修 士 論 文

法令文書を対象とした
並列構造解析の精緻化

北陸先端科学技術大学院大学
情報科学研究科情報科学専攻

松山 宏樹

2012 年 3 月

修 士 論 文

法令文書を対象とした
並列構造解析の精緻化

指導教員　白井 清昭 准教授

審査委員主査　白井 清昭 准教授
審査委員　島津 明 教授
審査委員　東条 敏 教授

北陸先端科学技術大学院大学
情報科学研究科情報科学専攻

1010062 松山 宏樹

提出年月: 2012 年 2 月

概 要

自然言語処理における並列構造解析は、解決が困難な問題の一つである。その原因として、文書ドメインごとに言語的性質が異なることが挙げられる。黒橋・長尾の研究では、並列関係にある句は互いに類似していると仮定し、句の類似度を DP マッチングで計算し、並列構造を解析している。これは実際に構文解析ツール KNP として実装されている。しかし KNP で法令文書を解析すると、特に並列構造解析の部分で解析誤りが頻出する。これは、法令文書は他の文書とは異なる言語的性質を有しているためと考えることができる。本論文では、法令文書を対象とした並列構造の検出及びその範囲の同定処理の精度向上を目的とし、法令文書に特化した並列構造を解析するための新しい手法を提案する。

本研究では並列構造を、1 つ以上の前方並列句、並列キー、後方並列句から構成されているとする。並列キーとは、並列構造で句を接続する働きをする語であり、前方及び後方並列句とは、並列キーの前方及び後方にある句で並列関係にあるものである。処理の流れを以下に述べる。まず並列キーを検出する。本研究では、「又は」「及び」「若しくは」「並びに」「と」「や」「かつ」「その他」の８つを並列キーとする。次に前方並列句の主辞を検出する。これは基本的には並列キーの直前の語である。次に後方並列句の候補を検出する。後方並列句の始点は基本的には並列キーの直後の語とする。後方並列句の終点は並列キーより後方にある語とし、前方並列句の主辞の品詞が助詞の場合は終点も助詞とし、主辞の品詞が動詞の場合は終点も動詞とする。名詞が主辞の場合、終点の候補は文節の最後の自立語で、かつ先に検出された前方並列句の主辞と類似度の高い上位 3 個の語とする。但し、読点、句点、他の並列キーに到達した時点で後方並列句の終点の探索を終了する。次に前方並列句の候補を検出する。前方並列句の終点は基本的には先に検出した主辞とし、始点は並列キーより前方にある文節の最初の単語とする。但し、読点、文頭、他の並列キーに到達した時点で前方並列句の始点の探索を終了する。次に得られた後方及び前方並列句の候補の類似度を計算する。並列関係にある句同士は互いに意味的に類似していると仮定し、全ての句の組み合わせから類似度の最も高い句の組を選択し、それぞれ前方並列句、後方並列句とする。句の類似度は、単語単位でアライメントをとり、対応関係にある単語の意味的類似度に基づいて算出する。また、対応関係のない単語があるときは句の類似度も低くし、また、その単語が句の主辞に近い位置にあるほどそのペナルティを大きくする。更に、「第」「条」「項」「号」は法令文の条件番号を表わすのに使われる特別な語であることから、これらの語は同じ語に対応付けられるときのみ句の類似度を大きくした。次に既に決定された前方並列句の前に、別の前方並列句があるかをチェックし、ある場合はその候補を検出する。得られた前方並列句の候補と、すでに同定された前方及び後方並列句の類似度を計算し、それが最も高い候補を次の前方並列句とする。別の前方並列句が発見できなくなるまでこの処理を繰り返す。

次に、階層的な並列構造を解析する手法について述べる。本研究では、下位の並列構造を上位の並列構造解析よりも先に解析する必要があると考え、並列構造を構成する並列

キーを下位の並列構造を構成するものから検出するようにした。すなわち、下位から上位の順に、ボトムアップ式に並列構造を解析する。解析を行う順序は、「又は」「及び」で結ばれる並列構造、「若しくは」「並びに」で結ばれる並列構造、「と」「や」「かつ」「その他」で結ばれる並列構造の順とした。これは、法令文では「又は」「及び」は内側の、「若しくは」「並びに」は外側の並列関係を記述するというルールを考慮したためである。また、階層的な並列構造解析を行う際に、前方並列句、後方並列句のどちらか一方に下位の並列構造が含まれるとき、両者の長さが大きく異なり、句の類似度を正確に見積もることができないという問題点がある。そこで、上位の並列構造の前方並列句、後方並列句が下位の並列構造を含むときは、それを下位の並列構造の後方並列句のみに置き換えることで、上位の並列句同士の長さのバランスをとるように工夫した。

提案手法に基づき、３つ以上の並列句を持つ並列構造や階層的な並列構造を解析するシステムを作成した。このシステムを用いて３００文（うち２００文を開発データ、１００文を評価データとして使用）からなる法令文の解析を行い、検出された並列構造を評価した。その結果、評価データにおける並列構造検出のＦ値は５０％、並列キー検出のＦ値は９３％、前方並列句の検出のＦ値は６５％、後方並列句の検出のＦ値は６４％となった。また、提案手法をKNPと比較して評価したところ、提案手法はKNPよりも高い評価値を得た。KNPの並列構造のＦ値は２６％であったのに対し、本研究では５０％であり、KNPを２４％上回った。

目 次

図 目 次

表 目 次

第1章　はじめに

1.1　研究の背景

自然言語処理研究における並列構造解析は、解決が困難な問題の一つである。その原因として、文書ドメインごとに言語的性質が異なることが挙げられる。現在では、文書ドメインに特化した並列構造の研究が試みられている。本研究では対象ドメインを法令文書にし、法令文書に特化した並列構造解析を行う。

法令工学は、複数の法令の論理的整合性の検証や、法令文の理解を助けるシステムの構築などを目的とした研究分野である。法令工学においては、法令文を解析し、その意味を理解することが重要な要素技術となる。しかしながら、法令文の構文・意味解析では特に並列構造解析の精度が低い。並列構造解析とは、「A かつ B」「A 又は B」のような並列関係にある句を同定する処理である。

以下に法令文における並列構造の特徴を例とともに挙げる [5][10]。

1. 長い句同士が並列関係にある
   (a) 第七条第一項第二号に規定する被保険者としての被保険者期間 **及び**同項第三号に規定する被保険者としての被保険者期間...
   (b) ...前条第四項の規定により被保険者の資格を取得した旨の報告を受けたとき、**又は** 同条第五項の規定により第三号被保険者の資格の取得に関する届出を受理したとき は、...
2. 短い句と長い句が並列関係にある
   (a) ...他の年金給付 **又は**被用者年金各法による年金たる給付...
   (b) ...地方公務員共済組合連合会**又は**私立学校教職員共済法の規定により私立学校教職員共済制度を管掌することとされた日本私立学校振興・共済事業団...
3. 「A、B かつ C」のように 3 つ以上の句が並列関係にある場合が多い
   (a) ...老齢、障害 **又は**死亡...
   (b) ...保険料全額免除期間、保険料四分の三免除期間、保険料半額免除期間 **及び** 保険料四分の一免除期間...

4. 選択的接続と併合的接続

   法令文書では「又は」「若しくは」「及び」「並びに」の４つが主に並列構造を表現する語として用いられる。

   (a) 「及び」「並びに」は併合的接続を表わす

   (b) 「又は」「若しくは」は選択的接続を表わす

5. 法令文における並列構造の優先度

   (a) 併合的接続では最も内側の並列に「及び」、それより外側には「並びに」を用いる
   （第十二条 第一項 **及び** 第四項） **並びに** 第百五条第一項）...

