

仕様書におけるパラグラフ内構成文の論理構造の可視化

緒方佑亮¹ 青山裕介¹ 久代紀之¹

概要: システム開発の最終工程では品質を担保するためにシステム試験が実施される。システム試験には、テスト仕様書が必要であるが、テスト仕様書は、テスト設計者が自然言語で記述された仕様書を熟読し、記載された各文から、“テスト条件（パラメータを含む）⇒期待値”のタプルを抽出することで生成される。上記の抽出作業は人手で行われ、かつ、自然言語で記述された仕様書には、誤りや曖昧さが含まれるため、その修正・確認のため膨大な工数を必要とし、筆者らの調査によれば、600ページの仕様書で約700時間が必要となっていた。上記課題を解決するために、本研究では仕様書記載の各文を形式的な記述へ変換することで、自然言語で記述された仕様書内に発生する誤りや曖昧さの補完を促したのちに、各文の構造を視覚的に表現するために“条件⇒期待値”の論理包含関係を基準としてパラグラフ内構成文の論理構造の可視化ツールの開発を行った。

キーワード: 文間構造, 談話処理

Visualization of the logical structure of a composition in a paragraph in a specification document

YUSUKE OGATA^{†1} YUSUKE AOYAMA^{†1}
NORIYUKI KUSHIRO^{†1}

Abstract: In the final process of system development, system tests are conducted to ensure quality. Test specifications are generated by test designers who read carefully the specifications written in natural language and extract tuples of "test conditions (including parameters) ⇒ expected values" from each sentence in the specifications. The above extraction process is done manually, and since specifications written in natural language contain errors and ambiguities, it requires a huge number of man-hours to correct and check them. According to the author's research, a 600-page specification requires about 700 hours.

In order to solve the above problem, this research promotes the completion of errors and ambiguities that occur in specifications written in natural language by converting each sentence in the specification into a formal description.

In order to visually express the structure of each sentence, we developed a tool to visualize the logical structure of the composition of sentences in a paragraph based on the logical inclusion relation of "condition ⇒ expected value."

Keywords: Discourse relation, Discourse analysis

1. はじめに

システムを開発する際には、システムの最終品質を担保するためにそのシステムに対してシステム試験を実施する。システム試験を実施するためにはテスト仕様書を必要とするが、テスト仕様書を作成するためには、テスト設計者が自然言語で記載されたシステム仕様書を熟読し、記述された各文からパラメータを含んだテスト条件と、その期待値のタプルを手作業で抽出して作成を行う。しかしながら、テスト仕様書のもととなっているシステム仕様書には、自然言語で記載されているために曖昧さや、記述漏れが含まれている。そのようなシステム仕様書をもとにするため、テスト設計者が解釈を誤ってしまう事態に陥ることもあり、最終的なテスト仕様書の成果物としてシステム仕様書の意図とは異なるテストケースを作成してしまうことがある。また、テストケースの作成プロセスが暗黙的であるがゆえに設計者の技量に依存してしまうことや、設計の根拠が不明瞭となりレビューが困難となる課題も存

在する。このような一連の作業工程には、修正や確認の為に600ページの仕様書で約700時間という膨大な時間や工数が必要となっていた。

先行研究[1]では、それらの課題を解決するために自然言語で記載されている文を形式的な記述に変換し、テストケースを自動生成するための研究が行われていた。しかし、変換プロセスが単文ごとに行われているために文間の関係性の把握ができなくなってしまう、テストケース作成に必要な“テスト条件⇒期待値”のタプルを抜き出したとしても、事前条件や事後条件を把握できないといった課題が存在する。

本研究では、“テスト条件⇒期待値”の包含関係を中心としてパラグラフ内の論理構造を文間の類似性により算出し、その構造を可視化することで、文間の関係性が失われてしまうといった課題の解決を図った。

2. アプローチ

本研究では、パラグラフに着目してアプローチを行った。

^{†1} 九州工業大学
Kyushu Institute of Technology.

