要な指摘が含まれることがある。そのため、ツールでは用語不一致の候補を抽出し、用語不一致の最終的な判断は人手で行う必要がある。

考案手法では、用語不一致のパターンを活用し、人手で、同義語を抽出する方針とする。省略パターン・修飾パターンという用語不一致の状態を検出する方法は、ツールを使用せずに、人手で同義語の候補を抽出するときにも活用できると考えた。単語の比較をするときに、ドメイン知識が不足していても、省略パターン・修飾パターンに当てはまるかという判定はしやすい。

3.3. 形態素解析とKH Coder

形態素解析^[13]は、自然言語処理の基礎となる要素技術である。形態素解析では、文の分かち書きを行い、単語列を同定し、個々の単語の品詞の決定などを行う。

KH Coder^{[14][15]}は、定量テキスト分析のためのフリーソフトウェアである。内部では形態素解析器である茶筌を利用している。非常に手軽なマウス操作によってテキスト型データの分析が行えるという特徴がある。

KH Coderでは、テキストファイルを入力に形態素解析器で、図3と図2に示すように抽出語のリストと複合語のリストを作成することができる。

形容動詞	副詞可能	未知語	形容詞	副詞	副詞B		
可能	22 場合	24 アクター	25 無い	2 再び	1 かつ		1
異常	2 時間	5 払	7 若い	1 次に	1 また		1
正当	2 結果	3 センサ	1 新しい	1 逐次	1 また		1
不当	2 前	2	大きい	1 同時に	1		
適切	1 すべて	1					
適切	1 一番	1					
	今	1					
	全て	1					
	途中	1					

図3 抽出語のリスト

複合語	出現数
販売ボタン	24
当該商品	14
販売可能	11
自動販売機	9
代金投入	8
合計金額	7
返金処理	7

図2 複合語のリスト

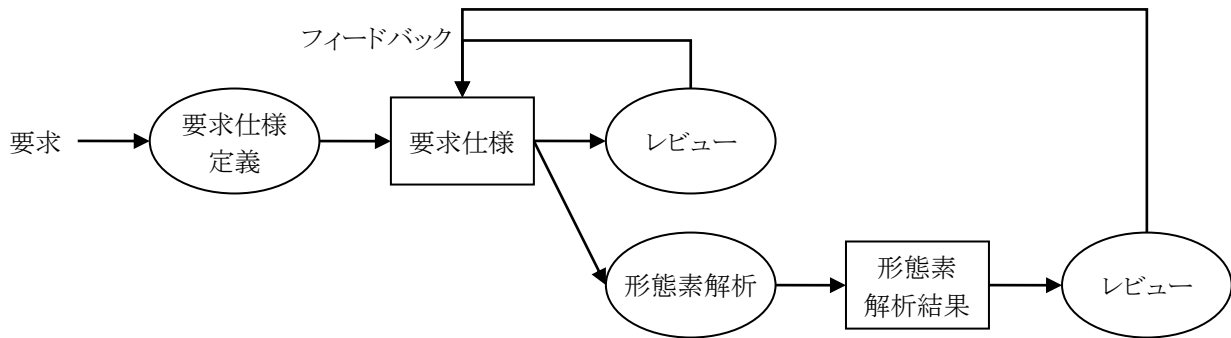


図 4 形態素ベースレビューの概要

抽出語のリストから、修飾語となり得る品詞の単語を、漏れなく抽出できると考えた。更に、抽出語のリストと複合語のリストから、同義語の判定のために比較をする対象の単語を、漏れなく抽出できると考えた。また、KH Coder は、フリーソフトウェアであり手軽に操作できる特徴により、導入コストが低い。

4. 解決策の提案

4.1. 課題の解決方針

考案手法は、導入コストを下げるために、事前にキーワードリストや辞書を作成しない方針とする。曖昧な文の事例が多数収集できていないと、有効なキーワードリストを作成することは困難である。要求仕様を定義する前の使用する単語が明確になっていない段階で、有効な辞書を作成することは困難である。

考案手法では、修飾語・同義語が使用されている文を抽出するために、形態素解析を活用する。形態素解析器で、品詞毎に単語を抽出し、修飾語となり得る品詞である単語を特定しやすくする。また、形態素解析器で、単語一覧を作成し、同義語を抽出するために実施する単語の比較をしやすくする。