   (b) 選択的接続では最も外側の並列に「又は」、それより内側には「若しくは」を用いる
   第七条第一項（（第二号 **若しくは** 第三号）に該当するに至ったとき **又は** 第三号から第五号までのいずれかに該当するとき）...

   この区別によって階層的な並列構造を表現する。

法令文にはこのような特徴が顕著に見られ、通常のテキストにおける並列構造とは異なる性質を持っている。法令文の並列構造を正しく認識することが出来れば、法令文の構造・意味解析の精度向上にも繋がり、法令工学における様々な研究に大いに貢献する。しかしながら、構文解析を行う既存ツールは、上記のような法令文の特徴を考慮していないため、解析誤りが多いという問題がある。法令文の構文解析の誤りの多くは並列構造が正しく同定できないことが主な要因の一つとなっている。

1.2 研究の目的

前節で述べた背景を踏まえ、本研究では、法令文書を対象とした並列構造の解析手法を提案し、その精度を向上させることを目的とする。本研究の特色は、法令文の特徴を考慮して並列構造を解析する点にある。法令文書における並列構造の特徴として「又は」「若しくは」は選択的接続、「及び」「並びに」は併合的接続を表わし、それらの間の優先度も規定されていることが挙げられる。また、法令文における並列構造は、長い句が並列の関係にあることや、「A、B 又は C」のように３つ以上の句が並列関係にある場合が多いなど、通常のテキストにおける並列構造とは異なる性質を持つ。

本研究では、２章で述べる並列構造解析の先行研究の成果を踏まえつつ、上記のような法令文の特徴を考慮し、法令ドメインに特化した並列構造解析手法を考案する。

1.3　本論文の構成

本論文の構成は以下のとおりである。２章では並列構造解析の手法や法令文に頻出する並列表現の分析に関する関連研究を示し、本研究との差異について述べる。３章では本研究で提案する並列構造解析手法について述べる。４章では提案手法を用いた並列構造解析の実験について報告し、結果の評価と考察を行う。５章では本研究のまとめ、および今後の課題について述べる。

第2章　関連研究

本章では、本論文の関連研究について述べる。また、本論文との相違について論じる。

2.1　並列構造解析の手法

2.1.1　句の類似性に基づく並列構造解析の手法

まず 1 つ目の例として句の類似性に基づく並列構造解析の手法を示す。従来の並列構造解析の方式が局所的な解析を基本としていたために、とくに長い文の解析が困難であった。黒橋・長尾による研究では、その問題を解決するために、文内のできるだけ広い範囲を同時的に調べる手法を提案している [1]。その手法はダイナミックプログラミングの手法によって実装されている。

この手法では、まず文節間の類似度を品詞の一致、文字列の一致、シソーラスによる意味的な近さ、などによって定義し、すべての文節間についてこの値を計算している。これは、１文中の並列する部分は何らかの意味において類似していると考えることが出来るという考えに基づいている。

図 2.1 の三角行列では、対角要素は文節を、(i,j) 要素は i 番目の文節と j 番目の文節の類似性を示している。ここで、この研究では以下の用語を定義している。

- 並列キー：並列構造の存在を示す文節
- 部分行列：並列キーの右上部分の行列（図１では点線で囲まれた部分）。
- パス：部分行列の中の１番下の行の 0 以外のある要素から１番左の列のある要素までの左上方向への要素の並び。
- パスのスコア：パスに含まれる要素の値の総和。パス内の要素の並びが真左上方向からずれる場合にはペナルティとして値を小さくする。

文節間の類似度をパスのスコアという形で計算を行っている。具体的には、ダイナミックプログラミングの手法によって並列のキーに対する最大スコアのパスを求め、そのパスによって示される最も類似度の高い文節列を並列構造の範囲とする。また、この研究では、並列構造間の範囲に関する情報をまとめることで文を簡単化し、文の大まかな構造を把握している。これによって、長い句の処理の精度を向上させている。実験の結果、１５０文に対して９６％の文節について正しい係り先を求めることができたと報告している。

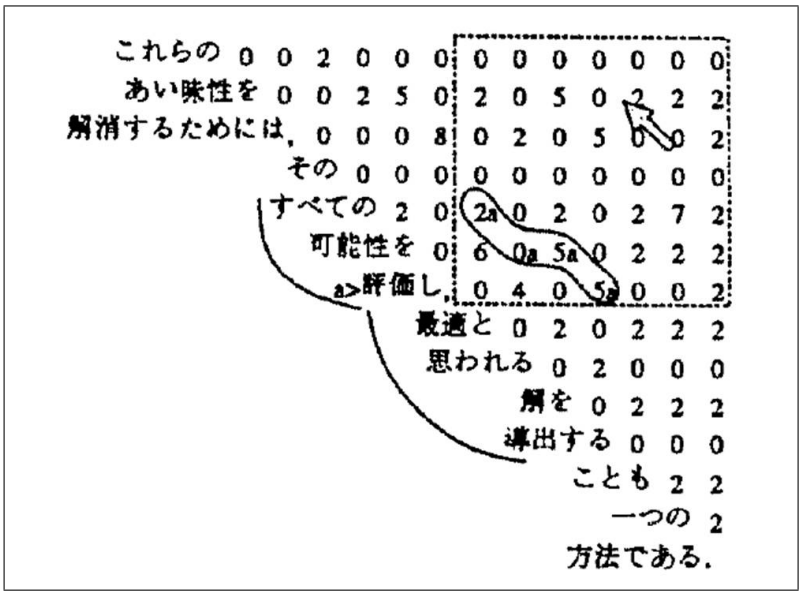

図 2.1: 並列構造の推定の例（論文 [1] より）

2.1.2 確率モデルに基づく並列構造解析の手法

河原・黒橋は黒橋・長尾の手法 [1] のような類似性だけで並列構造の解析をするのでは不十分だと主張し、構文解析と格構造解析とを統合した並列構造の持つ構文的曖昧性を解消するための確率モデルを提案している [2][3]。このモデルは、構文的または意味的類似性及び共起統計を使用することによって、並列構造の候補同士の類似性を確率的に評価している。この確率と、大規模格フレーム辞書に基づく格解析の確率を、語彙化された解析フレームワークとして統合している。

構文・並列・格構造解析の統合的確率モデルとして式 (2.1) のモデルを示している。

$$\begin{aligned}(T_{best}, L_{best}) = \quad & \mathbf{arg\ max}_{(T,L)} P(T, L|S) \\ & \mathbf{arg\ max}_{(T,L)} \frac{P(T, L, S)}{P(S)} \\ & \mathbf{arg\ max}_{(T,L)} P(T, L, S)\end{aligned} \tag{2.1}$$

これは入力文 S が与えられたときの構文構造 T と格構造 L の同時確率 $P(T, L|S)$ を最大にするような構文構造 T_{best} と格構造 L_{best} を選択するモデルである。並列構造のそれぞれの候補に対し、$P(T, L, S)$ を求め、それが最大となる並列構造の解釈を決める。すなわち、構文、格構造、並列構造の曖昧性を同時に解消する。また、このアプローチでは、並列構造の検出と、並列構造の範囲の同定という並列構造の持つ構文的曖昧性の二つのタスクを同時に評価している。実験はウェブテキストを使用し、「構文解析器 KNP」と「確率