具体的には、パラグラフとは前提として、最低限1つの主張が含まれていると考えており、ここでの主張とは包含関係を意味する命題「 $P \rightarrow Q$ 」を表している。例えば、話題沸騰ポット[2]、第6章エラー検知における記載内容において以下のような記述がある。文頭をわかりやすくするために数字を割り当てている。「①この沸騰ポットでは、ヒータが故障/誤動作した場合のエラーを検知します。②これは、ヒータの異常過熱による火災等の危険を回避するためです。③サーミスタの故障による異常過熱が発生した場合は、ソフトウェアではエラーを検知できないため、ハードウェア的に断線することにより安全を確保するようになっています。④水位センサ、蓋センサ、及びポンプの故障/誤動作については、今回の設計では扱わないものとします。」③がこのパラグラフにおける主張とする文であり、①や②は主張に至る経緯を④は補足説明を行っている。

文間の関係を同定する既存研究として、横山ら[3]は機械学習アルゴリズムである Support Vector Machine を用いた談話構造解析を行っているが、教師あり学習での解析を行うためにはじめに談話構造を付与したコーパスが必要となる。それに対し斎藤ら[4]は、タグがつけられたコーパスを必要としない代わりに大量のテキストデータからの統計的なデータに基づいて接続関係を推定している。しかしながらどちらの研究においても大量のデータを必要としている。本研究ではパラグラフから得ることが可能な情報と、主張に着目しパラグラフ内の論理構造を同定するというコンセプトのもとパラグラフ内構成文の論理構造を推定する。

「 $P \rightarrow Q$ 」が含まれているのであれば、パラグラフ自体もそもある一つの話題に対して述べた文の集まりであるということから、命題「 $P \rightarrow Q$ 」を中心としてパラグラフ全体の論理構造を把握できるのではないかという仮説を立てた。上記仮説検証のための予備調査として、情報処理学会の電子図書館より論文誌(54巻~61巻)のアブストラクトを対象として748編(4951文)に対して調査を実施した。748編の論文の中に包含の文が出現する事例数は、312編であった。また、接続詞として順接の関係により包含が出現する事例数としては4編であった。

そのような仮説のもと、パラグラフ全体の論理構想を把握するためには、主張の蓋然性を高める文の発見だと考え、仮説の検証・解決のために、図1のようなプロセスを提案する。また、以降提案プロセス達成のための過程(赤文字部分)について述べる。

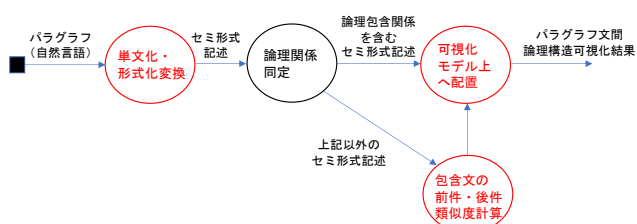


図1 提案プロセス

2.1 単文化・形式化変換

文章から欠落情報の抽出を行うために、自然言語で記載されている文を「関係語(主体, 対象, 制約)」のような命題プリミティブと呼ぶ単位で表現するセミ形式記述[1]に変換する。セミ形式記述に変換した際、文の中で主体や対象の表記漏れが存在している場合には、該当する漏れの箇所を「??」で表現することにより、レビューを行う際には、記述漏れの発見および漏れの補完を促すことができる。

例えば、「設定温度を100度に設定する。」という文をセミ形式化すると、「設定する(??, 設定温度, 100度)」と形式的に表現することができる。この際に、「??」に該当する箇所が表記漏れであることを示しており、例文における関係語の主体が欠落していることが分かる。このような欠落情報をセミ形式記述変換後にレビューを実施することでレビュー者に対して補完を促すことができ、元の文章に含まれていた曖昧性を排除することが可能となる。

また命題プリミティブ同士は論理演算子(Not「!」、And「&」、Or「|」、Imly「->」)でお互いを接続することで厳密な関係性を表記することが可能となっている。

本研究において着目している「Imly」関係(包含部分)を決定するためのアルゴリズムとしてはルールベースで行っている。具体的なルールの代表例としていくつか挙げると、形態素解析後の得られる文節の末尾が「場合」、「時」、「中」で終わる場合である。

