Vol.2018-SE-198 No.7
2018/3/9

自然言語仕様書からの試験ケース生成のための条件・動作の同定手法

村上響一[†1] 青山裕介[†1] 村上神龍[†1] 久代紀之[†1]
牧茂[†2] 田畑一政[†2] 神代勉[†2] 中村潤[†2]

概要：システム開発において，テスト設計者は自らの知識・経験に基づいて記載された文章から数百にも及ぶ試験ケースの作成を行う．試験ケースの作成は手作業で行われるため，人為的ミスの発生や，テスト仕様書がテスト設計者のスキルに大きく依存してしまうという課題があった．本研究では，自然言語で記載されたシステム仕様書から試験ケースを作成するまでのプロセスの提案を行い，欠落情報の抽出，条件・動作の同定，論理構造の可視化，試験ケースの作成を機械的に行うツールの開発を目的とする．本論文では，上記プロセス解決のために，自然言語の構文解析結果からセミ形式記述，条件・動作の同定機能の実装と，その評価実験を行った．

1. はじめに

システム開発において，テスト設計者はシステム仕様書をもとにテスト仕様書を作成する．システム仕様書は多くが自然言語で記載されており，テスト設計者は自らの知識・経験に基づいて記載された文章から，数百にも及ぶ試験ケースの作成を行う．この際，テスト設計者は手作業により試験ケースの作成を行うため，人為的ミスの発生，テスト仕様書の作成がテスト設計者のスキルに大きく依存してしまうという課題や，テスト設計者の持つ設計プロセスが暗黙的であり，設計の根拠がレビュアーに伝わりづらいという課題がある [1].

本研究では，自然言語の構文解析を行うことで，自然言語で記載されたシステム仕様書から試験ケースを作成するまでのプロセスの提案を行い，提案プロセスによる試験ケースの生成を機械的に行うツールの開発を目的とする．本論文では作成した自然言語の構文解析に関わる機能に対する評価を行う．

2. 課題解決へのアプローチ

課題解決のために，以下のようなテスト設計プロセスを提案する．また，提案プロセス達成のための手法について述べる．

2.1 提案プロセス

図 1は本研究で提案する設計プロセスである[1]．システム仕様書からテスト仕様書を作成するプロセスを定義することで，これまでテスト設計者が暗黙的に持っていた設計プロセスの明示化を行う．この際，プロセスごとに出力される生成物のレビューを行うことで，試験ケース作成までのレビューの品質向上を目指す．

2.2 提案プロセスの詳細

図 1に示した提案プロセスにより，システム仕様書からテスト仕様書への変換の概要を説明する．全体のプロセスは1：欠落情報の抽出，2：条件・動作の同定，3：試験ケースの抽出，4：品質要求試験ケースの抽出の4つに分けられる．以下に，各プロセスにおいてレビュー対象となる生成物とその詳細について述べる．

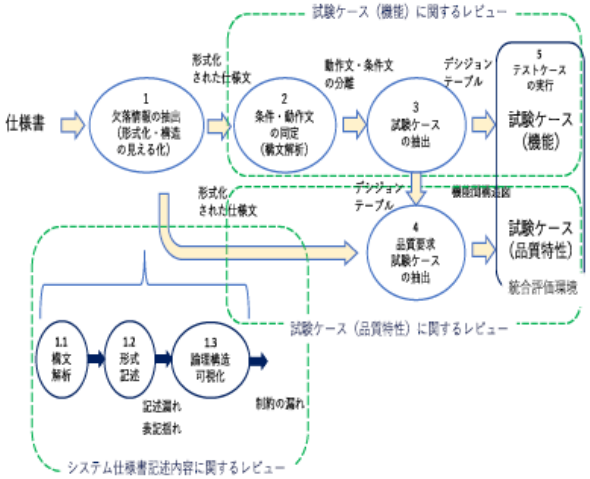

図 1 テスト設計の提案プロセス[1]

2.2.1 プロセス 1：欠落情報の抽出

これまで，システム仕様書に記載された文章からテスト仕様書を作成する際，テスト設計者は記載された文章の主語・目的語などの記述漏れや誤りといった欠落情報や論理関係の間違いを発見し，手作業で訂正していた．本研究では文章の欠落情報の発見・論理関係の可視化を機械的に支援するため，さらにプロセス1を1.1：構文解析，1.2：形式記述，1.3：論理構造の可視化の3つのサブプロセスに分ける．

2.2.1.1 プロセス 1.1：構文解析

システム仕様書に記載された文章を構文解析するため，自然言語処理を行う必要がある．

自然言語処理は大きく分けて次の4つのステップに分けることができる[2][3].