的構文・格解析モデル」(並列構造の曖昧性を考慮しないモデル) の 2 つのシステムと比較している。高い計算コストを下げるために、CKY 法に基づいて確率を効率よく求める。提案手法の精度がそれぞれに対して 1.4 %、1.0 %向上したとことを報告しており、このアプローチの有効性を示している。

2.1.3 階層的な並列構造の解析

Hara らは並列構造解析のためのハイブリッドアプローチを提案している [4]。並列構造解析のためのシンプルな文法と、並列句同士の局所的な対称性を得る為の系列アライメントに基づく素性を組み合わせたアプローチである。また品詞以外の補助的な情報を用いていないことも特徴である。

この研究で使用された非終端記号とシンプルな文法としての生成ルールの一部を以下に示す。

非終端記号の例

- COORD : Complete coordination （完全な並列構造）
- COORD' : Partially-built coordination （部分的な並列構造）
- CJT : Conjunct （並列句）
- N : Non-coordination （並列句以外の要素）

生成文法の例

- $\mathrm{COORD}_{i,m} \rightarrow \mathrm{CJT}_{i,j}\ \mathrm{CC}_{j+1,k-1}\ \mathrm{CJT}_{k,m}$
- $\mathrm{COORD'}_{i,m}[j] \rightarrow \mathrm{CJT}_{i,j}\ \mathrm{CC}_{j+1,k-1}\ \mathrm{CJT}_{k,m}$
- $\mathrm{CJT} \rightarrow (COORD|N)_{i,j}$
  ※文法における添字は非終端記号が支配する単語の範囲を表わす。

これらの生成文法と、句と句のアライメントに基づく類似度計算によって階層的な並列構造に対応した解析を行っている。また、考えられ得る複数の候補から正しい候補を選択するために、アライメントに基づく素性の重みを最適化する方法として、パーセプトロン学習アルゴリズムに基づく手法を提案している。これはトレーニング中のグローバル重みベクトル w を最適化する方法である。

この研究は以下の 2 つのタスクで評価を行っている。

1. フレーズタイプ（(Noun Phrase; 名詞句）や（Verb PHrase; 動詞句）など）を問わず、3598 個の並列構造の範囲を決定するタスク

比較対象として Bikel-Colins parser[6] と Charniak-Johnson parser[7] を用いている。再現率のみをそれぞれと比較した結果、全体として再現率は向上した。しかし、並列構造を構成するフレーズのタイプ別に比較した場合、動詞句と文に関しては再現率が低下したことを報告している。その原因としては、動詞句と文に関しては、それらの並列句（節）が必ずしも似ていないことが原因だと考察している。

2. NP の並列構造を見つけて、その範囲を決定するタスク

   比較対象として Bikel-Collins parser[6], Charniak-Johnson parser[7], Shimbo-Hara method[8] を用いている。精度、再現率、F 値でそれぞれ比較した結果、すべてにおいて提案手法が良い結果を示した。また、もし３つ以上の並列句だけを扱った場合は、差がより顕著になったことを報告している。その理由として、３つ以上の並列句に対応するために生成ルールを厳しくしたためと考察している。

2.2 法令文に頻出する並列表現の分析

加藤らは法律文の並列構造をパターン化し、そのうち Cabocha で正しく並列構造解析が出来ないパターンに対する修正方法を提案している [5]。法律文では、並列関係を表わすために用いる接続詞として、主に「又は」「若しくは」「及び」「並びに」の４つが用いられている。この論文では、これらを用いた並列表現を対象としており、法律文においてはこれらの接続詞の使い分けによって階層構造を表現している。使い分け方は、併合的接続として「及び」「並びに」が使用され、最も内側の並列に「及び」を、それより外側には「並びに」を用いる。選択的接続としては、「又は」「若しくは」が使用され、最も外側の並列には「又は」を、それより内側には「若しくは」を用いる。この区別により、法令文における並列構造の優先度が定められている。この論文では、具体的に以下について扱っている。

1. 階層がない名詞節２つの並列
2. 階層がない名詞節３つ以上の並列
3. 階層がある名詞節の並列
4. 人手の修正が必要な並列

これら４つのパターンについて、正しく解析するための指針を示している。

2.3 本研究との関連

黒橋・長尾により提案された句の類似性に基づく並列構造解析の手法は、解析対象が法令文書に特化しているわけではないため、並列構造における法令文特有の表現を解析することが困難であることがわかっている。

実際に黒橋・長尾の手法に基づく解析ツール KNP を法令文に適用すると、特に並列構造解析の部分で解析誤りが頻出する。主な解析誤りの特徴を以下に挙げる。

- ３つ以上の並列句の同定が出来ていないときが頻出する。
- 法令文特有のルールによる階層的並列構造に対応できていない。
- 並列キーの検出ができていないときが頻出する。

このような KNP の抱える問題を解決するために、法令文に特化したシステムを構築し、法令文書における並列構造解析の精度を向上させることを本研究の目的とする。但し、他の多くの先行研究と同様に、黒橋・長尾の手法は並列関係にある句の類似性に基づいており、本研究でもこの考えを参考としている。また、本研究の成果を評価するために、構文解析器（KNP）との比較を行う。一方、河原・黒橋によって提案された確率モデルに基づく並列構造解析の手法は、ウェブテキストを対象としているため、実際に法令文にそのシステムを適用させたときの精度は不明である。Hara らによる階層的な並列構造の解析については、解析対象が英語の文であり、ドメインも医学生物分野であるため、日本語の法令文書とはその言語的性質が異なる。ただし、階層的な並列構造に対応している点を参考にした。本研究でも階層的な並列構造に対応した手法を提案するが、本研究ではボトムアップ式に下位の並列構造からその範囲を決定していく手法を提案する。提案手法では、並列関係にある句と句の類似性を測ることで並列構造の範囲を決定するが、句と句の類似性を計算する際には黒橋・長尾によるダイナミックプログラミングに基づく手法 [1] を参考にした。但し、彼らの手法は文節単位の類似度を基に句の類似度を計算するのに対し、本研究では単語単位の類似度を基にしている。

第3章　提案手法

本節では提案する並列構造解析手法について述べる。3.1 節では説明に用いる用語を定義する。3.2 節では並列構造解析前に行う前処理について述べる。3.3 節では、まず階層的ではない並列構造を検出する手法について述べる。ここでは提案手法の基本的なアルゴリズムを示す。次に 3.4 節で階層的な並列構造を解析する手法について述べる。

3.1　用語の定義

本研究で用いる用語の定義を以下に示す。

- 並列キー：並列構造で句を接続する働きをする語（本研究では（又は、若しくは、及び、並びに、や、と、かつ、その他）を並列キーとして扱う)。以下、key と表わす。
- 前方並列句：並列キーの前方にある句で並列関係にあるもの。以下、pf と表わす。
- 後方並列句：並列キーの後方にある句で並列関係にあるもの。以下、pb と表わす。
- 並列構造：$(pf_n, ..., pf_{n-1}, pf_1\ key\ pb)$

  本研究では、並列構造は１つ以上の前方並列句、１つの並列キー、１つの後方並列句から構成されるとする。法令文では、以下のように、３つ以上の並列句から構成される並列構造は上記のパターンで出現する。２つ以上の前方並列句があるとき、key に近いものから順に $pf_1, pf_2, ...pfn$ 番号をつける。

