

品詞に注目した仕様書の論理記述変換手法 A Method for Converting Specification Documents into Logical Descriptions Focusing on Parts of Speech

松上 絢亮[†]
Kensuke Matsukami

岸 知二[†]
Tomoji Kishi

1. 研究背景

システム開発工程におけるシステムテストケースの設計では、テスト設計者が製品の仕様書からテストに必要な入力項目・確認項目を抽出し、テストケースが作成される。一方、自然言語で書かれた仕様書には文章に曖昧な表現を含むためテスト設計者の理解力が必要とされ、論理関係の解釈誤りが発生し、不適切なテストケースが作成されるという課題がある。また、現代では多機能な製品が増えたことによるシステムの複雑化に伴い、テストケースの作成には膨大な時間がかかり、開発現場ではテスト作業コストの削減が求められている。したがって、適切なテストケースを短い作業時間で効率良く作成できるような、テスト設計者への支援が重要である。

2. 背景知識

2.1 MeCab

MeCab とは、形態素解析エンジンの 1 種である[1]。形態素解析とは、文章に対して意味を持つ最小単位の単語に分解し、それらの品詞などを判別することである。MeCab の特徴として、学習モデルに CRF (条件付き確率場) を採用しており、他の形態素解析エンジンと比べ高速に動作する。また、様々なプログラミング言語に利用されており、本研究では Python にて実装する。

2.2 セミ形式記述

青山らは、自然言語で書かれた仕様書に含まれる論理関係の曖昧さを排除するための記述方法を提案した[2]。本研究では、この記述方法をセミ形式記述と呼ぶ。図 1 にセミ形式記述の例を示す。

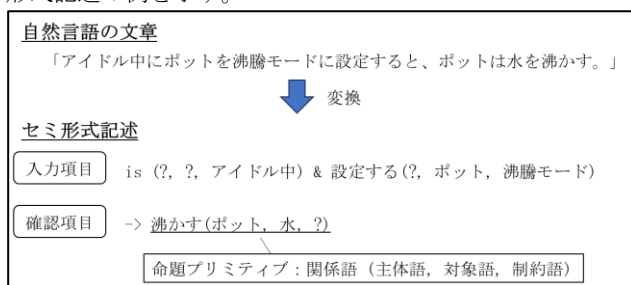


図 1 セミ形式記述の例

セミ形式記述では、仕様書の入力項目・確認項目をそれぞれ「関係語 (主体語, 対象語, 制約語)」として表現し、この形式を以降では命題プリミティブと呼ぶ。関係語には動詞の終止形か、「is」が入る。「is」は図 1 の「アイド

[†] 早稲田大学 創造理工学研究科 経営システム工学専攻

ル中に」のように状態を表す句に用いる。また、「?」は文中から主体語などが判断できないときに用いる。

命題プリミティブ間の論理関係は以下の 4 つの論理記号によって表現する。

「!」: not 「&」: and

「|」: or 「->」: 入力項目と確認項目の関係

論理記号を用いて入力項目・確認項目の論理関係を明確にすることができる。特に、「->」は機能仕様で見られる「入力項目を満たすならば、確認項目をもたらず」という意味を含む記述に用いられ、図 1 のように「入力項目 -> 確認項目」となる。

3. 関連研究

3.1 テストケース生成のためのシステム仕様書の論理記述変換アルゴリズム

青山らは自然言語で書かれた仕様書の曖昧さを取り除くため、仕様書をセミ形式記述へと変換するアルゴリズムを提案した[2]。アルゴリズムでは、自然言語文章の構文解析結果を基に独自のルールベースで変換を行う。そのため、構文解析による文章の係り受け構造に誤りがあった場合には、変換アルゴリズムでは対処できず、結果として誤ったセミ形式記述が作成されてしまうという問題点がある。

4. 研究目的

先行研究の問題点である構文解析誤りによる変換ミスを防ぐため、単語の品質に注目したルールベースの簡易的な変換アルゴリズムを提案する。そこで、本研究の目的は、提案手法によるセミ形式記述変換ツールの開発とその評価である。

5. 提案手法

図 2 に提案手法の概要を示す。

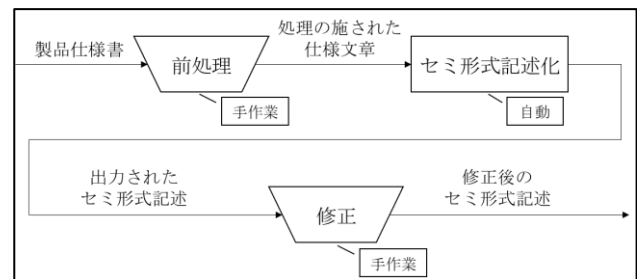


図 2 提案手法の概要

まず、前処理では、製品仕様書の文章を抽出する。このとき、グラフや数式や見出しなどは省くものとする。また、ツールにおけるセミ形式記述変換を容易にするため、語尾や文末表現を言い換える処理も行う。セミ形式記述化では、処理の完了した文章をツールに入力して、セミ形式記述へと変換する。修正プロセスでは、出力されたセミ形式記述

