

## 手がかり句を用いた特許請求項の修辞構造解析

新森昭宏† 奥村学‡ 丸川雄三‡ 岩山真\*

shinmori@lr.pi.titech.ac.jp, {oku, maru, iwayama}@pi.titech.ac.jp

† 東京工業大学 総合理工学研究科 / インテック・ウェブ・アンド・ゲノム・インフォマティクス

‡ 東京工業大学 精密工学研究所

\* 東京工業大学 精密工学研究所 / 日立製作所

特許の内容を記述する「特許明細書」において最も重要な箇所は、「特許請求項」(クレーム)の部分である。特許請求項は、1文で発明内容を記述するという制約と、その独特の記述形式により、専門家以外の人にとっては極めて読みにくいものになっているのが通例である。我々は、特許請求項の可読性を向上させることを目的とした、言語処理技術の研究を行っている。特許請求項の記述内容をわかりやすく提示するためにはまず、その構造を解明することが必要である。特許請求項の記述ではいくつかの定型的な表現が多用されており、それらが請求項の構造解明の手がかりとなることがわかった。こうした手がかり句を利用して特許請求項の修辞構造を解析する手法について提案し、その現時点での評価結果について報告する。

## Rhetorical Structure Analysis of Japanese Patent Claims using Cue Phrases

Akihiro Shinmori† Manabu Okumura‡ Yuzo Marukawa‡ Makoto Iwayama\*

† Department of Computational Intelligence and Systems Science,  
Tokyo Institute of Technology / INTEC Web and Genome Informatics Corp.

‡ Precision and Intelligence Laboratory, Tokyo Institute of Technology

\* Precision and Intelligence Laboratory, Tokyo Institute of Technology / Hitachi, Ltd.

The most important part of patent specification documents is patent claims. It is common that claims written in Japanese are described in one sentence with peculiar style and are difficult to understand for ordinary people. We are investigating NLP technologies to improve readability of patent claims. It is necessary to analyze the structure of patent claims so that they can be presented in readable ways. We found that there are several typical phrases used in claim descriptions and that they can be used as clues to analyze claim structure. We propose a method to analyze rhetorical structure of patent claims by using these cue phrases and report the result of its evaluation at this point.

## 1. はじめに

かつて特許は、機械・製薬・化学など特定分野の企業において、知的財産権担当者や研究者が主に関わるものであった。しかし、ビジネスやサービスの方法を権利の対象とする「ビジネスモデル特許」の出現や、コンピュータプログラムを対象とした「ソフトウェア特許」の認知により、広い範囲の企業関係者が特許に関わらざるを得ない状況が生まれている。大学においても、研究成果や技術移転のコアとしての特許の重要性が認識されるようになった。

特許は一種の法的文書であり、その内容を記述する「特許明細書」は、独特の記述形式を持っている。最も重要な箇所である「特許請求項」(クレーム)は、1文で発明内容を記述するという制約と、独特の記述スタイルにより、専門家以外の人にとっては極めて読みにくいものになっているのが通例である。

特許行政は、いち早く電子化が推進された分野であり、過去に出願された特許明細書の電子データは膨大に蓄積されている。さらに、年間40万件以上と言われる新規出願に伴い、データ量は日々増加している。こうした膨大な特許データを対象とした研究は従来、検索に関するものがほとんどであった。

我々は、特許請求項の低可読性に着目し、これを向上させることを目的とした、言語処理技術の研究を行っている[1]。

特許請求項の記述内容をわかりやすく提示するためには、その記述特性を調査し、その構造を解明することが必要である。以下では、特許請求項の記述特性についての調査結果を報告した後、特許請求項に多用される表現を手がかりとして、その構造を解析する手法について提案し、現時点におけるその評価結果について報告する。

## 2. 特許請求項の記述特性

### 2.1 一般的特性

日本語で記述された特許請求項はほとんどの場合、1文で記述されることが多い。複数の要素や処理を1文で説明しようとするため、構文が複雑になり、文の長さが長くなる。たとえば、国立

情報学研究所が主催する NTCIR3 ワークショップ[2]の特許検索タスク参加者に配布されている「特許データコレクション」のうち、サンプルデータ(59,962件)の第一請求項の長さを調査したところ、平均の文長が235文字であった。