  例.（保険料全額免除期間(pf_3)、保険料四分の三免除期間(pf_2)、保険料半額免除期間(pf_1) **及び**(key)保険料四分の一免除期間(pb)）

3.2　前処理

ここでは並列構造解析を行う前に事前に行う処理について述べる。

テキストの整形

本研究では、解析対象となる法令文として国民年金法を用いる。公開されている国民年金

法は、条文番号など法令文以外の情報も含まれる。そこで、テキストの整形として、空行の削除や、解析処理を行う際に不要な情報を削除する。具体的に削除した情報を以下に示す。

- 空行
- 「第一条　国民年金制度は、...」における文頭の「第一条」
- 「２　国民年金事業の事務の一部は、...」における文頭の「２」
- 「二　被用者年金各法の被保険者、...」における文頭の「二」

これらの語は簡単なパターンマッチプログラムで自動的に削除をした。人手で削除を行ったものに関しては以下のものがある。

- 「第一章　総則」などの語
- 「第一節　通則」などの語

括弧の処理

法令文書では丸括弧と鉤括弧が頻繁に使用される。括弧は解析を困難にさせる原因になることから、文を括弧の内側と外側に分ける処理を行う。以下に例を示す。

丸括弧の場合、

被保険者（第三号被保険者を除く。次項において同じ。）は、...

という文章を、

被保険者 P1_R は、... P1_R：第三号被保険者を除く。次項において同じ。

のように分割する。P1_R はそこに丸括弧があることを表わす。二行目は丸括弧の中の文を取り出している。鉤括弧の場合、

... 第一項において「財政均衡期間」という。

という文章を、

... 第一項において P1_S という。 P1_S：財政均衡期間

のようにのように分割する。P1_S はそこに鉤括弧があることを表わす。二行目は鉤括弧

の中の文を取り出している。

形態素解析
JUMAN[9] を用いて形態素解析を行う。単語の分割と品詞の決定を行う。

3.3 並列構造解析の手法

本研究で提案する並列構造解析の手法についての概要を以下に述べる。

1. **並列キー Key の検出**
   文に含まれる並列キーを検出する。

2. **前方並列句の主辞 $Head$ を検出**
   並列キーの直前の単語を前方並列句の主辞とし、検出する。

3. **後方並列句の候補 $PB = \{...pb_k[x, y]...\}$ を検出**
   並列キーの後方には一つの並列句しか存在しない。しかし、並列句の範囲がどこまでかを決めるのは難しい。ここではまず、後方並列句の候補を検出する。

4. **前方並列句の候補 $PF_1 = \{...pf_{1j}[x, y]...\}$ を検出**
   並列キーの前方には、読点を伴って複数の並列句が存在する可能性がある。まずは並列キーの直前の並列句 pf_1 の候補 PF_1 を検出する。

5. **前方並列句 pf_1、後方並列句 pb の決定**
   PB と PF_1 の全ての組み合わせについて、句の類似度を計算し、類似度が最大の組み合わせを決定する。

6. **pf_1 の前に別の並列句があるかの判定**
   pf_1 の前に別の並列句が存在するかを判定する。

7. **pf_1 の前に別の並列句がある場合は、前方並列句の候補を検出**
   すなわち、４の処理に戻る。

8. **pf_1 の前に別の並列句がない場合は、並列句の範囲を出力**
   検出された前方並列句及び後方並列句の範囲を出力する。

以下、それぞれの処理の詳細について述べる。

3.3.1　並列キーの検出

本研究では、法令文で頻繁に使用される「又は」「及び」「若しくは」「並びに」の４つの単語に加え、「と」「や」「かつ」「その他」も並列キーとする。並列キーの一覧を以下に示す。

「又は」「及び」「若しくは」「並びに」「と」「や」「かつ」「その他」

これらの語を考慮すれば、法令文書におけるほとんどの並列構造に対応できると考えられる。処理としては、まず JUMAN で形態素解析をした文書から並列キーを検出する。JUMAN では「並びに」という語に対して「並びに」と出力される場合と、「並び」と「に」のように区別されて出力される場合があることが確認された。以下に具体例を示す。

JUMAN の解析結果における「並びに」の出力結果の例

1. 「並びに」と出力される例

   ...
   四 よん 四 名詞 6 数詞 7 * 0 * 0 ...
   項 こう 項 接尾辞 14 名詞性名詞助数辞 3 * 0 * 0 ...
   P1_R 含む ふくむ 含む 動詞 2 * 0 子音動詞マ行 9 基本形 2 ...
   並びに ならびに 並びに 接続詞 10 * 0 * 0 * 0 ...
   第 だい 第 接頭辞 13 名詞接頭辞 1 * 0 * 0 ...
   百五 ひゃくご 百五 名詞 6 数詞 7 * 0 * 0 ...
   条 じょう 条 接尾辞 14 名詞性名詞助数辞 3 * 0 * 0 ...
   ...

2. 「並び」と「に」に区別されて出力される例

   ...
   負担 ふたん 負担 名詞 6 サ変名詞 2 * 0 * 0 ...
   の の の 助詞 9 接続助詞 3 * 0 * 0 ...
   額 がく 額 名詞 6 普通名詞 1 * 0 * 0 ...
   並び ならび 並ぶ 動詞 2 * 0 子音動詞バ行 8 基本連用形 8 ...
   に に に 助詞 9 格助詞 1 * 0 * 0 ...
   この この この 指示詞 7 連体詞形態指示詞 2 * 0 * 0 ...
   法律 ほうりつ 法律 名詞 6 普通名詞 1 * 0 * 0 ...
   に に に 助詞 9 格助詞 1 * 0 * 0 ...
   ...

   この場合には「並び」と「に」の２単語を並列キーとする。

更に、「と」に関してはJUMANが出力する詳細品詞の候補が格助詞のみの場合と、格助詞と接続助詞の２つとなる場合がある。「と」は、接続助詞が詳細品詞の候補に含まれ、かつその「と」の直後の語の品詞が名詞か連体詞の時にのみ、並列キーとして扱う。以下に具体例を示す。

JUMANの解析結果における「と」の出力結果

1. 「と」の詳細品詞の候補が格助詞のみの例

   ...
   要し ようし 要する 動詞 2 * 0 サ変動詞 16 基本連用形 8 ...
   ない ない ない 接尾辞 14 形容詞性述語接尾辞 5 イ形容詞アウオ段 18 基本形 2 ...
   もの もの もの 名詞 6 形式名詞 8 * 0 * 0 ...
   と と と 助詞 9 格助詞 1 * 0 * 0 ...
   さ さ する 動詞 2 * 0 サ変動詞 16 未然形 3 ...
   れた れた れる 接尾辞 14 動詞性接尾辞 7 母音動詞 1 タ形 10 ...
   保険 ほけん 保険 名詞 6 普通名詞 1 * 0 * 0 ...
   ...

   「と」の詳細品詞として接続助詞を候補に含んでいないので、並列キーとみなさない。

2. 「と」の詳細品詞の候補が格助詞と接続助詞の２つであり、直後の語の品詞が名詞である例

   ...
   納付 のうふ 納付 名詞 6 サ変名詞 2 * 0 * 0 ...
   済 済 済 未定義語 15 その他 1 * 0 * 0 ...
   期間 きかん 期間 名詞 6 時相名詞 10 * 0 * 0 ...
   と と と 助詞 9 格助詞 1 * 0 * 0 ...
   @ と と と 助詞 9 接続助詞 3 * 0 * 0 ...
   保険 ほけん 保険 名詞 6 普通名詞 1 * 0 * 0 ...
   料 りょう 料 名詞 6 普通名詞 1 * 0 * 0 ...
   免除 めんじょ 免除 名詞 6 サ変名詞 2 * 0 * 0 ...
   ...