また、ルールベースで二項演算子の決定を行っているため、ルールを追加することで予備調査において抽出されなかった包含文を抽出することができるように考えている。

2.2 包含文の前件・後件類似度計算

セミ形式記述へ変換することで、パラグラフ内に存在している包含関係となっている文を発見することが可能であるが、その主張の蓋然性を高める文を見つけ出すために、文間の「同等」の関係に着目した。理由としては、パラグラフ内で主張に至る経緯やその補足説明を行うにあたり、類似する単語や、似たような表現を用いて記述すると考えたからである。宮部ら[5]を参考にし、同等関係の特定を行うにあたり、表層的に類似している文をパラグラフ内から見つけるために2つのセミ形式記述間でのコサイン類似度での算出を行った。また、単純に単語の出現頻度でコサイン類似度を算出する場合、文中であまり重要でない単語であっても、重要な単語であってもすべての単語が等しい形で扱うため、2つのセミ形式記述間での適切な類似度を算出することができないという問題がある。そのため対象とするパラグラフ内の単語において、単語の重要度を測るためにtf-idf[6]を用いて単語の重要度を考慮したのちに、コサイン類似度で類似性の算出を行った。2つのセミ形式記述S1, S2のコサイン類似度は、tf-idfで重み付けした単語ベクトルw1, w2を用いて(1)式で求めることができる。

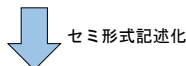
$$\cos(S1, S2) = \frac{\overline{w1} \times \overline{w2}}{|\overline{w1}| \times |\overline{w2}|} \quad (1)$$

今回 tf-idf を用いて重み付けした単語ベクトルを算出する際には、セミ形式記述に変換されて得られる「関係語、主体、対象、制約」より形態素解析器 Juman++[7]を用いて日本語を形態素解析したのちに、品詞大分類においてカテゴリが「名詞」となる単語の正規化代表表記の抽出を行った。抽出を行った理由としては、命題プリミティブに含まれる個々の要素では、自然言語の文より抽出を行うために、記載内容が統一されていないため、類似する単語や似た表現として扱うことができないためである。

セミ形式記述を用いた具体的な例を図 2 に示す。図 2 には自然言語で記述された文章とセミ形式記述へ変換した際の結果を示す。赤文字で記載している内容に関しては、「郵便局」というキーワードでのコサイン類似度を算出したいが、それぞれのセミ形式記述内では「郵便局の前」と「郵便局に」で完全一致しないといった問題がある。

自然言語の文章

太郎の靴には赤土がついている。
郵便局の前は赤土で、近所に他に赤土の場所はない。
太郎は郵便局に行ったのだ。



セミ形式記述

突く(赤土, ??, 太郎の靴には).
is(??, 郵便局の前) → is(赤土の場所, 無い, 赤土で, 近所に, 他に).
行く(太郎, ??, 郵便局に).

図 2 セミ形式記述の例

2.3 可視化モデル上へ配置

2.2 節で得られる文同士の類似度結果を用いて、パラグラフ内の構造を視覚的に把握することを可能にするために、本研究では、Toulmin's Model[8]を参照し可視化モデルの実現を行った。Toulmin's Model とは、一貫性のある議論を組み立てるための枠組みを提供し、議論における情報抽出漏れの検知や構造的かつ視覚的に関係性を可視化することのできる議論フレームワークである。また Toulmin's Model を用いた簡単な議論を図 3 に示す。

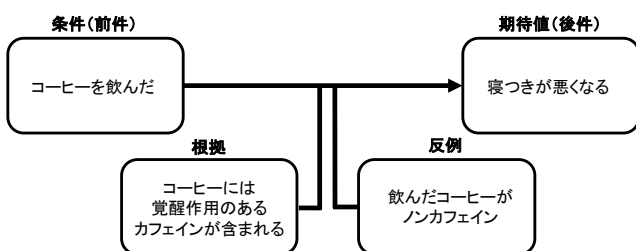


図 3 Toulmin's Model の使用例

主な構成エリアを、本研究において着目している包含関係である命題「P (以降、前件と呼ぶ) → Q (以降、後件と呼ぶ)」を表示するエリア、前件のみに類似性のあるエリア、後件のみに類似性のあるエリア、共通して類似性のあるエリア、前件・後件どちらにも類似性がないエリアの 5 つで構成している。可視化のモデル例を図 4 に示す。このような可視化モデルを設計した理由としては、検証段階において 2.2 節で説明した類似度計算で類似度を算出した結果、以下の 4 パターンに別れたためである。

● 前件のみに類似性が算出される
● 後件のみに類似性が算出される
● 前件・後件どちらにも共通して類似性が算出される
● 前件・後件どちらにも類似性が算出されない