また、特許請求項の記述においては、新聞記事や論文・報告書などとは異なり、読みやすさよりも正確性や厳密性が重視される。専門家(弁理士や知的財産権担当者)が作成することが多いため、一種の専門家コミュニティが形成されており、一定の記述スタイルが確立している。

特許出願者の立場から「どのように特許請求項を記述すべきか」を解説した文献[3]では、特許請求項の記述スタイルに関して、表-1に示すように、3つのスタイルへの類型化が行われている。ここで、各類型は排他的なものではない。たとえば、ジェプソン形式の公知部分または新規部分が書き流し形式または要素列挙形式で記述されることは多い。

表-1 特許請求項の記述スタイルの類型

スタイル	説明
書き流し形式	・「...し、...し、...した、...」のように、処理内容を「書き流す」形で記述する。 ・方法の発明で用いられることが多いが、物の発明において用いられることもある(=プロダクト・パイ・プロセス・クレーム)。
要素列挙形式 <sup>1</sup>	・「...と、...と、...とからなる、...」のように、構成要素を列挙する形で記述する形式。 ・主に、物の発明で用いられる。
ジェプソン(Jepson)形式	・「...において、...を特徴とする、...」 「...であって、...を特徴とする、...」のように、最初に、公知部分(既に知られている内容)または前提条件を述べた上で、新規部分(この発明の特徴となる部分)または本論部分を記述する形式。

<sup>1</sup> 文献[3]では、「要件列挙形式」という用語が使用されているが、「構成要素を列挙する形式」という意味合いを明確にするため、本稿では「要素列挙形式」という用語を使用している。

## 2.2 言語処理の観点から見た特性

### (1) 明示的に挿入された改行

長い特許請求項の場合、下例のように、改行が明示的に挿入されていることがある。

<例>(特願平8 - 182670より引用) [145文字]

【請求項1】原稿が載置される原稿台と、<改行>  
この原稿台に対して主走査方向に移動する走査光学手段と、<改行>

この走査光学手段上に配置され原稿を副走査方向に照明する照明手段と、を備えた画像読取装置において、<改行>

前記照明手段は、前記走査光学手段に対して走査移動平面に略平行に回動自在に取付けられることを特徴とする画像読取装置。<改行>

(注：上記において、<改行>と明記した箇所に、改行コードが挿入されている。なお、表示レイアウトのためのマークアップ(<BR>)は削除している)

こうした改行は、特許明細書作成者(弁理士等)が自分で理解しやすいようにするために、記述のまとまりごとに挿入したものと推測される。

言語処理の観点からは、こうした改行の近傍に頻出する表現を、改行が挿入されていない請求項の解析のための手がかり句(cue phrase)として利用するということが考えられる。

NTCIR3の「特許データコレクション」のサンプルデータを調査した結果、第一請求項の記述中に改行が挿入されているものの割合は、約46%であった。改行直前の形態素を3つ分抽出して集計したところ、表-2のような結果が得られた。第1位から第4位までのパターンで、約85%のカバー率となっている。

表-2 明示的に改行が挿入された請求項における、改行直前のパターン

順位	パターン	割合	(1位からの累積)
1	(名詞 記号)と、	46.4%	46.4%
2	(名詞 記号)において、	15.8%	62.1%
3	(動詞連用形)、	15.6%	77.7%
4	(名詞 記号)であって、	6.9%	84.6%

(2) 記述末尾の「名詞まとまり」直前の修飾表現  
特許請求項は通常、名詞句であり、その末尾は「 の 」の形式をとることが多い。ここで、 と は、名詞または複合名詞であるが、特

許の場合は、造語や辞書に登録されていない未知語であることが多い。そして、末尾の「 の 」の直前には、「を特徴とした」のような定型的な修飾表現が付けられていることが多い。こうした表現も手がかり句として利用できる。

### (3) 請求項解析のための手がかり句

以上の予備調査結果と、表-1のような類型化を考慮した目視分析等により、表-3に示すような手がかり句を収集した。

なお、表-3では、手がかり句の表記に、Perlの正規表現記法を使用している。また、後述する修辞構造解析アルゴリズムを説明するために、手がかり句ごとに、トークン名を割り当てている。