考案手法は、要求仕様の文ではなく、形態素解析器により抽出された単語をもとにレビューをすることで、以下の特徴をもつ手法となる。

- ・ 文脈の力を発動させず、文の曖昧さに気づきやすくする
- ・ 短期記憶に頼らず、単語の比較をすることを可能にする

考案手法は、形態素ベースレビューと呼ぶ。図 4 に形態素ベースレビューの概要を示す。アドホックレビュ

ーに加えて、形態素ベースレビューを実施することで、アドホックレビューでは見逃しやすい曖昧な文を検出することを可能とする。

本研究では、形態素解析を実施するために、KH Coder を使用した。本稿では、形態素ベースレビューの手順を KH Coder を使用することを前提として定義する。

4.2. 修飾語の抽出の方法

考案手法では、修飾語は品詞をもとに抽出する。修飾語となり得る品詞は、KH Coder で特定可能な形容詞・副詞・形容動詞・副詞可能の 4 種類とする。この 4 種類の品詞に分類される単語を修飾語と判定する。この品詞に分類された単語をキーワードとして、要求仕様書を検索し、単語の使用箇所を特定する。表 5 に、品詞をもとに抽出される修飾語の例を示す。この例に示した単語は、テスト設計コンテスト'14 の ASTER 自動販売機ユースケース仕様書を対象に抽出したものの一部である。

表 5 品詞をもとに抽出される修飾語の例

品詞	単語
形容詞	無い, 若い, 新しい, 大きい
副詞	再び, 次に, 逐次, 同時に
形容動詞	可能, 異常, 適切
副詞可能	場合, 結果, 前, すべて, 途中

4.3. 同義語の抽出の方法

考案手法では、人手により同義語の抽出を行うため、効率化と抽出漏れ防止の対策が必要となる。

同義語の抽出を効率化するために、比較対象の単語の絞り込みを行う。同義語を抽出するために比較対象と

する単語は、検索で使用する可能性の高い「ドメイン用語」に限定する方針とする。ドメイン用語である可能性の高い単語は、形態素解析器で入力としている標準の辞書に登録されていない単語であると考え、KH Coderでは、辞書に登録されていない単語は、「未知語」「複合語」として抽出可能である。同義語を抽出するため比較する単語は、未知語・複合語とする。

同義語の抽出の漏れを防ぐために、同義語を抽出するときの3つのポイントを以下に示す。

- ・ 出現頻度の少ない単語を確認する
同義語は、出現頻度の少ない単語のほうが、修正すべき単語の可能性が高い。誤って他と異なる表現を使ってしまった単語は、出現頻度が低くなる。誤記の場合にも、出現頻度が低くなる。出現頻度が高い単語は、表現が統一されているということであり、修正対象とはならない。
- ・ ソートして比較する
未知語・複合語をソートすることで、似ている単語が並ぶことになり、比較しやすくなる。例えば、「合計額」と「合計金額」が並ぶことで、同義語を抽出しやすくなる。
- ・ 用語不一致パターンに当てはまるか判定する
複合語に対して、省略パターンまたは修飾パターンの用語不一致になっていないか確認する。複合語の一部の語を省略した単語が使用されていないかを確認する。複合語の末尾に同じ名詞が使われている単語を比較対象とする。例えば、「懸賞あたりランプ」という複合語があった場合、「あたりランプ」「ランプ」という単語が使用されていないかを確認する。

4.4. 形態素ベースドレビューの手順

形態素ベースドレビューは、「1.形態素解析」「2.修飾語の抽出」「3.同義語の抽出」「4.曖昧な文の検出」の4ステップで実施する。各ステップで実施することを次に示す。

1. 形態素解析
要求仕様を入力に、形態素解析を実施し、抽出語のリストと複合語のリストを作成する。
2. 修飾語の抽出
抽出語のリストから形容詞・副詞・形容動詞・副詞可能に分類される単語を抽出する。
3. 同義語の抽出
抽出語のリストから比較対象の未知語を抽出