また、今回 Toulmin's Model 内に存在する要素の「反例」を省いた理由として、現段階では 2.2 節の類似度から根拠であるかまたは反例であるかの判別ができないため、個々の要素に対してサポートしている文として可視化するためである。前件・後件どちらにも共通して類似性が算出される場合については、よりどちら側に関してサポートされているかを視覚的に分かりやすくすることを期待して、類似度に応じて左右へ位置を移動させて配置させている。

可視化をする際には、セミ形式記述をそのまま用いるのではなく、論理関係の把握支援のための命題ネットワーク[1]を活用した。まず命題ネットワークについて説明した後に、それぞれのエリアの役割について説明していく。

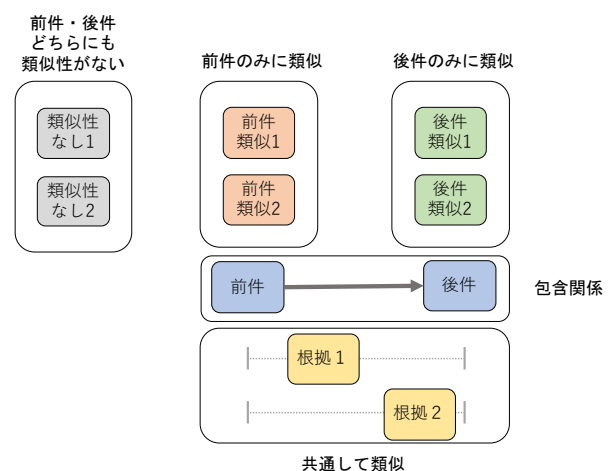


図 4 可視化モデル

2.3.1 命題ネットワーク

セミ形式記述はテキストベースで記述するものであるが、文内における構造が複雑化した際には、構造の把握が難しくなる。そこで命題プリミティブ間の論理関係を図式化するために命題ネットワークを用いて可視化を行う。図 5 に命題ネットワークでの表現方法について示す。セミ形式記述における「関係語」を中心として、左側に「主体」、右側に「対象」、下側に「制約」を配置して可視化を行う。

図 6 に命題ネットワークで表現した例を示す。

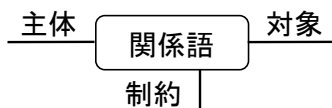


図 5 命題ネットワークでの表現方法

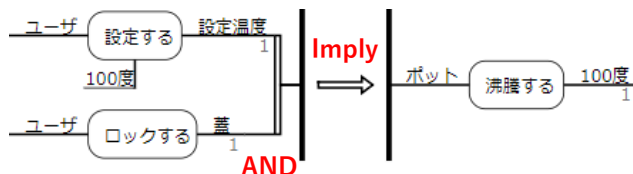


図 6 命題ネットワークの例

2.3.2 前件あるいは後件のみに類似性のあるエリア

2.2 節において算出される類似度により、前件のみに類似性が算出されたものについては、包含関係の上部に存在するオレンジ色で示されている前件のみに類似のエリアへ命題ネットワークの配置を行う。また、後件のみに類似性が算出されたものについては、包含関係上部に存在する緑色で示されている後件のみに類似のエリアへ命題ネットワークの配置を行う。どちらのエリアにおいても、複数個類似するものが存在する場合には、縦一列に連ねて配置する。前件と後件それぞれに対して命題ネットワークの配置を分ける理由として、事前条件や事後条件に対して言及されていない場合に、視覚的に把握することを可能にするためである。後件のみに類似している際の出力例を図 7 に示す。

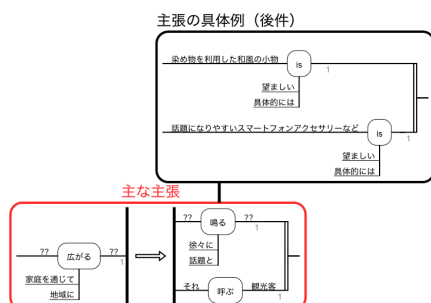


図 7 後件のみに類似している出力例

2.3.3 前件・後件に共通して類似性のあるエリア

2.2 節において算出される類似度が、前件・後件どちらにも共通して類似しているものに関しては、包含関係の下部に存在している黄色のエリアへの配置を行う。

また、配置の際には前件・後件のそれぞれに対して類似していることから、前件・後件のどちらにより類似しているサポートしている文であるかをひと目で把握することができるよう、類似度に応じて左右へ移動させて表示を行うようにしている。例えば前件より後件の類似性が高ければ、図 4 にける根拠 2 のように後件側に寄せて配置を行うようにしている。共通して類似している際の出力例を図 8 に示す。今回前件より後件の要素への類似性が高いため