## 3. 修辞構造理論の特許請求項解析への応用

日本語の長い文の構造解析において、シソーラスとダイナミックプログラミングを用いて文中の並列構造を検出するアルゴリズムと、それに基づく係り受け解析プログラムが開発され、科学文献や新聞記事の解析において一定の精度が達成された[4]。

しかし、特許請求項の記述は複雑であり、かつ、1つの事項を説明した後で、それを用いて別の事項を説明するという、連鎖的な記述が多くみられるため、このアルゴリズムが必ずしもうまく動作しない。たとえば、前述例の「特願平8 - 182670」の場合、「～原稿台と」・「～走査光学手段と」・「～照明手段と」の3つの部分を並列構造としなければならないが、下線・二重下線で示すような連鎖の存在により、この並列構造がうまく検出されない。この場合はむしろ、「(名詞)と、」を手がかりとした方が、並列構造を検出できる。

特許請求項は複数の要素や処理の説明を1文の中に詰め込んだ形で記述されている。つまり、相互に関係を持つ複数の文から成り立っていると考えるべきである。そこで我々は、複数の文から構成された談話の構造を解析するための理論である修辞構造理論(RST: Rhetorical Structure Theory) [5]を適用することを考えた。修辞構造理論は、1980年代に提唱され、自動要約[6]や自動レイアウトなどに応用されている。修辞構造を対話型で編集するためのツールとして、Tel/Tkに

表 - 3 特許請求項解析のための手がかり句

トークン名	手がかり句	出現箇所
JEPSON_CUE	に(お 於)いて、 であって、 にあたり、 に当(た)?り、	ジェブソン形式で、公知部分と新規部分、前提部分と本論部分を区切る箇所
FEATURE_CUE	を特徴と(した する)(、)?	末尾の名詞まとまりの直前
COMPOSE_CUE	を(具 備 そなえ(た る)(、)? を具備(した する)(、)? (で から)構成され(た ている)(、)? を有(する した)(、)? を包含(する した)(、)? を含(む んだ)(、)? からなる(、)? を設け(、)?	構成を明示する箇所
NOUN POSTP_TO PUNCT_TOUTEN	「(名詞)と、」の並び	要素列挙形式で、構成要素(名詞句)を並列させる箇所
VERB_RENYOU PUNCT_TOUTEN	「(動詞連用形)、」の並び+「(動詞基本形)+(名詞)」	書き流し形式で、処理(述語)を並列させる箇所

表 - 4 特許請求項用の修辞関係とその検出方法

区分	修辞関係	説明	検出方法
多核	PROCEDURE	書き流し形式の場合 例： [~し、][~し、][~する]XXX 	「書き流し形式で、処理(述語)を並列させる箇所を示す手がかり句」が存在するとき
	COMPONENT	要素列挙形式の場合 例： [~と、][~と、][~と]を備えた XXX 	「要素列挙形式で、構成要素(名詞句)を並列させる手がかり句」と、「構成を明示する手がかり句」が存在するとき
単核	ELABORATION	詳述 例： [XXX を YYY した][ZZZ の AAA] 	「ZZZ の AAA」等の形式をした「名詞まとまり」が後の部分に存在し、その直前には他の修辞関係に組み込まれていない部分が存在するとき
	FEATURE	特徴付け 例： [XXX である YYY][を特徴とする] 	「特徴を明示する手がかり句」が存在するとき
	PRECONDITION	公知部分、または前提部分 例： [XXX であって、][YYY した ZZZ] 	「ジェブソン形式の公知部分と新規部分、前提部分と本論部分を区切る手がかり句」が存在するとき
	COMPOSE	構成対象 例： [~と、 ~と、 ~と][を備えた]XXX 	「構成を明示する手がかり句」が存在するとき

よる RSTTool [7]も開発されている。

特許請求項の修辞構造を解析するにあたり、我々は表 - 4 に示すような、修辞関係を定義した。また、同じく表 - 4 に示すように、それぞれについて、手がかり句を用いた検出方法を考案した。なお、表 - 4 において、多核(multi-nuclear)とは、関係を構成する要素群が対等である関係を指し、単核とは、重要な要素(nucleus : 核 = 主要部)と補足的な要素(satellite : 衛星 = 周辺部)とから構成される関係を指す。単核関係の説明中の矢印は、衛星から核の方向に引かれている。