1. 形態素解析：文章を単語に分割する．単語が語形変化している場合は，原型へ戻す．単語の品詞を決定する．
2. 構文解析：単語間の構文的関係を決定する．
3. 意味解析：単語，文の意味を決定する．
4. 文脈解析：複数の文にまたがる処理を行う．

本研究では，文章の形態素解析を形態素解析ツールjuman[4]，構文解析を構文解析ツールknp[5]により行う．

†1 九州工業大学
Kyushu Institute of Technology.
†2 AVC テクノロジー株式会社
AVC Technology Co. ,Ltd.

2.2.1.2 プロセス 1.2：形式記述

文章を，関係語（主体，対象，［制約，{“,”制約}］）という形式的な表現で記述することで[6]，欠落情報の抽出を行う．以下，この形式記述のことをセミ形式記述と呼ぶ．この際，主語や目的語の抜けがあればセミ形式化した際，抜けに相当する箇所を「??」と表示することで，レビュアーは文章の記述漏れを発見することができる．また，図 2のように可視化を行うことで文章の直感的理解をサポートする[7]．

例えば，「アプリは画面上に表示する．」という文章をセミ形式化すると，「表示する（アプリ，??，画面上に）」と表現できる．この際，レビュアーは「??」によって「表示する」という関係語の対象が欠落していることを発見することができる．複文も論理式を用いて表すことが可能であり，「ユーザはアイコンをタップして，アプリを起動する．」という文章をセミ形式化すると，「タップする（ユーザ，アイコン）∧起動する（??，アプリ）」と表現する．また，否定表現についても，対応する関係語に「¬」を付けることで表すことができる．

テスト設計者はシステム仕様書の文章をセミ形式記述として表すことで，省略または欠落している情報を発見し，それを補うことでシステム仕様書に含まれた曖昧性を排除した試験ケースを作成することができる．

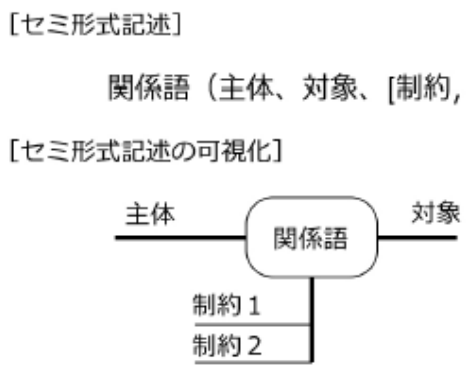

図 2　セミ形式記述および可視化[15]

2.2.1.3 プロセス 1.3：論理構造の可視化

上記手法により文章をセミ形式化すると，関係語ごとに単文化を行うため，もとの文章では表現されていた単文相互の論理関係が失われてしまう．そのため，セミ形式記述を図 3のように単文ごとに関係を持たせるようなネットワーク的に表現することで，文章の意味を保持したまま文章の論理構造可視化を行う．この表現を命題ネットワークと呼び，論理包含を矢印．選言または連言は縦に結合して表現する．さらに，単文間の論理構造として，AND, OR, XOR, Conditional, One, Inclusive, Require, Maskの関係の記述が可能である．[12][15]（図 3）

テスト設計者はシステム仕様書に記載された文章の論理構造を命題ネットワークにより可視化することで，仕様書に記載された文章の論理が破綻していないかということを確認することができる．また，レビュアーに試験ケース作成の根拠について明示的に示すことが可能になる．

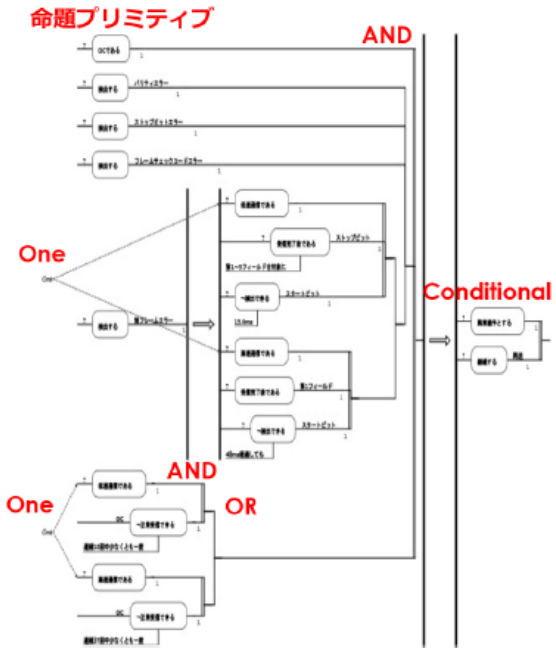

図 3　命題ネットワーク表示例[15]

2.2.2 プロセス 2：条件・動作の同定

システム仕様書に記載された文章から，機械的に試験ケースを作成するため，文中に含まれる条件・動作の同定を行う．増田らの提案する，文章を条件句と動作句として捉え，その係受け関係から意味的役割を付与するアプローチ[2]をベースとして，条件・動作の同定を行う．