後件である右側へ配置されている。

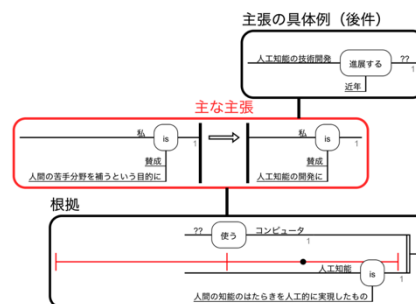


図 8 共通して類似している要素の表示例

2.3.4 前件・後件どちらにも類似性のないエリア

2.2 節において算出される類似性において、包含関係の前件と後件どちらにも類似性がないと算出された場合は、図 4 における灰色で示されている箇所へ配置を行い、直接的には包含関係のサポートをしていない文として、配置を行う。前件・後件どちらにも類似性の無い要素が配置されている際の出力例を図 9 に示す。

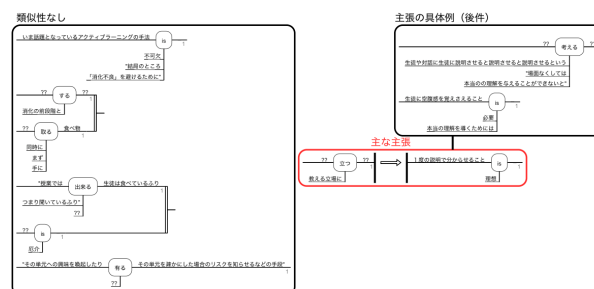


図 9 可視化の出力例

3. 検証

今回実際にシステム仕様書を用いての検証を実施する前に、提案するアプローチで文間の論理構造を同定することが可能であるかを検証するために、情報処理学会の電子図書館より論文誌に掲載されている論文よりアブストラクトの抜粋を行った 29 件に対して検証実験を実施した。今回論文のアブストラクトを検証対象として選定した理由としては 2 つある。1 つ目は、原則としてアブストラクトは 1 つか 2 つ程度のパラグラフで構成されることが多い。2 つ目は、アブストラクトは記述している論文全体の論理を要約し主張している文章であることから、そもそも論理的に記述されているため、パラグラフ内構成文の論理構造を可視化する題材として最適であると考えたためである。

確認項目として、課題解決に対するアプローチの実現可能性と、類似度計算で用いたコサイン類似度において、パラグラフ内の文間構造を同定するための情報としての活用可能性の 2 項目である。

検証方法としては対象とした論文のアブストラクト 29 件に対して、2.2 節で得られる類似度をもとに、2.3 節での可

視化モデル上に配置を行った。その後、筆者が可視化され構造化されたものと、もとの文章を照らし合わせてパターン分けを行い、それぞれの特徴から確認項目についての検証を実施した。検証により得られた結果のパターン内容とその件数を表 1 に示す。

パターン分けする際に用いた「構成要素」という概念について説明する。本研究において構成要素を、一文以上が集まって一つの「役割」を果たしているものとしている。「役割」というのは、今回対象として扱った論文のアブストラクトにおいて、対象としている研究分野の現状を記述するような内容の「前置き」の役割、どのような研究課題が存在しているかの「問題提起」の役割、実施している研究の内容を紹介している「研究内容紹介」の役割、研究により得られた結果を簡単に報告している「研究結果報告」の役割、結果より得られる考察を述べる「主張」の役割、以上の 5 つの役割が存在していた。

表 1 パターン内容と件数

パターン内容	件数(件)
包含関係が含まれる構成要素が、 2 文以上から構成	15
包含関係が含まれる構成要素が、 1 文のみで構成	4
説明過程を経て最終的に主張を述べる	3
包含関係に指示語が含まれる	4
セミ形式記述の変換による 包含関係の誤り	3

以降それぞれのパターンについて説明していく。赤文字で記載されている文はセミ形式記述に変換した際に包含関係が含まれて変換された文である。

4. 検証結果の概要

4.1 包含関係が含まれる構成要素が、2 文以上から構成

論文のアブストラクト[9]より、引用した文章とその構成要素を検証段階において割り当てたものを図 10 に示す。また得られた類似度より可視化したものを図 11 に示す。図 10 のように包含関係が含まれる構成要素「前置き」が 2 文以上で構成されている。