修辞構造解析を行うことにより、特許請求項の構成要素を明確化できることが期待される。

#### 4. 手がかり句を用いた解析アルゴリズム

前述した通り、特許請求項の記述は法律文と類似した一種の制限言語と考えられる。また、前章で説明した通り、記述構造解明の手がかりとなる句がいくつか存在している。そこで、トークン切り出しの部分に工夫を加えた字句解析処理と構文解析処理から構成される解析アルゴリズムを考案した。そのポイントは、字句解析処理における以下の3つの工夫点にある：

- ・ 字句解析時に、手がかり句を検出し、手がかり句用のトークン(手がかり句トークン)を出力させるようにしていること
- ・ 字句解析時に、条件判定を行い、文脈に依存する形で形態素に対応したトークン(形態素トークン)の生成処理を行わせていること
- ・ 文法記述の複雑化を避けるため、上記の処理対象とならなかった形態素についてはすべて、同一の形態素トークン(WORD)としていること

以下に、アルゴリズムの概要を示す。

##### <アルゴリズムの概要>

###### ・ 字句解析処理

1. 特許請求項テキストを茶筌[8]で形態素解析する。
  - ・ もともと挿入されている改行コードは、そのままの状態を入力する。(茶筌には、-j オプションを使用)
2. 形態素解析結果を、トークン列に変換する。
  - ・ 手がかり句トークン

JEPSON\_CUE, FEATURE\_CUE,  
COMPOSE\_CUE

- ・ 形態素トークン

NOUN,  
POSTP\_TO, POSTP\_NO,  
VERB\_RENYOU, VERB\_KIHON,  
PUNCT\_TOUTEN  
WORD

2-1. 形態素解析結果について、(文脈に依存せず)無条件に以下の変換を行う。

- ・ 末尾からの検索を行い、JEPSON\_CUE トークンに置換する(1回だけ)。
- ・ 末尾からの検索を行い、FEATURE\_CUE トークンに置換する(1回だけ)。

2-2. 以下のような文脈依存の変換を行う。

- ・ 末尾の直前に存在する、名詞・助詞「の」・助詞「と」の連続を、NOUN, POSTP\_NO, POSTP\_TO トークンに変換する。
- ・ JEPSON\_CUE, FEATURE\_CUE の直前に存在する、名詞と助詞「の」の連続を、NOUN, POSTP\_NO トークンに変換する。

2-3. 以下のような条件判定のもとに、文脈依存の変換を行う。

- ・ JEPSON\_CUE の前方と後方をそれぞれ探索し、
  - (動詞|助動詞)基本形 + NOUN
  - COMPOSE\_CUEに対応するパターンのいずれが後に出現するかを調査する。(探索中に、「(動詞|助動詞)基本形 + NOUN」に遭遇したときは、その直前3形態素を調査し、「COMPOSE\_CUE に対応するパターン」にマッチしないかどうか調べる。マッチした場合、COMPOSE\_CUE を優先する)
- ・ (JEPSON\_CUE の前方で、「(動詞|助動詞)基本形 + NOUN」が後に出現する場合)その前方において、他の手がかり句トークンの後までの範囲に存在する「(動詞連用形)」、「及びその「(動詞・助動詞)基本形」をそれぞれ、VERB\_RENYOU, PUNCT\_TOUTEN, VERB\_KIHON トークンに変換する。
- ・ (JEPSON\_CUE の前方で、COMPOSE\_CUE に対応するパターンが後に出現する場

合)COMPOSE\_CUE トークンに変換する。

- ・ (JEPSON\_CUE の後方で、「(動詞|助動詞)基本形 + NOUN」が後に出現する場合)その前方において、他の手がかり句トークンの後までの範囲に存在する「(動詞連用形)、」及びその「(動詞|助動詞)基本形」をそれぞれ、VERB\_RENYOU, PUNCT\_TOUTEN, VERB\_KIHON トークンに変換する。
- ・ (JEPSON\_CUE の後方で、COMPOSE\_CUE に対応するパターンが後に出現する場合)COMPOSE\_CUE トークンに変換する。