する。複合語のリストから比較対象の複合語を抽出する。未知語・複合語として抽出した単語を比較し、同義語であるか判定する。

4. 曖昧な文の検出

ステップ2、ステップ3で抽出した修飾語・同義語の使用箇所を要求仕様から特定し、修正要否を判断する。

5. 解決策の評価

5.1. 評価方法

考案手法「形態素ベースドレビュー」を以下の3つの観点で評価する。

- ・ アドホックレビューでは見逃した曖昧な文を検出できるか
- ・ 不具合を誘発した曖昧な文を検出できるか
- ・ 稼働中のプロジェクトに対して導入しやすいか

考案手法を評価するために、以下の3種類の実験を行う。

- A) 複数の異なる要求仕様書に対して実行
- B) 不具合を誘発した曖昧な文に対して実行
- C) 考案手法の未経験者が実行

3つの実験の方法を以下に示す。形態素解析を実施するために使用したツールは、KH Coder 3である。選択した形態素解析器は茶筌である。

- A) 複数の異なる要求仕様書に対して実行
アドホックレビュー済みの要求仕様書を対象として、考案手法を実行し、修飾語・同義語を使用している曖昧な文が検出できるかを確認する。表6に計測するデータを示す。

表 6 計測データの種類(実験 A)

データ	説明
仕様定義者数	仕様定義者の数。 複数人か一人を示す。
総抽出語数	抽出語のリストに出力される各抽出語の出現頻度の合計。
修飾語数	考案手法で検出した曖昧な文で使用していた修飾語の数。
同義語数	考案手法で検出した曖昧な文で使用していた同義語の数。

確認対象は、4 つの異なる案件の要求仕様書とする。形態素ベースレビューの実施者は、仕様定義者自身とする。

- B) 不具合を誘発した曖昧な文に対して実行
不具合の原因となった曖昧な文を含む要求仕様書を対象として、考案手法を実行し、不具合の原因となった箇所を検出できるかを確認する。
2 つの不具合事例を対象に確認をする。同義語により不具合を誘発した要求仕様書は用意できなかったため、事例はどちらも修飾語により不具合を誘発した要求仕様書である。

- C) 考案手法の未経験者が実行
形態素ベースレビューを未経験の状態から、手順の説明を受けるだけで、実行できるかを確認する。形態素解析は事前に実施しておき、抽出語のリストと複合語のリストを入力に、「2.修飾語の抽出」「3.同義語の抽出」「4.曖昧な文の検出」ができるかを確認する。表 7 に計測するデータを示す。

4 人の被験者を対象に確認をする。1 名のみ形態素ベースレビューの経験者、残りの 3 名は未経験者とし、結果を比較する。全員、同じ要求仕様書を対象にレビューを実施する。「ASTER 自動販売機ユースケース仕様書」をレビュー対象の要求仕様書とする。

表 7 計測データの種類(実験 C)

データ	説明
レビュー時間	考案手法の実施時間。形態素解析の作業時間は除く。
修飾語数	考案手法で検出した曖昧な文で使用していた修飾語の数。
同義語数	考案手法で検出した曖昧な文で使用していた同義語の数。

5.2. 評価結果

各実験の結果を以下に示す。

- A) 複数の異なる要求仕様書に対して実行
表 8 に、4 つの要求仕様書毎に、形態素ベースレビューの実行結果を示す。

すべての要求仕様書で、修飾語または同義語を使用している曖昧な文を検出した。検出した問題は、すべてアドホックレビューでは見逃したものであった。仕様定義者や形態素数が増えるほど、検出される問題が増える傾向があった。曖昧な文で使用していた修飾語は、すべて「形容動詞」であった。

表 8 形態素ベースレビューの結果(仕様毎)

ID	仕様定義者数	総抽出語数	修飾語数	同義語数
01	複数人	5,611	5	10
02	一人	1,447	2	0
03	一人	490	2	0
04	一人	237	1	0

- B) 不具合を誘発した曖昧な文に対して実行
2 つの不具合事例で、不具合の原因となった曖昧な文を検出することができた。不具合を誘発した修飾語は、どちらも「形容動詞」であった。

- C) 考案手法の未経験者が実行
表 9 に、被験者毎に、形態素ベースレビューの実行結果を示す。入力とした要求仕様書の総抽出語数は、1,036 である。また、表 10 に実験に参加した被験者の特徴を示す。

被験者全員が、手順の説明を 1 時間程度受けるだけで、考案手法を実行できた。考案手法の未経験者であっても、全員が 1 時間のレビューで、曖昧な文を検出することができた。ただし、検出した問題については差がある。考案手法の経験者が検出した問題の一部を考案手法の未経験者は見逃していた。要求仕様のレビュー経験がない場合、修飾語を使用している曖昧な文を検出できない傾向があった。