例えば，話題沸騰ポット第6版[9]の記述例である，「水温が110℃を超えた場合，ヒータ用電源をoffし，30秒間ブザーを鳴らす．」という文章の場合，条件としては，「水温が110℃を超える」の1つ，動作としては「ヒータ用電源をoffする」，「30秒間ブザーを鳴らす」の2つが得られる．

上記例のように，1文を条件・動作としてそれぞれ単文化してしまうと，1つの文としての論理関係が失われてしまうため，自然言語に対して条件・動作の同定を行う際，セミ形式記述と命題ネットワークを組み合わせてこれら問題の解決に取り組む．

2.2.3 プロセス 3：試験ケースの抽出

試験ケースの作成には様々な手法があるが，本研究ではプロセス2により同定した条件・動作を用いるため，条件と動作の組み合わせから試験ケースの記述を行うディシジョンテーブルテストを採用する[8]．例えば，上述した例文において同定した条件・動作からディシジョンテーブルを作成すると図 4のようになる．

		ルール1	ルール2
条件1	超える（水温、110℃）	Y	N
動作a	offする（ポット、ヒータ用電源）	Y	N
動作b	鳴らす（ポット、30秒間ブザー）	Y	N

図 4　条件・動作から作成したディシジョンテーブル例

2.2.4 プロセス 4：品質要求試験ケースの抽出

本論文では，システム仕様書に記載された文章から機能試験に関する試験ケースを作成することを目的とするため，品質要求試験ケースの抽出については今後の課題とする．

3. 実装機能

上記プロセスについて，プログラミング言語Python[10]により，自然言語を構文解析した結果から文章の条件・動作を同定する機能，文章をセミ形式記述化する機能の実装をおこなった．

3.1 実装のための準備

上記機能の実装にあたり，変換アルゴリズムのために[2]を参考として以下のような定義を行う．

- 文をSとする
- 文Sの文節を$m(i)$とする
- 文末の文節番号をTとし，文末文節を$m(T)$とする
- $m(i)$の子文節の文節の集合をD_iとする
- $m(i)$とD_iの集合を句B_iとする($m(i)$は句B_iの終端文節となる)

文章をjumanにより形態素解析した各分節に対して，knpによる解析で得られるfeature情報をもとに属性付けを行う．文節に付与する属性は，type, case, parallel, attributeの4種類である．以下に各属性の定義を示す．各表はknpにより得られるfeatureと，それによって決定されるラベルの種類である．

○type属性に関する定義

type属性は，action（動作），state（状態），epithet（形容詞），others（その他）の4種類のラベルのいずれかを持つ．表 1にその定義を示す．

表 1　featureとtype属性のラベル関係

type	feature
action	<用言：動>
state	<用言：判>
epithet	<用言：形>
others	上記以外

○case属性に関する定義

case属性は，main（主格），object（目的格），others（その他の格）の3種類のラベルのいずれかを持つ．表 2にその定義を示す．

表 2　featureとcase属性のラベル関係

case	feature
main	<解析格：ガ>, <ハ：True>かつ<主題表現：True>
object	<解析格：ヲ>, <係：ヲ格>
others	<係：ニ格>, <係：ヘ格>, <係：カラ格>, <係：ヨリ格>, <係：デ格>, <係：マデ格>, <係：ノ格>, <係：ト格>

○parallel属性に関する定義

parralel属性は，and（連言），or（選言）の2種類のラベルのいずれかを持つ．表 3にその定義を示す．

表 3　featureとparallel属性のラベル関係

parallel	feature
and	<並列タイプ：AND>, <ID：～て(用言)>, <ID：～で(判)>, <ID：～ても>, <ID：～ながら>, <ID：～て>かつ<用言：動>
or	<並列タイプ：OR>, <ID：～たり>

○attribute属性に関する定義

attribute属性は，interrogative（疑問詞），condition（条件），restriction（制限）の4種類のラベルのいずれかを持つ．conditionは「～する場合」など関係語と条件となる語句が異なる形態素である場合，restrictionは「～すると」など関係語自身が条件となる形態素である場合に用いる．表 4にその定義を示す．

表 4　featureとattribute属性のラベル関係

attribute	feature
interrogative	<疑問詞：True>かつ<解析格：ヲ>
condition	<外の関係：True>かつ「場合」，「時」，「際」， <外の関係：True>かつ<ID：～ため>, <外の関係：True>かつ<時間：True>かつ<相対名詞：True>
restriction	<係：連用>かつ<ID：～ば>, <ID：～たら>, <ID：～ので>, <ID：～が>, <ID：～ように>, <ト：True>

例えば，「私は昼ご飯を食べて，運動をする．」を入力文章Sとすると，形態素解析により$m(1)$={私}，$m(2)$={昼ご飯}，$m(3)$={食べる}，$m(4)$={運動}，$m(5)$={する}及びD_3 = { $m(1)$, $m(2)$}，D_5 = { $m(4)$}を決定することができる．また，knpによる解析でm(3)は<用言：動>と<ID：～て(用言)>をfeatureとして持つことから，m(3)の属性をm(3)={食べる, type:action, parallel:and}と決定することができる．