2-4. 以下のような文脈依存の変換を行う。

- ・ NOUN または JEPSON\_CUE の直前に VERB\_KIHON が存在する場合、その前方で、他の手がかり句の後までの範囲に存在する「(動詞連用形)、」を、VERB\_RENYOU, PUNCT\_TOUTEN に変換する。
- ・ COMPOSE\_CUE の直前に「と(、)?」が存在するときその前方を探索し、他の手がかり句トークンの後までの範囲に存在する「(名詞)と、」(記号)と、」を、NOUN, POSTP\_TO, PUNCT\_TOUTEN に変換する。

#### 構文解析処理

3. 付録に添付した文脈自由文法により、構文解析を行う。構文解析には Bison[9]など、LALR(1)型のパーサジェネレータで生成されたパーサを使用する。

### 5. 特許請求項の修辞構造解析プログラム

前章のアルゴリズムに基づき、文脈自由文法のアクション部の処理として修辞構造木を組み上げるプログラム(rst\_claim)を実装した。

特許請求項には、記述が1つの請求項で完結するもの(独立形式請求項)と、他の請求項を参照するもの(引用形式請求項)とが存在するが、今回は対象を独立形式請求項に絞ることとした。表示レイアウトのためのマークアップ(<BR>)を削除し、記述のまとまりを区切るためにもともと挿入されている改行は保存する形で切り出したテキストを入力とした。出力は、RSTTool[7]のv2.7におけるファイルフォーマットである「.rs2 ファイ

ル」(XML ライクな形式で、修辞構造をマークアップしたファイル)とした。このことにより、rst\_claim の実行結果を視覚的に確認することができる。

図 - 1 に、修辞構造解析プログラム(rst\_claim)の位置付けを示す。

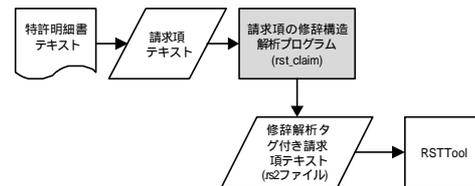


図 - 1 修辞構造解析プログラム(rst\_claim)の位置付け

以下のテキストを入力して修辞構造解析を行った結果を RSTTool で表示している様子を図 - 2 に示す。

<入力> (特開平10 - 11007より引用)

[304 文字]

各種情報を蓄積し揭示用情報として出力するホストコンピュータと、このホストコンピュータと大学構内に付設されたデータ回線網を介して接続し、前記揭示用情報を入力あるいは受信して表示し、前記揭示用情報に含まれる各種サービスの要求と任意の情報の入力および出力とを行う複数の端末とからなる大学構内掲示板サービスシステムにおいて、前記端末として各種揭示用情報を入力し利用者の要求を受付ける事務端末および図書端末と、前記利用者が使用し前記揭示用情報の取得と前記揭示用情報に含まれる各種サービスの要求と登録と予約とこれらへの入力に対応する回答とを表示し出力する利用者端末とを備えることを特徴とする大学構内掲示板サービスシステム。

### 6. 評価と考察

#### 6.1 評価

NTCIR3 の「特許データコレクション」のサンプルデータから抽出した第一請求項 59,962 件を処理したところ、55,633 件(92.8%)が受理された。処理にかかった時間は、4 時間 34 分 16 秒であった(1 件あたり、0.27 秒)(Pentium III 1GHz、メモリ 512MB、OS は Vine Linux 2.1 のマシンを使用)。

解析精度について、「プログラムによる解析結果が、人間が見て正しいものとなっているか」をすべての解析結果について客観的に評価することは現時点では不可能であるが、ランダムに選択