   「と」の詳細品詞として接続助詞を候補に含み、かつ直後の語の「保険」の品詞が名詞なので、並列キーとする。

3. 「と」の詳細品詞の候補が格助詞と接続助詞の２つであり、直後の語の品詞が連体

詞である例

...
に に に 助詞 9 格助詞 1 * 0 * 0 ...
@ に に に 助詞 9 接続助詞 3 * 0 * 0 ...
基準 きじゅん 基準 名詞 6 普通名詞 1 * 0 * 0 ...
障害 しょうがい 障害 名詞 6 普通名詞 1 * 0 * 0 ...
と と と 助詞 9 格助詞 1 * 0 * 0 ...
@ と と と 助詞 9 接続助詞 3 * 0 * 0 ...
他の たの 他の 連体詞 11 * 0 * 0 * 0 ...
障害 しょうがい 障害 名詞 6 普通名詞 1 * 0 * 0 ...
と と と 助詞 9 格助詞 1 * 0 * 0 ...
@ と と と 助詞 9 接続助詞 3 * 0 * 0 ...
...

「と」の詳細品詞として接続助詞を候補に含み、かつ直後の語の「他の」の品詞が連体詞なので、並列キーとする。

4. 「と」の詳細品詞の候補が格助詞と接続助詞の 2 つであり、直後の語の品詞が助詞である例

...
料 りょう 料 名詞 6 普通名詞 1 * 0 * 0 ...
免除 めんじょ 免除 名詞 6 サ変名詞 2 * 0 * 0 ...
期間 きかん 期間 名詞 6 時相名詞 10 * 0 * 0 ...
と と と 助詞 9 格助詞 1 * 0 * 0 ...
@ と と と 助詞 9 接続助詞 3 * 0 * 0 ...
を を を 助詞 9 格助詞 1 * 0 * 0 ...
合算 がっさん 合算 名詞 6 サ変名詞 2 * 0 * 0 ...
した した する 動詞 2 * 0 サ変動詞 16 タ形 10 ...
...

「と」の詳細品詞として接続助詞を候補に含むが、直後の語の「を」の品詞が助詞なので、並列キーとみなさない。

5. 「と」の詳細品詞の候補が格助詞と接続助詞の 2 つであり、直後の語が「その他」である例

...
と と と 助詞 9 格助詞 1 * 0 * 0 ...

なった なった なる 動詞 2 * 0 子音動詞ラ行 10 タ形 10 ...
@ なった なった なる 動詞 2 * 0 子音動詞ラ行 10 タ形 10 ...
障害 しょうがい 障害 名詞 6 普通名詞 1 * 0 * 0 ...
と と と 助詞 9 格助詞 1 * 0 * 0 ...
@ と と と 助詞 9 接続助詞 3 * 0 * 0 ...
その他 そのた その他 名詞 6 普通名詞 1 * 0 * 0 ...
障害 しょうがい 障害 名詞 6 普通名詞 1 * 0 * 0 ...
P3_R 障害 しょうがい 障害 名詞 6 普通名詞 1 * 0 * 0 ...
と と と 助詞 9 格助詞 1 * 0 * 0 ...
@ と と と 助詞 9 接続助詞 3 * 0 * 0 ...
...

「と」の直後の語が「その他」であり、２つの並列キーが連続して出現しているため、例外処理を行う。このような場合には、「と」だけを並列キーと見なし、「その他」は並列キーと見なさない。但し、「と」の詳細品詞の候補として接続助詞が含まれていなければならないとする。

また、文頭に並列キーが出現する場合は、並列キーとしては扱わないこととする。以下に例を示す。

1. 文頭に並列キーが出現する例

   その他 そのた その他 名詞 6 普通名詞 1 * 0 * 0 ...
   @ その他 そのほか その他 接続詞 10 * 0 * 0 * 0 ...
   障害 しょうがい 障害 名詞 6 普通名詞 1 * 0 * 0 ...
   が が が 助詞 9 格助詞 1 * 0 * 0 ...
   二 に 二 名詞 6 数詞 7 * 0 * 0 ...
   ...

   このような場合には、明らかに並列構造の前方並列句が存在しないので、並列キーとして扱わない。尚、このような例は丸括弧内で頻出することを付け加えておく。

3.3.2 前方並列句の主辞の検出

日本語の特徴として、句の主辞はその句の最後の単語である。よって、前方並列句の主辞は、並列キーの直前の単語とする。本論文では、前方並列句の主辞を $Head$ と表記する。しかし、例外処理を行わなければならない場合も存在する。以下、検出される主辞の例を、並列キーのその前後に出現する語に対する JUMAN の解析結果とともに示す。太

字は並列キーを、下線の引いてある語は $Head$ を表わす。

「並列キー」とその前後の語に対する JUMAN の解析結果

1. 「並列キー」の直前の語が主辞となる例

   ...
   遺族 いぞく 遺族 名詞 6 普通名詞 1 * 0 * 0 ...
   基礎 きそ 基礎 名詞 6 普通名詞 1 * 0 * 0 ...
   年金 ねんきん 年金 名詞 6 普通名詞 1 * 0 * 0 ...
   又は または 又は 助詞 9 接続助詞 3 * 0 * 0 ...
   寡婦 やもめ 寡婦 名詞 6 普通名詞 1 * 0 * 0 ...
   年金 ねんきん 年金 名詞 6 普通名詞 1 * 0 * 0 ...
   は は は 助詞 9 副助詞 2 * 0 * 0 ...
   ...

   並列キー「又は」の直前の語が「年金」であるり、この語を $Head$ とする。

2. 「並列キー」の直前の語が鉤括弧となる例

   ...
   P1_S 者 しゃ 者 接尾辞 14 名詞性名詞接尾辞 2 * 0 * 0 ...
   、 、 、 特殊 1 読点 2 * 0 * 0 ...
   P2_S 夫 おっと 夫 名詞 6 普通名詞 1 * 0 * 0 ...
   及び および 及び 助詞 9 接続助詞 3 * 0 * 0 ...
   P3_S 妻 つま 妻 名詞 6 普通名詞 1 * 0 * 0 ...
   に に に 助詞 9 格助詞 1 * 0 * 0 ...
   @ に に に 助詞 9 接続助詞 3 * 0 * 0 ...
   は は は 助詞 9 副助詞 2 * 0 * 0 ...
   ...

   P2_S は、括弧の前処理によって挿入された行であり、元はこの位置に鉤括弧があったことを表わす。また、「夫」以降は鉤括弧内で最後に出現する単語とその品詞などの形態素情報である。並列キー「及び」の直前の語が鉤括弧のときは、括弧内の最後の語（この場合は「夫」）を $Head$ とする。

3. 「並列キー」の直前の語が丸括弧となる例

   ...
   第 だい 第 接頭辞 13 名詞接頭辞 1 * 0 * 0 ...
   四 よん 四 名詞 6 数詞 7 * 0 * 0 ...
   項 こう 項 接尾辞 14 名詞性名詞助数辞 3 * 0 * 0 ...
   P1_R 含む ふくむ 含む 動詞 2 * 0 子音動詞マ行 9 基本形 2 ...