前置き	1 ソースコード中のコードクローンの存在は、ソフトウェアの保守性を悪化させる1つの要因であるといわれている。
前置き	2 しかしその一方で、コードクローンに対して同時に変更が行われることは少なく、コードクローンの存在が問題を引き起こすことは少ないという調査報告もある。
前置き	3 また、コードクローンはソフトウェア開発に有用であると報告している研究もある。
前置き	4 この論文では、コードクローンメトリクスとテストケース数や検出バグ数といったプロジェクトメトリクス間の相関分析の結果を報告する。
前置き	5 この分析対象は9つのウェブシステムであり、同一の仕様に基づいて異なる組織によって開発された。
前置き	6 つまり、これらは機能が同一であり実装が異なるシステムである。
前置き	7 この9つのシステムを対象とすることで、実装の違いとプロジェクトメトリクス間の関係を調査できる。
前置き	8 調査の結果、コードクローンが多く存在しているプロジェクトほど実装の速度が遅いことが分かった。
前置き	9 また、コードクローンの存在によりバグの発見が遅れてしまう傾向にあることが分かった。

図 10 引用文章と構成要素の割り当て

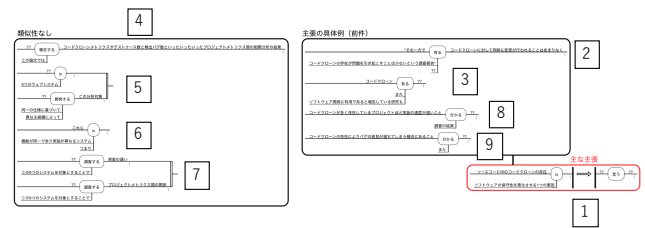


図 11 類似度結果により可視化した図

4.2 包含関係が含まれる構成要素が、1 文のみで構成

論文のアブストラクト[10]より、引用した文章とその構成要素を検証段階において割り当てたものを図 12 に示す。また得られた類似度より可視化したものを図 13 に示す。図 12 より包含関係が含まれる構成要素「前置き」が 1 文のみで構成されている。

前置き	1 組織におけるWebアクセスの利用には多くの脅威が存在するため、悪性Webサイト情報のブラックリストが利用される。
前置き	2 悪性Webサイト情報には登録理由やレビュー情報などが付与されることがあるが、CSIRTのインシデントレスポンスにおいて合理的な判断を行うために適した情報提供しているとはいえない。
前置き	3 本稿では、インシデントレスポンスの「トリガー」[対応策]「業務」において、既存のブラックリストを活用した業務における悪性Webサイト情報の分類を可能とするための「所有者」「コンテンツ」「現在の状態」「最終確認」という4項目による悪性Webサイト情報の分類を提案する。
前置き	4 さらに、公開されている38種類のブラックリストの仕様について調査し、この分類に相当する情報がほとんど含まれていないことを示す。
前置き	5 また、実際にブラックリストに一致してインシデント判定が求められた400件の悪性Webサイト情報について手動で調査を行い、一定数のWebサイト情報が外部からの調査によって定義に従った分類ができることを示す。
前置き	6 この分類を活用することで速やかな対応ができることをケーススタディとして紹介し、提案分類が有効に働くことを示す。

図 12 引用文章と構成要素の割り当て

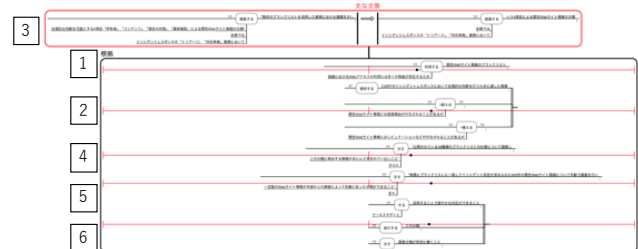


図 13 類似度結果により可視化した図

4.3 説明過程を経て最終的に主張を述べる

論文のアブストラクト[11]より、引用した文章とその構成要素を検証段階において割り当てたものを図 14 に示す。また得られた類似度より可視化したものを図 15 に示す。図 14 より包含関係が含まれる構成要素「前置き」が 2 文以上構成されている。