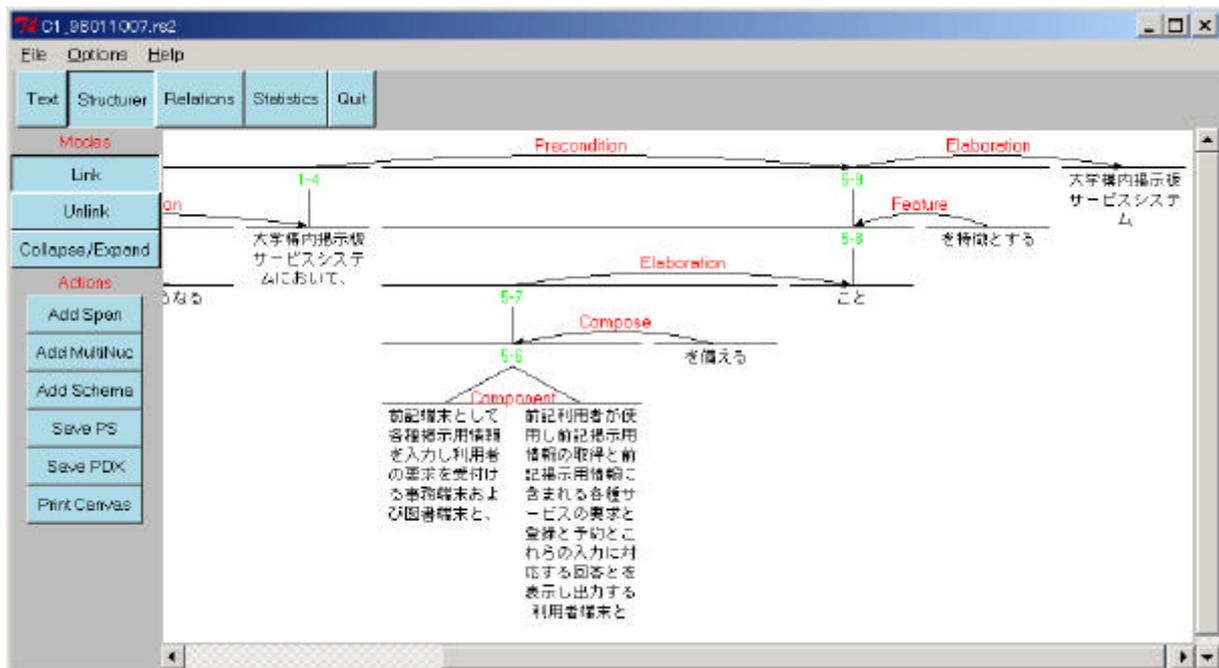


図 - 2 特許請求項の修辞構造解析結果(RSTTool v2.7 により表示)

した 20 件を目視した結果は、(以下の考察で述べる課題を除けば)概ね良好であった。

## 6.2 考察

図 2 のように表示することで、特開平 10 - 111007 の第一請求項が主張する新規性は、2 つの構成要素から成り立っているということが一目でわかるようになる。また、特開平 10 - 111007 の第一請求項にはもともと改行が挿入されていないが、4 の解析アルゴリズムにおける構文解析の結果を利用すれば、以下のように改行コードを自動挿入したテキストを生成して提示することも可能である。

### <改行コードを自動挿入した結果>

各種情報を蓄積し掲示用情報として出力するホストコンピュータと、<改行>  
このホストコンピュータと大学構内に付設されたデータ回線網を介して接続し、前記掲示用情報を入力あるいは受信して表示し、前記掲示用情報に含まれる各種サービスの要求と任意の情報の入力および出力とを行う複数の端末と<改行>  
からなる<改行>  
大学構内掲示板サービスシステム<改行>  
において、<改行>  
前記端末として各種掲示用情報を入力し利用者の要求を受付ける事務端末および図書端末と、<改行>  
前記利用者が使用し前記掲示用情報の取得と前記掲示用情報に含まれる各種サービスの要求と登録と予約とこれらの入力に対応する回答とを表示し出力する利用者端末

と<改行>  
を備える<改行>  
こと<改行>  
を特徴とする<改行>  
大学構内掲示板サービスシステム<改行>

本稿で提案しているアルゴリズムとプログラムは、特許請求項の記述構造を末端レベルまで解析するものではない。たとえば、図 - 2 において、「前記端末として各種掲示用情報を入力し利用者の要求を受付ける事務端末および図書端末と、」というまとまりは、

- ・ 前記端末として各種掲示用情報を入力し
- ・ 利用者の要求を受付ける
- ・ 事務端末および図書端末

の 3 要素に細分解されるが、本アルゴリズムとプログラムではこのレベルまでの解析は行っていない。とは言え、図 2 のレベルまで解析ができれば、後の処理は、KNP[4]などの既存の解析ツールにまかせるということも可能であると考えられる。