表 9 形態素ベースレビューの結果(人毎)

ID	レビュー時間(H)	修飾語数	同義語数
ア	0.5	1	4
イ	1.0	1	2
ウ	1.0	0	6
エ	1.0	0	2

表 10 被験者の特徴

ID	考案手法の 経験有無	要求仕様 レビューの 経験有無	ソフト開発の 経験年数
ア	あり	あり	10年以上
イ	なし	あり	2年
ウ	なし	なし	1年
エ	なし	なし	1年

実験結果をもとに、5.1評価方法に示した3つの観点で考案手法を評価する。評価結果を以下に示す。

- ・ 実験Aの結果から、アドホックレビューでは見逃した曖昧な文を考案手法で検出できることが確認できた。
- ・ 実験Bの結果から、実際に不具合を誘発した曖昧な文を考案手法で検出できることが確認できた。
- ・ 実験Aの結果から、既存の要求仕様書を入力に、成果物の追加・変更をせずに、考案手法が実行できることが確認できた。また、実験Cの結果から、考案手法を未経験の状態からでも、手順の説明を受けただけで実行できることが確認できた。

この結果から、形態素ベースレビューは、2.2課題提起で示した3つの要求を満たす手法であると考えられる。ただし、形態素ベースレビューでも、他のリーディング技法と同様に、検出できる問題はレビューアの能力に依存する。形態素ベースレビューのレビューアには、要求仕様のレビューの経験が必要となる。要求仕様のレビューの経験が不足している場合、曖昧な文の判断の効率が落ちたり、曖昧な文を見逃したりする可能性が高まる。

5.3. 結果の考察

形態素ベースレビューは、文ではなく単語を確認する。これにより、アドホックレビューより、修飾語・同義語を使用している曖昧な文を検出しやすくなった。これは、形態素ベースレビューがもつ以下の2つの特徴が要因と考えられる。

- ・ 文脈の力を発動させず、文の曖昧さに気づきやすくなる
- ・ 短期記憶に頼らず、単語の比較をすることが可能である

形態素ベースレビューは、導入コストが低いため、稼働中のプロジェクトでも導入できた。キーワードリストや辞

書を事前に用意せず実行できる点が、導入コストを下げられた大きな要因である。

また、実験の結果、考案手法の改善が必要な点を2点明らかにした。1つ目は「修飾語の抽出手順」、2つ目は「同義語の抽出手順」に関する改善点である。

修飾語を使用している曖昧な文を検出するための手順は改善の余地がある。具体的には、不具合につながる可能性のない文を無駄に検出することがないように、確認対象の品詞の見直しが必要である。形態素ベースレビューで問題ありと判断された文で使用していた修飾語の品詞は、形容動詞のみであった。また、不具合の原因となっていた文で使用していた修飾語の品詞も、形容動詞のみであった。実験で入力とした要求仕様書において、形容動詞の単語は、名詞+「な」の形で表現され、登場することが多かった。定性的な語^[3]である形容詞や副詞等は、考案手法を使わずに、一つの文を普通に読むだけで、曖昧と判断できている可能性が高い。

同義語の抽出には時間がかかるため、手順の改善を検討する必要がある。同義語を抽出する工数を削減することで、手法の導入のしやすさを更に向上させることができる。要求仕様書で使用している単語が多くなればなるほど、単語を比較する回数が増え、同義語を抽出する工数が増える。同義語の抽出は、すべての要求仕様の定義完了後にまとめて1回実施のではなく、要求仕様の定義途中に繰り返し複数回実施するほうが、工数が削減できる可能性がある。また、評価結果から、“仕様定義者が一人”かつ“使用している単語が少ない”という条件では、同義語が使用される可能性は低いことがわかった。このような条件を満たす場合は、同義語のチェックを省略しても不具合にはつながらず、チェックを省略することで工数を削減できる可能性がある。

6. おわりに

6.1. まとめ

実際に不具合を誘発させた要求仕様の表現を分析し、「修飾語」または「同義語」を使用している文が、不具合を誘発する要因となることを明らかにした。

本研究では、要求仕様書から修飾語・同義語が使用されている曖昧な文を検出するために、「形態素ベースレビュー」を考案した。評価の結果、形態素ベースレビューは、以下の条件を満たす手法であることが確認できた。