をレビューし、命題プリミティブや論理記号に誤りがあれば修正を行う。修正されたセミ形式記述をテストケース作成に使用するものとする。

図 3 に変換アルゴリズムの全体像を示す。まず、形態素解析を行い、各単語の品詞を修正する処理を行う。その後、動詞を関係語に割り当て、命題プリミティブのリストを作成する。格助詞「は」「が」の前にある名詞を主体語、「を」の前にある名詞を対象語として、直後にある動詞の命題プリミティブのリストに格納する。制約語は、「時」「場合」など状態を表す単語の前の名詞とし、同様に格納する。論理記号は、文章内に特定の単語があるとき、もしくは特定の単語の組み合わせがあるときに、どの種類の論理記号を付与するのルールベースで定める。このようにして、セミ形式記述への変換を行う。

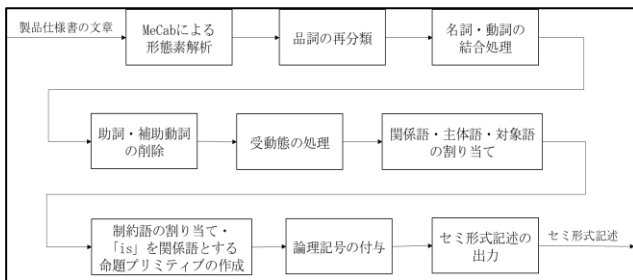


図 3 変換アルゴリズムの全体像

6. 評価実験

6.1 実験目的

開発したセミ形式記述への変換ツールが、適切に仕様書からセミ形式記述へと変換できたか精度を確認することが目的である。また、適切に変換できない文章がある場合、その特徴や原因を分析する。

6.2 実験方法

仕様書には、先行研究と同様に話題沸騰ポットの仕様書を用いて実験を行う。また、この仕様書を著者自身が手作業でセミ形式記述に変換したものを正解データとして、評価を行う。

実験手順として、まず前処理により機能について述べられている文章を抽出する。この際に、図やグラフや数式は省く。また、見出しがありその単語に関する説明が続いているような文章には、単語を文中に補完することとする。抽出した文章をツールに入力し、セミ形式記述へと変換された結果と、手作業による正解データを比較する。ツールの精度を調査することが目的のため、修正プロセス前のセミ形式記述が評価対象となる。

また先行研究との比較のため、命題プリミティブと論理記号における TP (True Positive)、FP (False Positive)、FN (False Negative)の数を測定する。ツールによる命題プリミティブの集合を P_t 、手作業による命題プリミティブの集合を P_m としたとき、TP、FP、FN は以下の命題プリミティブの数となる。

TP : P_t と P_m のどちらにも含まれている数

FP : P_t に含まれているが、 P_m に含まれていない数

FN : P_t に含まれていないが、 P_m に含まれている数

論理記号の場合でも同様に数えるものとする。

6.3 評価方法

評価項目として、命題プリミティブと論理記号に対して Precision、Recall の 2 つの項目を設定する。Precision と Recall では、どの程度適切に変換できているのか測る。また、2 つの項目を先行研究の結果と比較することで、本ツールの精度を相対的に判断する。

6.4 実験結果

仕様書から機能を含む全 62 文を抽出し、ツールによりセミ形式記述へと変換した。出力結果と手作業による解答データを比較した結果を表に記す。表 1 では、Precision と Recall の値、表 2 では命題プリミティブ数ごとの一致率についてまとめた。

表 1 Precision と Recall の先行研究との比較

	TP	FP	FN	Precision	Recall
命題プリミティブ (本研究)	93	49	50	65.5%	65.0%
命題プリミティブ (先行研究)	147	65	62	69.3%	70.3%
論理記号 (本研究)	103	23	26	81.8%	79.8%
論理記号 (先行研究)	66	13	27	83.5%	71.0%

6.5 考察

表 1 より、各値で本研究と先行研究との大きな差は見られなかった。この結果から、本ツールは構文解析誤りによる変換ミスは防ぐことができたが、他の要因により変換できない文章が増加していたことがわかる。本ツールでは、命題プリミティブ数が 2 つ以下の単純な文章では 7 割以上を適切に変換できていたため、変換できない原因として、文章の構造が複雑であり対応できない場合と、複合動詞のような単語を含む文章に対応できない場合の 2 種類あることが判明した。前者の場合では、セミ形式記述化後のレビュー時に修正を加えるといった対応が必要となる。後者の場合では、セミ形式記述化前の仕様書の文章に処理を施すといった対応が必要となる。現状、どちらも手作業による対応が不可欠であるといえる。

7. おわりに

本研究では、自然言語で書かれた仕様書をセミ形式記述へと変換するツールを開発し、話題沸騰ポットの仕様書に適用する実験を行いツールの評価を行った。先行研究と比べ同程度の精度が得られ、構造の単純な文章に対しては十分に變換できており、有効であることが確認できた。今回、簡易的な変換アルゴリズムであったため、今後ルールを追加することにより複雑な文章でも変換が可能となるだろう。

参考文献

- [1] Taku Kudo, Kaoru Yamamoto, Yuji Matsumoto, "Applying Conditional Random Fields to Japanese Morphological Analysis", Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, pp.230-237 (July 2004).
- [2] 青山 裕介, 黒岩 文瑠, 久代 紀之, "テストケース生成のためのシステム仕様書の論理記述変換アルゴリズム", 情報処理学会論文誌, Vol.61, No.3, pp.521-534 (2020).