## 7. 今後の課題

現在、プログラムの実装をした直後であり、細部のチューニングを行っていない。今後は、アルゴリズムとプログラムのチューニングを行うと

ともに、解析実験を、NTCIR3 の「特許データコレクション」の全データに対して実施し、評価する予定である。その際、出願分野の違いによる影響がどの程度でるかどうを見極めたい。

特許請求項の修辞構造を解析することで、特許の構成要素を抽出することができる。これにより、言い換えだけでなく、複数特許間の相互関係分析などの技術への展望も開けると思われる。

## 謝辞

本研究では、NTCIR3 ワークショップで配布された「特許データコレクション」を使用しています。

## 参考文献

- [1] 新森昭宏、齋藤豪、奥村学、「特許請求項の可読性向上のための自動言い換えについての考察」、言語処理学会第 7 回年次大会併設ワークショップ「言い換え/パラフレーズの自動化に向けて」、2001 年 3 月。
- [2] 岩山真、藤井敦、高野明彦、神門典子、「特許コーパスを用いた検索タスクの提案」、情報処理学会情報学基礎研究会 No.063-007, 2001 年 7 月。
- [3] 葛西泰二、「特許明細書のクレーム作成マニュアル」、工業調査会, 1999.
- [4] 黒橋禎夫、「結構やるな、KNP」、情報処理学会誌、Vol.41, No.11, 2000.
- [5] Bill Mann, "An Introduction to Rhetorical Structure Theory (RST)", 1999. (<http://www.sil.org/~mannb/rst/rintro99.htm>)
- [6] Daniel Marcu, "The Theory and Practice of Discourse Parsing and Summarization", MIT Press. 2000.
- [7] Michael O'Donnell, "RST-Tool: An RST Analysis Tool", Proc. of the 6th European Workshop on Natural Language Generation. 1997. (<http://www.wagsoft.com>)
- [8] 松本裕治、北内啓、山下達雄、平野善隆、松田寛、高岡一馬、浅原正幸、「形態素解析システム『茶釜』 version 2.2.9 使用説明書」、奈良先端科学技術大学院大学 松本研究室、2002.
- [9] Charles Donnelly, Richard Stallman, "Bison - The YACC-compatible Parser Generator", Version 1.25, 1995.

## 付録（特許請求項解析用の文法）

```
%token JEPSON_CUE
%token FEATURE_CUE
%token COMPOSE_CUE
%token POSTP_NO POSTP_TO
%token NOUN
%token VERB_RENYOU VERB_KIHON
%token PUNCT_TOUTEN
%token WORD

%%
claim_spec:
```

```
after_jepson
| before_jepson JEPSON_CUE after_jepson
;

before_jepson:
word_noun_group
| word_noun_group COMPOSE_CUE word_noun_group
| composed word_noun_group
| processed
| processed word_noun_group
;

after_jepson:
word_noun_group
| word_noun_group COMPOSE_CUE word_noun_group
| composed word_noun_group
| processed word_noun_group
| word_noun_group FEATURE_CUE word_noun_group
| word_noun_group COMPOSE_CUE word_noun_group
FEATURE_CUE word_noun_group
| composed word_noun_group FEATURE_CUE
word_noun_group
| processed word_noun_group FEATURE_CUE
word_noun_group
| word_noun_group FEATURE_CUE word_noun_group
COMPOSE_CUE word_noun_group
;

composed:
youso_rekkyo_seq COMPOSE_CUE
;

processed:
verb_group
| shori_rekkyo_seq verb_group
;

youso_rekkyo_seq:
each_youso_complex youso_connect
| youso_rekkyo_seq each_youso_complex youso_connect
;

youso_connect:
POSTP_TO
| POSTP_TO PUNCT_TOUTEN
;

each_youso_complex:
word_noun_group
;

shori_rekkyo_seq:
each_shori_complex
| shori_rekkyo_seq each_shori_complex
;

each_shori_complex:
word_seq VERB_RENYOU PUNCT_TOUTEN
;

word_seq:
WORD
| word_seq WORD
;

word_noun_group:
noun_group
| word_seq noun_group
;

noun_group:
NOUN
| noun_group NOUN
| noun_group POSTP_NO NOUN
;

verb_group:
word_seq VERB_KIHON
;

%%
```