並びに ならびに 並びに 接続詞 10 * 0 * 0 * 0 ...
第 だい 第 接頭辞 13 名詞接頭辞 1 * 0 * 0 ...
百五 ひゃくご 百五 名詞 6 数詞 7 * 0 * 0 ...
条 じょう 条 接尾辞 14 名詞性名詞助数辞 3 * 0 * 0 ...
...

この例以降が例外処理である。P1_R は、括弧の前処理によって挿入された行で、元はこの位置に丸括弧があったことを表わす。並列キー「並びに」の直前の語が丸括弧のときは、この語より更に前の語である「項」を $Head$ とする。

4. 「並列キー」の直前の語が読点となる例

...
受けた うけた 受ける 動詞 2 * 0 母音動詞 1 タ形 10 ...
とき とき とき 名詞 6 副詞的名詞 9 * 0 * 0 ...
、 、 、 特殊 1 読点 2 * 0 * 0 ...
又は または 又は 接続詞 10 * 0 * 0 * 0 ...
同 どう 同 接頭辞 13 名詞接頭辞 1 * 0 * 0 ...
条 じょう 条 名詞 6 普通名詞 1 * 0 * 0 ...
第 だい 第 接頭辞 13 名詞接頭辞 1 * 0 * 0 ...
...

並列キー「又は」の直前の語が読点なので、この語より更に前の語である「とき」を $Head$ とする。

以上の例で挙げたものは全て $Head$ が名詞の場合であるが、並列キーの直前の語がどのような品詞であっても同様に $Head$ とみなす。以下に $Head$ が助詞及び動詞の例を示す。

1. $Head$ が助詞の例

...
至った いたった 至る 動詞 2 * 0 子音動詞ラ行 10 タ形 10 ...
日 にち 日 名詞 6 時相名詞 10 * 0 * 0 ...
@ 日 ひ 日 名詞 6 時相名詞 10 * 0 * 0 ...
@ 日 にち 日 名詞 6 地名 4 * 0 * 0 ...
に に に 助詞 9 格助詞 1 * 0 * 0 ...
@ に に に 助詞 9 接続助詞 3 * 0 * 0 ...
、 、 、 特殊 1 読点 2 * 0 * 0 ...
その他 そのた その他 名詞 6 普通名詞 1 * 0 * 0 ...
の の の 助詞 9 接続助詞 3 * 0 * 0 ...
者 もの 者 名詞 6 普通名詞 1 * 0 * 0 ...
に に に 助詞 9 格助詞 1 * 0 * 0 ...

...

2. *Head* が動詞の例

...
が が が 助詞 9 格助詞 1 * 0 * 0 ...
消滅 しょうめつ 消滅 名詞 6 サ変名詞 2 * 0 * 0 ...
し し する 動詞 2 * 0 サ変動詞 16 基本連用形 8 ...
、 、 、 特殊 1 読点 2 * 0 * 0 ...
又は または 又は 接続詞 10 * 0 * 0 * 0 ...
同 どう 同 接頭辞 13 名詞接頭辞 1 * 0 * 0 ...
一 ひと 一 名詞 6 数詞 7 * 0 * 0 ...
@ 一 いち 一 名詞 6 数詞 7 * 0 * 0 ...
人 り 人 接尾辞 14 名詞性名詞助数辞 3 * 0 * 0 ...
...

また、JUMAN では名詞を動詞の連用形として誤って解析する場合が多い。以下に例を示す。

1. JUMAN の解析誤りの例

偽り いつわり 偽る 動詞 2 * 0 子音動詞ラ行 10 基本連用形 8 ...
その他 そのた その他 名詞 6 普通名詞 1 * 0 * 0 ...
不正 ふせい 不正だ 形容詞 3 * 0 ナ形容詞 21 語幹 1 ...
の の の 助詞 9 接続助詞 3 * 0 * 0 ...
手段 しゅだん 手段 名詞 6 普通名詞 1 * 0 * 0 ...
...

この例では、並列キー「その他」の直前の語である *Head* が「偽り」であり、JUMAN の解析結果では品詞は動詞になっている。これは明らかに JUMAN の解析誤りであり、この品詞は正しくは「名詞」である。よって、品詞が動詞であり、活用形の中に「連用」という言葉が含まれる場合には、シソーラス (日本語語彙大系 [11]) を用いて *Head* の語を検索し、ヒットすれば *Head* の品詞を名詞として扱い、ヒットしなければそのまま動詞として扱う。但し、上述の「*Head* が動詞の例」のように *Head* と並列キーの間に読点がある場合は、このような例外処理はせず、JUMAN の解析結果の通りに *Head* の品詞を動詞とする。

3.3.3 後方並列句の候補の検出

ここでは後方並列句の候補 $pb_k[x, y]$ を検出する。x は後方並列句の始点となる単語の位置を、y は終点を表わす。

後方並列句の始点の検出

後方並列句の始点 x は並列キーの直後の語とする。但し、以下の場合は例外処理として更にその次の語を x とする。太字は並列キーを、下線の引いてある語は x を表わす。

- 「その他」が並列キーで、直後の語が「の」である場合

  ...
  の の の 助詞 9 接続助詞 3 * 0 * 0 ...
  生活 せいかつ 生活 名詞 6 サ変名詞 2 * 0 * 0 ...
  水準 すいじゅん 水準 名詞 6 普通名詞 1 * 0 * 0 ...
  その他 そのた その他 接尾辞 14 名詞性名詞接尾辞 2 * 0 * 0 ...
  の の の 助詞 9 接続助詞 3 * 0 * 0 ...
  諸 しょ 諸 接頭辞 13 名詞接頭辞 1 * 0 * 0 ...
  事情 じじょう 事情 名詞 6 普通名詞 1 * 0 * 0 ...
  に に に 助詞 9 格助詞 1 * 0 * 0 ...
  ...

  「の」の次の語を x とする。この場合、「の」の次の語である「諸」を x とする。

- 並列キーの直後が読点である場合

  ...
  を を を 助詞 9 格助詞 1 * 0 * 0 ...
  下回り したまわり 下回る 動詞 2 * 0 子音動詞ラ行 10 基本連用形 8 ...
  、 、 、 特殊 1 読点 2 * 0 * 0 ...
  かつ かつ かつ 動詞 2 * 0 子音動詞タ行 6 基本形 2 ...
  @ かつ かつ かつ 接続詞 10 * 0 * 0 * 0 "代表表記:且つ/かつ ...
  、 、 、 特殊 1 読点 2 * 0 * 0 ...
  物価 ぶっか 物価 名詞 6 普通名詞 1 * 0 * 0 ...
  変動 へんどう 変動 名詞 6 サ変名詞 2 * 0 * 0 ...
  率 りつ 率 接尾辞 14 名詞性名詞接尾辞 2 * 0 * 0...
  ...
  「、」の次の語を x とする。この場合、「、」の次の語である「物価」を x とする。

後方並列句の終点の検出

後方並列句の終点 y は、並列キーの直後の文節の最後の自立語、及び $Head$ との類似度が高い上位３個の文節の最後の自立語とする。ここで、後方並列句の終点は基本的に文節の最後に出現する自立語であるとする。以下、文中の全ての単語の中から文節の最後の自立語を検出する手法について述べる。

文節の最後の自立語の定義

まず、文節の最後に出現する自立語の条件として、品詞が接尾辞、名詞、動詞、形容詞、副詞のいずれかでなければならないとする。そして、以下の判定ルールによって文節の最後の自立語であるかを判定する。

1. 判定対象とする語が丸括弧内の最後の語である場合は、文節の最後の自立語と判定しない。

   ...
   P1_R いう いう いう 動詞 2 * 0 子音動詞ワ行 12 基本形 2 ...
   に に にる 動詞 2 * 0 母音動詞 1 基本連用形 8 ...
   @ に に にる 動詞 2 * 0 母音動詞 1 基本連用形 8 ...
   ...