3.2 アルゴリズム

実装に際して，文章をセミ形式記述への変換するアルゴリズムと条件・動作を同定するアルゴリズムを，3.1で定義した属性により定める．

3.2.1 セミ形式記述変換アルゴリズム

文章をセミ形式記述に変換するアルゴリズムを図 5に示す．$m(i)$のtype属性がaction, state, またはm(T)かつepithetの時，$m(i)$を関係語とする．

例外として，ステップ8.でobject(目的格)がinterrogative(疑問詞)の場合，それを関係語としてセミ形式化したものを，$m(i)$を関係語とするセミ形式記述の目的格に挿入する．

例）入力：私はそれが事実であるかどうか分からない．
　　出力：￢分かる（私，事実であるかどうか（それ, ??））

1.入力文章 S を形態素・構文解析(juman/knp)
2.文節$m(i)$ごとに属性決定
3. for $m(1)$: $m(T)$ [$m(1)$から $m(T)$まで文節の type 属性を見ていく]
4. if $m(i)$の type 属性が関係詞
5. then $m(i)$を関係詞としてセミ形式記述化
6. セミ形式の初期記述を「$m(i)$(?-main, ?-object)」とする
7. for all D_i [$m(i)$の子文節の集合について case 属性を見ていく]
8. if $D_i(j)$の case 属性が main(主格)または object(目的格)
9. then セミ形式記述の「?-main」または「?-object」の位置に$D_i(j)$を挿入
10. else セミ形式記述の最後の位置に$D_i(j)$を挿入
11. end if
12. end for
13. if $m(i)$の parallel 属性が and または or
14. then $m(i)$を関係詞とするセミ形式記述に論理記号∧または∨を付与
15. else $m(i)$を関係詞とするセミ形式記述は独立している
16. end if
17. else 何もしない [$m(i)$の type 属性が関係詞でない]
18. end if
19.end for

図 5 セミ形式記述変換アルゴリズム

3.2.2 条件・動作同定アルゴリズム

文章の条件･動作を同定するアルゴリズムを図 6 に示す.

1.入力文章 S を形態素・構文解析(juman/knp)
2.文節$m(i)$ごとに属性決定
3. for $m(1)$: $m(T)$ [$m(1)$から $m(T)$まで文節の attribute 属性を見ていく]
4. if attribute 属性が condition、または$m(i)$の type 属性が関係詞かつ restriction
5. then $m(i)$を終端文節として持つ句B_iを条件句とする
6. else $m(i)$を終端文節として持つ句B_iを動作句とする
7. end if
8.end for

図 6 条件・動作同定アルゴリズム

3.3 受動態の変換

受動態に関して，関連研究では文の主題を重視するため受動態のまま扱うとしている[13]. 本研究では表記揺れを機械的に是正するために文章をセミ形式記述に変換する際，受動態も能動態に変換する．受動態は関係語の語尾が「れる」と活用されている場合であるが，「れる」には４つの用法がある[14]

1. 受け身：携帯が壊される
2. 可能：りんごは食べられる
3. 自発：彼のやる気が感じられる
4. 尊敬：先生が出発される

knpにおいて，上記表現を解析すると，受動態と判断される関係語については＜受動＞または＜受動｜可能＞という解析結果が得られる．

本研究では，仕様書において尊敬表現が使用されることはないとして，尊敬表現についての受動態の変換は考えないものとする．自発は受け身文として扱い，例えば上記例文3. に対して「感じる（??，彼のやる気）」とセミ形式化しても対象語としての意味を表現できている．基本的な受動態については，主格（ガ格），対象語（ヲ格），動作の主体（ニ格）を置き換えることで能動態に変換することが可能である．例えば，「携帯が壊される」は「壊す(??，携帯)」としても文の意味を保持したままセミ形式記述で表現できている．ただし，一つの文章に主格，対象語，動作の主体が同時に含まれる受動態については，能動態に変換する際に，受動態の主格＋対象語を能動態の一つの対象語として扱うことで文章本来の意味を失わないようにする．例えば，「私は弟に携帯を壊された」という文章であれば，「壊す（弟，私の携帯）」と変換する．

3.4 実装機能の出力例

2.1提案プロセスで述べた，セミ形式記述，条件・動作の同定機能の出力例を示す．本実装では，3.2.2アルゴリズムにより条件･動作文を同定した結果と，3.2.1アルゴリズムによるセミ形式記述の変換を組み合わせた出力を行っている．

3.4.1 セミ形式記述変換の出力例

実装機能により，入力した文章をセミ形式記述として出力した結果を図 7，図 8に示す．例文は，話題沸騰ポット第6版[9]から引用した．

①宣言的な文章

実装した機能に，「サーミスタはポット内の水温を検出します」という文章を入力すると，図 7の結果が出力される．

動作:検出する(サーミスタ,ポット内の水温)