- ・ “上から順に文を読む”アドホックレビューで見逃した要求仕様の曖昧な文を検出できる
- ・ 実際に不具合を誘発した要求仕様の曖昧な文を検出できる
- ・ 要求仕様が自然言語で定義されている稼働中のプロジェクトに対して導入しやすい

要求仕様書に対して、形態素ベースレビューを実施することで、「仕様の誤解釈」「検索での関連仕様の抽出漏れ」を防止し、不具合を減らす効果が期待できる。更に、形態素ベースレビューと Domain Word Modeling を組み合わせることで、過去に不具合を誘発したすべての種類の曖昧な文を検出することが期待できる。

文献[16]では、推敲の効果を上げるために、多様な読み方で多様な推敲を行うことが良い方法であるということが、推敲のコツとして示されている。リーディング技法に対しても、推敲のコツを当てはめることができる。文献[17]でも、要求仕様書の特性に着目し、レビュー手法毎に、検出しやすい欠陥の特徴を明らかにする研究がなされている。要求仕様に対するレビューで、欠陥を見逃さないためには、一つのリーディング技法だけを使用するのではなく、複数のリーディング技法を使用することが有効と思われる。形態素ベースレビューは完全性や妥当性の検証はしにくいいため、形態素ベースレビューも他のリーディング技法に加えて実行することが有効と思われる。

文献[18][19]では、形態素解析を利用したレビュー支援手法が提案されている。同じ形態素を入力とするレビュー手法を組み合わせることで、効果を高められる可能性がある。

6.2. 今後の課題

今後は、形態素ベースレビューの実開発での活用を継続させながら、有効性と効率性を更に高めるために、以下の取り組みを行う。

- ・ 手法適用の効果の確認
手法適用による効果の大きさは確認できていない。形態素ベースレビューで防止できる不具合の種類については確認できているが、防止できる不具合の量については確認できていない。
形態素ベースレビューの適用前後で、不具合の傾向がどう変化するかを分析する。また、効果確認を通して、形態素ベースレビューにより効果を得るための前提条件も明らかにする。
- ・ 手法の適用対象の拡大
自分のプロジェクトで定義した要求仕様書だけ

でなく、要求仕様定義の入力となる文書に対しても、受け入れ確認の位置づけで形態素ベースレビューを適用し、効果を確認する。

また、例えばマニュアルなど要求仕様書以外の文書に対しても形態素ベースレビューを適用し、効果を確認する。

- ・ 修飾語の抽出手順の改善
曖昧な文の検出の漏れと誤りを減らすために、不具合事例をもとに、曖昧な修飾語を検出するための判定条件を見直す。具体的には、確認対象とする品詞を見直す。評価結果から、確認対象は形容動詞のみでも効果があると考えられる。
- ・ 同義語の抽出手順の改善
同義語の抽出にかかる工数を削減するために、手順の改善を検討する。評価の結果をもとに検討した同義語の抽出手順の改善案を図 5 に示す。改善した手順では、形態素ベースレビュー後に、形態素解析の結果から、ドメイン用語の辞書を作り、作成した辞書を次の形態素ベースレビューの入力とする。そして、確認対象の単語の数が多くならないうちに、小さく継続的にこのサイクルを回す。これにより、同義語の抽出作業の効率を向上させられる可能性が高い。また、要求仕様定義時に、辞書を参照すれば、同義語を使用することが予防され、同義語のチェック自体を省略できる可能性もある。

更に、Word2Vec^[20]などの自然言語処理を活用し、辞書を作成せずに同義語の抽出を自動化することを検討する。

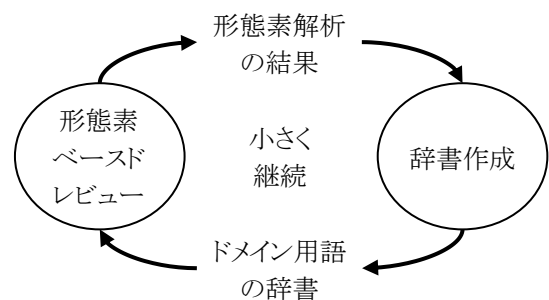


図 5 同義語の抽出手順の改善案

謝辞

本研究にご参加いただいた 株式会社デンソークリエイトとSky株式会社プロジェクトメンバに感謝の意を表す。

本研究のきっかけをいただいた ソニー株式会社 栗田太郎氏, 国立情報学研究所 石川冬樹氏, 第 35 年度ソフトウェア品質管理研究会研究コース5のメンバに感謝の意を表す。