   判定対象とする語「いう」が丸括弧 (P1_R) 内の最後の語であるため、文節の最後の自立語と判定しない。

2. 判定対象とする語の品詞が「接尾辞」であり、かつその詳細品詞が「名詞性名詞助数辞」であり、次の語の品詞も「接尾辞」でかつその詳細品詞が「名詞性名詞接尾辞」であり、更に次の語の品詞が「名詞」でない場合は、対象とする語を文節の最後の自立語と判定しない。

   ...
   歳 さい 歳 接尾辞 14 名詞性名詞助数辞 3 * 0 * 0 ...
   未満 みまん 未満 接尾辞 14 名詞性名詞接尾辞 2 * 0 * 0 ...
   である である だ 判定詞 4 * 0 判定詞 25 デアル列基本形 16 ...
   ...

   判定対象とする語「歳」の詳細品詞が「名詞性名詞助数辞」であり、次の語「未満」の詳細品詞が「名詞性名詞接尾辞」であり、更に次の語「である」の品詞は「名詞」ではない（この場合「判定詞」）」。よって「歳」は文節の最後の自立語と判定しない。

3. 判定対象とする語の品詞が「名詞」であり、かつ次の語の品詞も「名詞」である場合は、対象とする語を文節の最後の自立語と判定しない。

   ...
   年金 ねんきん 年金 名詞 6 普通名詞 1 * 0 * 0 ...
   給付 きゅうふ 給付 名詞 6 サ変名詞 2 * 0 * 0 ...
   が が が 助詞 9 格助詞 1 * 0 * 0 ...
   ...

   判定対象とする語「年金」の品詞が「名詞」であり、次の語「給付」の品詞が「名詞」である。よって「年金」は文節の最後の自立語と判定しない。

4. 判定対象とする語の品詞が「名詞」であり、かつ次の語の品詞が「接尾辞」である場合は、対象とする語を文節の最後の自立語と判定しない。

   権 けん 権 名詞 6 普通名詞 1 * 0 * 0 ...
   者 しゃ 者 接尾辞 14 名詞性名詞接尾辞 2 * 0 * 0 ...
   が が が 助詞 9 格助詞 1 * 0 * 0 ...
   ...

   判定対象とする語「権」の品詞が「名詞」であり、次の語「者」の品詞が「接尾辞」である。よって「権」は文節の最後の自立語と判定しない。

5. 判定対象とする語の品詞が「動詞」であり、かつ次の語の品詞も「動詞」である場合は、対象とする語を文節の最後の自立語と判定しない。

   ...
   に に にる 動詞 2 * 0 母音動詞 1 基本連用形 8 ...
   @ に に にる 動詞 2 * 0 母音動詞 1 基本連用形 8 ...
   係る かかる 係る 動詞 2 * 0 子音動詞ラ行 10 基本形 2 ...
   債務 さいむ 債務 名詞 6 普通名詞 1 * 0 * 0 ...
   ...

   判定対象とする語「に」の品詞が「動詞」であり、次の語「係る」の品詞も「動詞」である。よって「に」は文節の最後の自立語と判定しない。

6. 判定対象とする語の品詞が「接尾辞」「名詞」「動詞」「形容詞」「副詞」でない場合は、文節の最後の自立語と判定しない。

7. １から６のいずれにも該当しないとき、文節の最後の自立語と判定する。

次に、後方並列句の候補を得るために、後方並列句の終点 (y) になり得る単語を並列キーより後方に探索する。これは、前方並列句の主辞 $Head$ の品詞に応じて以下のように処理が分かれる。

- $Head$ の品詞が名詞のとき

  文節の最後の自立語であり、品詞が名詞である語を全て後方並列句の終点の候補とする。但し、読点、句点、他の並列キーを検出したときには探索を終了する。次に、終点と $Head$ との類似度を計算し、類似度の高い上位３語のみを最終的な候補とする。単語間の類似度は日本語語彙大系で計算する。類似度の計算方法を式 (3.1) に示す。

  $$sim_w(w_i, w_j) = \frac{2 \times d_c}{d_i + d_j} \tag{3.1}$$

  d_i は日本語語彙大系の木構造における w_i のルートからの深さ、d_j は w_j のルートからの深さ、d_c は w_i と w_j の共通上位ノードの深さを表わす。

  終点の候補は名詞でなければならないが、JUMAN の解析誤りによって名詞の品詞がそれ以外の品詞になっているときがある。以下に例を示す。

  - **JUMAN の解析誤りの例**

    ...
    財政 ざいせい 財政 名詞 6 普通名詞 1 * 0 * 0 ...
    の の の 助詞 9 接続助詞 3 * 0 * 0 ...
    現況 げんきょう 現況 名詞 6 普通名詞 1 * 0 * 0 ...
    及び および 及び 助詞 9 接続助詞 3 * 0 * 0 ...
    見通し みとおし 見通す 動詞 2 * 0 子音動詞サ行 5 基本連用形 8 ...
    を を を 助詞 9 格助詞 1 * 0 * 0 ...
    作成 さくせい 作成 名詞 6 サ変名詞 2 * 0 * 0 ...
    ...

    この例では並列キーの直後の「見通し」の品詞が動詞となっている。しかしこの語は本来「名詞」であり、明らかな JUMAN の解析誤りである。よってこの語を日本語語彙大系を使用して $Head$ との類似度を計算することで、後方並列句の候補とする。すなわち、JUMAN の品詞に関わらず、全ての文節の最後の自立語に対して $Head$ との類似度を計算し、類似度計算が可能なら（その語が日本語語彙大系に登録されていれば)、その語は名詞であるとする。

更に以下の例外処理を行う。

– 並列キーの後方で最初に出現する文節の最後の自立語は常に候補に加える。

  並列キーの直後の文節の最後の自立語に関しては、後方並列句の終点となる場合が多いが、$Head$ との類似度が低く算出される時がある。以下に例を示す。

... 健全な国民生活の 維持 **及び**向上 に寄与すること...

  この場合、$Head$ である「維持」と、並列キーの直後の文節の最後の自立語である「向上」は並列の関係にあるにも関わらず、日本語語彙大系を使用して類似度計算 (式 (3.1)) を行うと、0.18 という低い値となる。よって、並列キーの直後の文節の最後の自立語に関しては、$Head$ との類似度に関わらず後方並列句の候補の一つとする。

– $Head$ と同じ単語が存在するとき、その同じ単語のみを後方並列句の終点とする。

  これは $Head$ と同じ単語が後方並列句の終点の候補にある場合、その単語が正しい並列句の候補になる確率が高いためである。以下に具体例を示す。太字は並列キーを、下線は $Head$ を、二重下線は並列キーの後方にある $Head$ と同じ単語 (但し、読点、句点及び他の並列キーを越えない範囲) を表わす。

... 事務 **並びに**附則第九条の三の四の規定により市町村が処理することとされる 事務 は、...