図 7 宣言的な文章を入力した出力結果

②並列な動作を含む文章

実装した機能に，「ポンプは，ポット内の水を吸い上げて，給湯口から排出します．」という文章を入力すると，図 8の結果が出力される．

動作:吸う(ポンプ,ポット内の水)∧ 動作:排出する(??,??,給湯口<カラ>)

図 8 並列な動作を含む文章を入力した出力結果

3.4.2 条件・動作同定機能出力例

実装機能により，入力した文章の条件・動作文を同定し，セミ形式化して出力した結果を図 9，図 10に示す．同様に例文は話題沸騰ポット第6版から引用した．

①条件となる語を含む文章の場合

実装した機能に，「水温が 110℃を超えた場合，ヒータ用電源を off して，30 秒間ブザーを鳴らします．」という文章を入力すると，図 9 の結果が出力される．

条件(場合):超える(水温,110℃)
動作:offする(??,ヒータ用電源)∧ 動作:鳴らす(??,30秒間ブザー)

図 9 条件語となる語を含む文章を入力した出力結果

②関係語自身が条件となる文章の場合

実装した機能に，「給湯ボタンを押すと，ポンプを動作させて給湯口から水を排出します．」という文章を入力すると,図 10の結果が出力される．

条件:押す(??,給湯ボタン)
動作:動作する(??,ポンプ)∧ 動作:排出する(??,水,給湯口<カラ>)

図 10 関係語自身が条件となる文章を入力した出力結果

3.4.3 受動態の変換

実装機能により，受動態の文章は能動態に変換して出力する．出力結果を図 11，図 12に示す．

①基本的な受動態

実装した機能に，「携帯が壊される」という文章を入力すると，図 11の結果が出力される．

動作:壊す(??,携帯)

図 11　基本的な受動態を入力した出力結果

②主格，対象語，動作の主体が同時に含まれる受動態

実装した機能に「私は弟に携帯を壊される」という文章を入力すると，図 12 の結果が出力される．

動作:壊す(弟,私の携帯)

図 12　主格，対象語，動作の主体が同時に含まれる受動態を入力した出力結果

4. 実装機能評価実験

3.実装機能について，機械的に同定した主語・目的語の抜け及び条件・動作同定の精度を確認するための評価実験を行った．図 13は実験データとして作成した架空のアプリケーションについての仕様書の文章（16文）である．架空の仕様書を実験データとして使用する理由は以下の通りである．

1. 実際の仕様書だと文量が多く，被験者に負担がかかる
2. 実際の仕様書だと今回の実験で確認したい条件・動作の同定，主語・目的語の漏れ検出と関係のない文章が含まれる
3. 非公開の仕様書の場合，実験データとして公開することができない

本実験の被験者として，同研究室の試験ケース作成・評価の研究を行っている学生3名に協力してもらった．実験の手順を以下に示す．

1. 被験者は文章を読み，各文を試験ケース作成にあたり必要だと思う句節に分ける
2. 分けた句節ごとに，その句節が条件であるか，動作であるかを判断する
3. 句節が論理関係　AND/OR/EXOR　であると判断すれば，論理関係を明示する
4. 各句節について主語・目的語の欠落があるかどうかを判断する
5. 主語・目的語の欠落があると判断し，語の補完ができれば補完する
6. 各句節について主語・目的語の欠落はなく，主語・目的語を必要ないと判断すれば「必要なし」とする

このアプリケーションは、スマートフォン上で動作する。ユーザはこのアプリを使って、撮影した写真を加工することができる。アプリを起動すると、カメラロールにアクセスし、写真の情報を取得する。

アプリが写真の情報を取得する際、ユーザがカメラロールへのアクセスを許可していなければ、通知する。アプリを起動すると、写真一覧を表示し、ユーザは加工したい写真を選択する。

加工ボタンをタップすると、写真の加工を開始する。ユーザは線の描画、またはスタンプを押すことで写真を加工する。描画する線はサイズと色を選択することができる。

加工した写真は保存ボタンをタップすることで保存することができる。写真加工中に、ユーザがアプリを閉じる、またはスマートフォンの電源が切れた場合、変更を保存していなければ、通知する。

アプリはサーバーと通信することで、新しいスタンプをダウンロードすることができる。ダウンロード中に通信障害が起こった場合、ダウンロードを中止して、トップ画面に戻る。

ユーザが加工した写真を消去する時は、消去ボタンをタップする。消去ボタンをタップした後、複数の写真を選択すると、選択した写真を消去することができる。

写真加工中に、消去ボタンをタップすると、変更を保存せずに写真の加工を終了するかどうかをユーザに通知する。ユーザが「終了する」を選択する場合、トップ画面に戻り、選択しない場合、写真の加工を継続する。

図 13　実験に使用した架空のシステムについての文章

4.1 セミ形式記述変換機能について

図 14 は実装機能によって実験データを入力としたときの出力結果である．実装機能によって，主語・目的語が欠落していると判断した場合はセミ形式記述に置いて対応する位置に「??」として表示することで主語・目的語の欠落を検出することができる

動作:動作する(このアプリケーション,??,スマートフォン<デ>)

動作:使う(??,このアプリ)∧ 動作:加工する(ユーザー,撮影した写真)

条件:起動する(??,アプリ)
動作:アクセスする(??,??,カメラロール<ニ>)∧ 動作:取得する(??,写真の情報)

条件(際):取得する(??,写真の情報)
条件:¬許可する(ユーザー,カメラロールへのアクセス)
動作:通知する(アプリ,??)