前置き	1 近年IoT機器を狙ったサイバー攻撃が高度化・広範化の一途をたどっており、IoT機器の利用状況および設定状況、セキュリティ対策の実態をインターネットから観測するための広域ネットワークスキャン技術および機器種別・機器名の特定技術の重要性が高まっている。
前置き	2 本研究では、Webサービスが稼働するネットワーク機器を対象に、WebUIの画面的な特徴に基づいて効率的にIoT機器のホストを検出する手法の構築を試みる。
前置き	3 広域ネットワークスキャンにより取得したWebUIのスクリーンショット画像に対してクラスタリングを行い、多くのIoT機器のWebUI画像を含むクラスターを検出する。
前置き	4 実験の結果、本手法により、同一または類似する機器のWebUI画像が同一のクラスターに凝集する傾向にあることが分かった。
前置き	5 本手法はサイバー攻撃の対象となりうるIoT機器の早期発見と対策の実施を行ううえで有効といえる。

図 14 引用文章と構成要素の割り当て

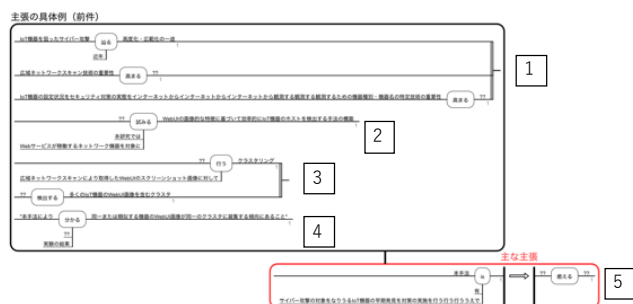


図 15 類似度結果により可視化した図

4.4 包含関係に指示語が含まれる

今回類似度を算出する際には、照応解析を行っての指示語や代名詞の補完を行っていないため、図 16 の例における命題ネットワークのような前件に指示語が含まれる場合には類似性を算出できないために、前件あるいは後件に類似するものが偏って算出されてしまうものを表す。

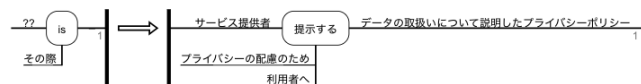


図 16 指示語を含む包含関係

4.5 セミ形式記述の変換による包含関係の誤り

セミ形式記述に変換後得られる包含関係において、前件後件似たような構成になっているなど「条件→期待値」として不都合な変換結果が得られた際にセミ形式記述の変換による包含関係の誤りとして分類した。不都合と判断した例を図 17 に命題ネットワークで示す。

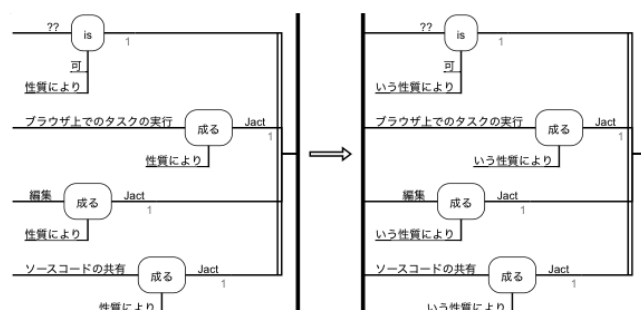


図 17 セミ形式

5. 考察

5.1 期待する命題を期待する位置にコサイン類似度で配置可能か

提案するアプローチの実現可能性を確認するために、情報処理学会の電子図書館より論文誌に掲載されている論文のアブストラクトから抜粋したパラグラフで検証を行った結果、包含関係が含まれている構成要素の文が 2 文以上の場合、同じ構成要素の文を前件または後件をサポートしている文として取得することが可能であると考えられる。

しかしながら、前件・後件どちらにも類似度が存在する共通類似要素に関しては、直接的に包含関係をサポートして

いる文ではないものがある場合があるので、その際には抽出数が多くなり構造の把握が難しくなることがある。

以上の結果を踏まえると、コサイン類似度での類似度判定では、表記揺れには対応することができていないが、前件または後件をサポートする文を可視化で近くに配置するための判断材料としては活用できることがわかった。

5.2 パラグラフ内の論理関係が分かりやすくなるか

評価結果から得られるパターンのうち、「包含関係が含まれる構成要素が、2 文以上から構成」のパターンにおいては、同じ構成要素の文を抽出することができており、包含文に直接関係する文の抽出ができました。前件・後件で個別に類似度を算出していたことにより、より強く主張をサポートしている文が分かることで論理展開を視覚的に把握することができていると考える。