参考文献

- [1] 厳密な仕様記述WG委員, “厳密な仕様記述入門”, 独立行政法人情報処理推進機構, 2013
- [2] 山本修一郎, “要求工学 連載第87回 要求文の曖昧さの抽出法”, <https://www.bcm.co.jp/site/youkyu/youkyu87.html>, 2005
- [3] 阿部圭一, “情報伝達型の日本語文章に現れるあいまい表現の類型化とその改善例”, 情報処理学会デジタルプラクティス Vol.5 No.1 p.70-79, 2014
- [4] 誉田直美, “レビュー技術動向”, ソフトウェア品質シンポジウム 2017, 2017
- [5] 柏原一雄, 長井亘, 不破慎之介, 林香織, 石川冬樹, 栗田太郎, “要求仕様の誤解釈を検出する Domain Word Modeling の提案”, ソフトウェア・シンポジウム 2020, 2020
- [6] 原正幸, 松本裕治, “ipadic version 2.7.0 ユーザーズマニュアル”, 奈良先端科学技術大学院大学, 2003
- [7] 井上昇, 高橋宏季, 位野木万里, “振る舞い用語の同義語に着目したあいまいな要求仕様の検証手法の提案”, 情報処理学会第81回全国大会講演論文集, 2019
- [8] 中島千壽, 高橋宏季, 位野木万里, “要求仕様のあいまいさを解消するためのプロセスフレームワークの提案”, 情報処理学会第82回全国大会, 2020
- [9] 西林克彦, “わかったつもり 読解力がつかない本当の原因”, 光文社新書, 2005
- [10] 村田真樹, 内元清貴, 馬青, 井佐原均, “日本語文と英語文における統語構造認識とマジカルナンバー7±2”, 自然言語処理 6巻 7号 p.61-71, 1999
- [11] 河野哲也, 猪塚修, 藤森麻紀子, 本間周二, 茂中義典, “キーワードベースドレビューードキュメントのあいまいさや不備に着目したレビュー手法”, ソフトウェアテストシンポジウム東京 2010, 2010
- [12] 位野木万里, 近藤公久, “省略と修飾パターンを用いた用語不一致検証による要求仕様の一貫性検証支援ツールの実現と適用評価”, コンピュータソフトウェア 35巻 3号 p.109-127, 2018
- [13] 保田明夫, “形態素解析と分かち書き処理”, テキスト・マイニング研究会 テキスト型データのマイニング, p.145-180, https://www.wordminer.org/wp-content/uploads/2013/04/63_30.pdf, 2006
- [14] 樋口耕一, KH Coder 3, <http://khc.sourceforge.net/>, 2019
- [15] 樋口耕一, “フリーソフトウェア「KH Coder」による計量テキスト分析: 手軽なマウス操作による分析からプラグイン作成まで”, 研究報告人文科学とコンピュータ(CH) Vol.2015-CH-107 No.9 p.1-2, 2015
- [16] 結城浩, “数学文章作法 推敲編”, ちくま学芸文庫, 2015
- [17] 岡本博幸, 内山敬太, 鈴木彩子, 高橋一仁, 西山茂, 野中誠, “要求仕様書の特性に着目した個人レビュー手法の実験的評価”, ソフトウェア品質管理研究会 第20年度 (2004年度) 分科会成果報告, 2005
- [18] Takehiro Wakabayashi, Shuji Morisaki, Norimitsu Kasai, Noritoshi Atsumi, Shuichiro Yamamoto, “Tool Supported Detection of Omissions by Comparing Words between Requirements and Design Document”, Journal of Information Processing 28巻 p.136-149, 2020
- [19] Shuji Morisaki, Masato Nishiguchi, Tetsuya Yonemitsu, Atsushi Motoyama, “A Method to Remove Extraneous Words in Defect Log by Using Common Vocabulary”, コンピュータ ソフトウェア 37巻 2号 p.2_120-2_126, 2020
- [20] 伴凌太, 高橋宏季, 位野木万里, “word2vecを用いた同義語辞書自動作成手法の提案と適用評価”, 情報処理学会第81回全国大会講演論文集, 2019