条件:起動する(??,アプリ)
動作:表示する(??,写真一覧)∧ 動作:選択する(ユーザー,加工したい写真)

条件:タップする(??,加工ボタン)
動作:開始する(??,写真の加工)

動作:線の描画(??,??)∨ 動作:加工する(ユーザー,写真,スタンプを押すこと<デ>)

動作:選択する(描画する線,サイズと色)

動作:保存する(??,加工した写真,タップすること<デ>)

条件:写真加工中(??,??)
条件:閉じる(ユーザー,アプリ)∨ 条件(場合):切れる(スマートフォンの電源,??)
条件:¬保存する(??,??)
動作:通知する(??,変更)

条件:である(アプリ,サーバーと通信すること)
動作:ダウンロードする(??,新しいスタンプ)

条件:ダウンロード中(??,??)
条件(場合):起こる(通信障害,??)
動作:中止する(??,ダウンロード)∧ 動作:戻る(??,??,トップ画面<ニ>)

条件(時):消去する(??,加工した写真)
動作:タップする(??,消去ボタン)

条件(後):タップする(??,消去ボタン)
条件:選択する(??,??)
動作:消去する(??,選択した写真)

条件:写真加工中(??,??)
条件:タップする(??,消去ボタン)
動作:¬保存する(??,変更)∧ 動作:通知する(??,終了するかどうか?(??,写真の加工),ユーザー<ニ>)

条件(場合):選択する(??,「終了する」を)
条件:戻る(??,??,トップ画面<ニ>)∧ 条件(場合):¬選択する(??,??)
動作:継続する(??,写真の加工)

図 14　実装機能によるセミ形式記述変換結果

実装機能により機械的に判断した主語・目的語の欠落判断の精度検証のため，被験者により判断された句節ごとの主語・目的語の抜けと，実装機能により判断された主語・

目的語の抜けの比較を行った．実装機能により主語または目的語が抜けていると判断され，被験者によって主語または目的語が抜けていると判断された場合TP（True Positive），実装機能により主語または目的語が抜けていると判断されたが，被験者によって主語・目的語が抜けていると判断されていない場合FP（False Positive），被験者によって主語または目的語が抜けていると判断されたが，実装機能によって主語または目的語が抜けていないと判断された場合FN（False Negative）として，それぞれの件数を数えた．式(1)，式(2)，式(3)によりTP,FP,FNを数えた結果から，精度（Presicion），再現率（Recall）およびF値（F-Measure）を求めた．結果を表 5に示す．

$$Precision = TP/((TP+FP)) \quad \cdots(1)$$
$$Recall = TP/((TP+FN)) \quad \cdots(2)$$
$$F = 2 \times Precision \times Recall/((Precision+Rcall)) \quad \cdots(3)$$

表 5　主語・目的語の欠落情報抽出精度・再現率・F 値

	精度	再現率	F値
主語の欠落	0.825	0.881	0.852
目的語の欠落	0.741	0.917	0.820

4.2 条件・動作同定機能について

条件・動作同定機能の評価について，実装機能により条件・動作と判断された句節の比較を行った．実装機能により条件または動作と判断された句で，実際に条件または動作である句をTP，実装機能により条件または動作と判断された句で，実際には条件または動作でない句をFP，実際には条件または動作であるが，実装機能により条件または動作と判断できなかった句をFNとした．それぞれの件数を数えて，式(1)，式(2)，式(3)により精度（Presicion），再現率（Recall）およびF値（F-Measure）を求めた.結果を表 6に示す．

表 6　同定した条件・動作の精度・再現率・F 値

	精度	再現率	F値
条件	0.883	0.854	0.868
動作	0.851	0.954	0.90

5. 考察

4.で行った実験結果から，今回評価するセミ形式記述変換機能，条件・動作同定機能についての考察を行う．

5.1 セミ形式記述変換機能について

4.1表 5より，目的語の欠落情報抽出精度が他の項目と比較して低いことが分かる．その理由として，以下の2点が原因として考えられる．

1点目は，実装機能において，目的語を持たない動詞についても，＜ヲ格＞を持つ目的語がなければ欠落しているとして判断していたためである．例えば，「トップ画面に戻る」という単文について実装機能によってセミ形式化すると「戻る（??,??，トップ画面＜ニ＞）と出力され，主語・目的語が欠落していると判断される．しかし，「戻る」という動詞は目的語を持たないため，被験者はこの文について主語の抜けはあるが，目的語の抜けはないと判断した．