ただし、現状の問題点として 2 点あり、1 つ目は共通類似項目が多く抽出された場合に、もとの文章との違いとしては主張が分かるだけであまり有用性を感じないのが現状である。2 つ目は「前件・後件どちらにも類似性のないエリア」に抽出される文については直接的ではなく間接的に主張をサポートしているような文が含まれることがあり、コサイン類似度だけでの抽出の限界も感じられた。

5.3 自然言語の曖昧さへの対応

現状例えば「ねこ」と「猫」のように単語の記述に関して表記揺れが生じている場合、類似性を算出することができない。また、主張を抽象化した内容で言い換えているような文は現状だと類似性の無いものとして抽出されてしまう。そこで単語の分類や分類されている概念の関係などが付与されている概念辞書を活用してこれらへの対応をしていきたいと考えている。

6. まとめ

本研究では論文のアブストラクトを対象として包含関係を中心として、パラグラフ内構成文の論理構造を可視化することが可能であるかの検証を行った。検証結果としては、包含関係を中心として「包含関係が含まれる構成要素が、2 文以上から構成」のパターン (29 件中 15 件の約 52%) に対してパラグラフ内構成文の論理構造を可視化することができる。ただコサイン類似度では、「前件・後件どちらにも共通して類似している」文だと判断された場合に、直接的には包含関係の文をサポートしていないような文も含まれるため、コサイン類似度だけの判断には限界が感じられた。

今後の予定としては、対象としているシステム仕様書においても構造化することが可能であるかの検証を実施していきたいと考えている。論文のアブストラクトにおいて、包

含関係がセミ形式記述変換により得られないが順接の役割を果たす接続詞により包含関係が出現する事例数が、169件中4件と少ないが、逆説の役割を果たす接続詞の事例数が169件中103件と多かった。これは問題提起からの解決案の提案をする際にこのような書き方が多かったためである。一方今後対象としていく予定の仕様書では、機能の動作とその条件が記載されることが予想されるため順接の役割を果たす接続詞を用いての文と文の接続が論文のアブストラクトに比べると増加すると予想している。そのためも

との文章において、接続詞でつながっている場合から包含関係を生成し今回提案するアルゴリズムに組み込んでいくことも検討していかなければならない。

また、類似度判定においては現在コサイン類似度のみしか判断材料として用いていないため、概念辞書やもとの文章に含まれる接続詞等の情報を活用し、より精度が高い構造化が可能となるように努めていきたい。

参考文献

- [1] 青山裕介, 黒岩丈瑠, 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* 61.3 (2020): 521-534.
- [2] 話題沸騰ポット(GOMA-1015 型) 要求仕様書[第 6 版], SESSAME, 2004.
- [3] 横山憲司, 難波英嗣, and 奥村学. "Support Vector Machine を用いた談話構造解析." *情報処理学会研究報告自然言語処理(NL)* 2003.23 (2002-NL-154) (2003): 193-200.
- [4] 齋藤真実, 山本和英, and 関根聡. "大規模テキストを用いた 2 文間接続関係の同定." *言語処理学会第 12 回年次大会, March 2006* (2006).
- [5] 宮部泰成, 高村大也, and 奥村学. "文書横断文間関係の特定." *言語処理学会第 12 回年次大会発表論文集* (2006): 496-499.
- [6] 難波 英嗣 “テキスト間の類似度の測定” *情報の科学と技術* 70 巻 7 号, 373~375 (2020)
- [7] 京都大学黒橋・河原研究室, “日本語形態素解析システム JUMAN++”, <http://nlp.ist.i.kyoto-u.ac.jp/?JUMAN%2B%2B>
- [8] Kirschner,P.A.,Buckingham-Shum,S.J.,and Carr,C.S.(2012): “Visualizing argumentation: Software tools for collaborative and educational sense-making”, Springer Science & Business Media.
- [9] 肥後芳樹, et al. “同一仕様プロジェクトを利用したコードクローンの影響調査.” *情報処理学会論文誌* 59.12 (2018): 2191-2200.
- [10] 羽田大樹, 後藤厚宏. “CSIRT のための Web ブラックリストの分類の提案.” *情報処理学会論文誌* 59.9 (2018): 1596-1609.
- [11] 藤田彬, et al. “WebUI の画像的特徴に基づく IoT 機器判別手法.” *情報処理学会論文誌* 60.3 (2019): 849-858.