2点目は「名詞」＋「サ変動詞」＋「時間条件」となるような文についてである．例えば，「写真加工中に，～」という文章を実装機能によりセミ形式化すると「写真加工中（??, ??）」と出力される．このような文について，被験者は「写真を加工中に」と暗黙的に助詞を補い，目的語の抜けはないと判断した．「ダウンロード中に，～」などの「サ変動詞」＋「時間条件」となるような文については目的語となる名詞が存在しないため，被験者は目的語の抜けがあると判断していた．

上記2点によって，実装機能では目的語の欠落情報が過剰に検出され，精度が低くなった．この考察を踏まえ，目的語を必要としないセミ形式記述については，「??」ではなく「_」と表示することでセミ形式記述により目的語が欠落しているのか，必要ないのかの区別をつける．

また，実装機能では「～すること」などの動詞を名詞化する表現を名詞として捉えており，セミ形式記述に変換していなかった．例えば，「スタンプを押すことで写真を加工する」という文章を実装機能によりセミ形式化すると「加工する（??，写真，スタンプを押すこと＜デ＞）」という出力が得られていた．しかし，被験者による判断では，「スタンプを押すこと」という句節は「ユーザー」という主語が抜けていると判断していた．そこで，制約となる動作句も「押すこと（??，スタンプ）」のようにセミ形式化することで，欠落情報抽出精度の改善を行う．

上述した実装機能の改善に基づき，プログラムの修正を行った．図 15は修正後のセミ形式記述出力結果で，丸で囲った箇所が修正後に出力が変化した箇所である．また，表 7は修正後の主語・目的語の欠落情報抽出精度である．表 5，表 7より，実験から発見できた問題によって実装機能の修正を行った結果，主語の欠落情報抽出の再現率が0.05，目的語の欠落情報抽出の精度が0.07向上した．今後「戻る」などの目的語を持たない動詞については，EDR電子化辞書[11]などを用いて動詞ごとの共起パターンを利用して判断する機能を実装することを検討している．

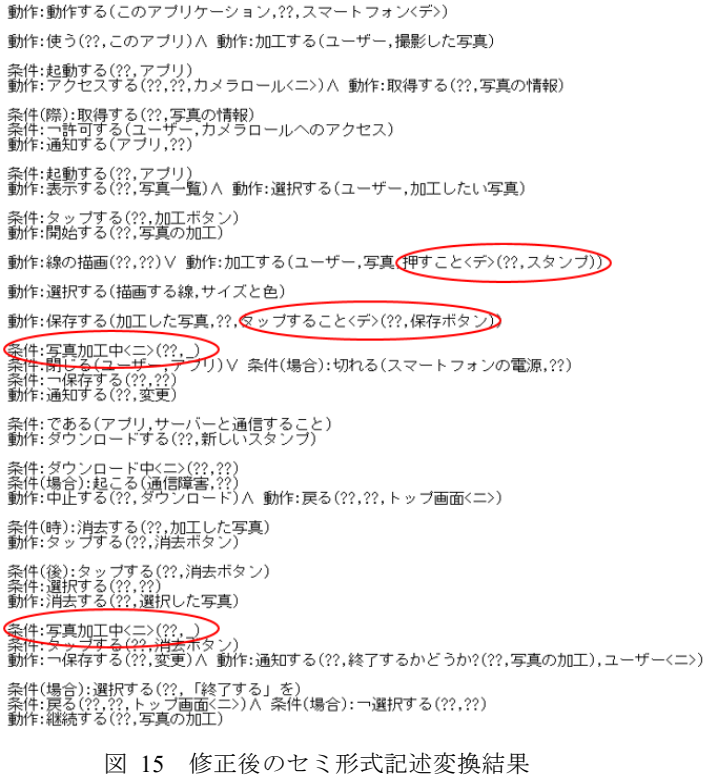

図 15　修正後のセミ形式記述変換結果

表 7　修正後の欠落情報抽出精度・再現率・F 値

	精度	再現率	F値
主語の欠落	0.834	0.940	0.884
目的語の欠落	0.816	0.971	0.886

5.2 条件・動作同定機能について

実装機能によって同定された条件・動作について，条件の精度が0.883，再現率が0.854，F値が0.868となり，動作の精度が0.851，再現率が0.954，F値が0.90となる結果となった．既存研究[2]では，条件の精度0.901，制限率0.960，F値0.930，動作の精度0.985，再現率0.955，F値0.970となっていることから．既存研究よりも低い結果となった．これは，関連研究では，文章の動作句を条件句以外の句として定義しているのに対し，本研究では，セミ形式記述変換するために，単文ごとに条件・動作の判断をしているためと考えられる．

今回の実験により発見できた条件・動作同定精度が下がっている要因の一つとして，接続助詞「て」の扱いがある．例えば，「ユーザはこのアプリを使って，撮影した写真を加工することができる．」という文章に対しては，接続助詞「て」を含む句「ユーザはこのアプリを使う」は条件句，それ以降の句「撮影した写真を加工することができる」は動作句となる．しかし「ボタンをタップすると，音量をあげて画面を明るくする．」という文章があった場合，接続助詞「て」を含む句「音量を上げる」とそれ以降の句「画面を明るくする」は並列関係となる．

また，条件と動作が交互に出現するような重文も精度が下がる一因となる．例えば，「ユーザが「終了する」を選択する場合，トップ画面に戻り，選択しない場合，写真の加工を継続する．」という文章は，それぞれの句に対して被験者は条件句「ユーザが「終了する」を選択する」，動作句「トップ画面に戻る」，「条件句「選択しない」，動作句「写真の加工を継続する」と判断した．しかし，実装機能では動作句「トップ画面に戻る」と条件句「選択しない」が並列関係であると判断してしまい，動作句を条件句と誤判定してしまった．実装機能による出力結果が図 16である．

条件(場合):選択する(??,「終了する」を)
条件:戻る(??,??,トップ画面<ニ>)∧ 条件(場合):¬選択する(??,??)
動作:継続する(??,写真の加工)

図 16　重文を入力したときの出力結果

上記課題は現状の構文解析，アルゴリズムでは対応が難しく，試験設計者に修正を促すなど，レビュー時に修正を行うことで対応する方法を検討している．

6. まとめと今後

本研究ではシステム開発における，テスト仕様書作成のコストを削減するためのツール開発を目的として，機械的に文章のセミ形式記述化，条件・動作の同定を行う機能の実装をした．本論文では，自然言語の構文解析に関わるセミ形式記述化，条件・動作の同定機能についての評価実験を行った．評価実験により実装機能により欠落情報発見，条件・動作文同定精度と，新たな課題を発見することができた．また実際の仕様書は図や表の挿入も多い．これらの情報は命題ネットワークにより文章と図・表の情報をマージすることで整理することができると考えている．今後は条件・動作同定アルゴリズムの精度向上など実装機能の調整を行うとともに，本格的なツール化にむけて各実装機能の統合化を進めていきたい．最終的には，ツールを用いて作成された試験ケース自身の評価及び時間的コスト・作業コストがどのくらい削減されるかを評価する予定である．

謝辞

本研究は科研費 16K00100, JST CREST JPMJCR1304 の支援を受けて実施されたものである．

参考文献

[1] 青山裕介，黒岩丈瑠，久代紀之，自然言語使用からの機能間の並列・順序動作の抽出と左記テスト環境, FITPaper, 2017.
[2] 増田聡，松尾谷徹，津田和彦，試験ケース作成自動化のための意味的役割付与方法，特集ソフトウェア工学の基礎，Vol.34, No.2, pp.16-27, May 2017.
[3] 奥村学，自然言語処理の基礎，コロナ社, 2010.
[4] 京都大学黒橋・河原研究室，自然言語処理のためのリソース, http://nlp.ist.i.kyoto-u.ac.jp/index.php?JUMAN
[5] 京都大学黒橋・河原研究室，ＫＮＰで付与されるfeature一覧, http://nlp.ist.i.kyoto-u.ac.jp/index.php?plugin=attach&refer=KNP&openfile=knp=feature.pdf.
[6] C. Rolland, C.B.Achour: Guiding the construction of textual use case specifications, Data & Knowledge Engineering, Vol.25, Issues 1-2, pp.125-160, ELSEVIR, 1998.
[7] 久代紀之，鳥飼諒一，システムデザインにおける論理的枠みとデータ活用戦略，ヒューマンインタフェース学会誌, Vol.17, No.2, 2015.
[8] 石原一宏，　田中英和　著，田中真史　監修，ソフトウェアテストの教科書，ソフトバンククリエイティブ株式会社, 2012.
[9] 話題沸騰ポット(GOMA-1015型) 要求仕様書[第6版], SESSAME,　2004.
[10] http://www.python.org
[11] 日本電子化辞書研究所:EDR電子化辞書, 1996.
[12] J. マイヤーズ, M. トーマス：ソフトウェアテスト・テストの技法　第2報，近代科学社, 2006.
[13] 日本語文から拡張型述語論理式への自動変換ツール : CONV, 高柳俊祐，上條敦史，石川勉，人工知能学会論文誌 27号5巻 B, pp.271-pp.280, 2012.
[14] 新村出編:広辞苑第四版，岩波出版, 1997.
[15] 久代紀之，藤田裕司，村上響一，青山裕介，リスク認知における知ってる/知らない知識の表出化と可視化，電子情報通信学会, 2018.