  $Head$ と同じ単語が後方並列句の候補にあるので、その語だけに候補を限定する。この場合は「事務」である。

- $Head$ の品詞が動詞のとき

文節の最後の自立語であり、かつ品詞が動詞である語を後方並列句の終点の候補とする。尚、類似度計算は行わない。読点、句点、他の並列キーを検出したときに探索を終了する。また、以下の条件を導入する。

– 文節の最後の自立語であり、かつ品詞が動詞である語の直後の語の詳細品詞が「動詞性接尾辞」ならば、その語を代わりに終点の候補とする。

以下にこの例が適用される例を示す。

– **語順が「文節の最後の自立語の動詞」「動詞性接尾辞」の例**

...
なり なり なる 動詞 2 * 0 子音動詞ラ行 10 基本連用形 8 ...
@ なり なり なる 動詞 2 * 0 子音動詞ラ行 10 基本連用形 8 ...
、 、 、 特殊 1 読点 2 * 0 * 0 ...
又は または 又は 接続詞 10 * 0 * 0 * 0 ...
その その その 指示詞 7 連体詞形態指示詞 2 * 0 * 0 ...
額 がく 額 名詞 6 普通名詞 1 * 0 * 0 ...
の の の 助詞 9 接続助詞 3 * 0 * 0 ...
加算 かさん 加算 名詞 6 サ変名詞 2 * 0 * 0 ...
の の の 助詞 9 接続助詞 3 * 0 * 0 ...
対象 たいしょう 対象 名詞 6 普通名詞 1 * 0 * 0 ...
と と と 助詞 9 格助詞 1 * 0 * 0 ...
なって なって なる 動詞 2 * 0 子音動詞ラ行 10 タ系連用テ形 14 ...
@ なって なって なる 動詞 2 * 0 子音動詞ラ行 10 タ系連用テ形 14 ...
いた いた いる 接尾辞 14 動詞性接尾辞 7 母音動詞 1 タ形 10 ...
...

この例は並列キー「又は」の $Head$「なり」（動詞）と、その後方に出現する語を示しているが、正しい後方並列句の終点は「いた」である。しかし、この語は「接尾辞」であるため、$head$ とは品詞が異なる。そのため、$Head$ の品詞が動詞のときの後方探索において、「動詞」の次の語の詳細品詞が「動詞性接尾辞」であれば、その語を後方並列句の候補とする。

- $Head$ の品詞が助詞のとき

  品詞が助詞である語を終点の候補とする。尚、類似度計算は行わない。

- $Head$ の品詞が上記以外のとき

  $Head$ が名詞の場合のときと同じ処理を行う。

3.3.4 前方並列句の候補の検出

ここでは最初の前方並列句 $PF_1 = \{...pf_{1j}[x, y]...\}$ を検出する。前方並列句の終点は常に $Head$ とする。一方、始点 x は、$Head$ より前方方向へ、文節の先頭に出現する単語を探索し、その全てを候補とする。但し、読点、他の並列キー、文頭を検出したときに探索を終了する。

文節の先頭の単語の定義

以下に文節の先頭の単語の判定方法について述べる。原則として、文節の先頭は、品詞が名詞、接頭辞、動詞、形容詞、連体詞、副詞のいずれかでなければならないとする。

1. 判定対象とする語が丸括弧である場合は、文節の先頭の語と判定しない。

   ...
   P1_R いう いう いう 動詞 2 * 0 子音動詞ワ行 12 基本形 2 ...
   に に にる 動詞 2 * 0 母音動詞 1 基本連用形 8 ...
   @ に に にる 動詞 2 * 0 母音動詞 1 基本連用形 8 ...
   ...

   判定対象とする語「いう」が丸括弧内の最後の語であるため、文節の先頭の語と判定しない。

2. 文頭の語が「名詞」「動詞」「接頭辞」「副詞」「形容詞」の場合は文節の先頭の語と判定する。

   年金 ねんきん 年金 名詞 6 普通名詞 1 * 0 * 0 ...
   給付 きゅうふ 給付 名詞 6 サ変名詞 2 * 0 * 0 ...
   の の の 助詞 9 接続助詞 3 * 0 * 0 ...
   ...

   判定対象とする語「年金」が文頭にあるので、文節の先頭の語と判定する。

   偽り いつわり 偽る 動詞 2 * 0 子音動詞ラ行 10 基本連用形 8 ...
   その他 そのた その他 名詞 6 普通名詞 1 * 0 * 0 ...
   不正 ふせい 不正だ 形容詞 3 * 0 ナ形容詞 21 語幹 1 ...
   ...

   判定対象とする語「偽り」が文頭にあるので、文節の先頭の語と判定する。

3. 判定対象とする語の品詞が「名詞」であり、かつその直前の語の詳細品詞が「時相名詞」である場合は、対象とする語を文節の先頭の語と判定する。

   ...
   当時 とうじ 当時 名詞 6 時相名詞 10 * 0 * 0 ...

その者 そのもの その者 名詞 6 普通名詞 1 * 0 * 0 ...
に に に 助詞 9 格助詞 1 * 0 * 0 ...
...

判定対象とする語「その者」（名詞）の直前の詳細品詞が「時相名詞」なので、文節の先頭の語と判定する。

4. 判定対象とする語の品詞が「連体詞」であり、かつその直前の語の詳細品詞が「時相名詞」である場合は、対象とする語を文節の先頭の語と判定する。

...
当時 とうじ 当時 名詞 6 時相名詞 10 * 0 * 0 ...
当該 とうがい 当該 連体詞 11 * 0 * 0 * 0 ...
遺族 いぞく 遺族 名詞 6 普通名詞 1 * 0 * 0 ...
...

判定対象とする語「当該」（連体詞）の直前の詳細品詞が「時相名詞」なので、文節の先頭の語と判定する。

5. 判定対象とする語の品詞が「動詞」であり、かつその語の活用型が「サ変動詞」以外であり、その直前の語の品詞が「名詞」である場合は、対象とする語を文節の先頭の語と判定しない。尚、これは JUMAN の解析誤りに対応した処理である。一般に、サ変動詞以外の動詞は直前に助詞を伴うときが多い。

...
手 て 手 名詞 6 普通名詞 1 * 0 * 0 ...
取り とり 取る 動詞 2 * 0 子音動詞ラ行 10 基本連用形 8 ...
賃金 ちんぎん 賃金 名詞 6 普通名詞 1 * 0 * 0 ...
...

判定対象とする語「取り」の品詞が「動詞」であり、かつ「サ変動詞」以外、更にその直前の語「手」の品詞が「名詞」であるため、文節の先頭の語と判定しない。

6. 判定対象とする語の品詞が「名詞」であり、かつその直前の語の活用型が「サ変動詞」以外であり、更にその前の語の品詞が「名詞」である場合は、対象とする語を文節の先頭の語と判定しない。これも JUMAN の解析誤りに対応した処理である。

...
手 て 手 名詞 6 普通名詞 1 * 0 * 0 ...
取り とり 取る 動詞 2 * 0 子音動詞ラ行 10 基本連用形 8 ...
賃金 ちんぎん 賃金 名詞 6 普通名詞 1 * 0 * 0 ...
変動 へんどう 変動 名詞 6 サ変名詞 2 * 0 * 0 ...
...

判定対象とする語「賃金」の品詞が「名詞」であり、直前の語「取り」の活用型が「サ変動詞」以外であり、更にその直前の語「手」の品詞が「名詞」であるため、文節の先頭の語と判定しない。

7. 判定対象とする語の品詞が「名詞」であり、かつその直前の語の品詞が「名詞」である場合は、対象とする語を文節の先頭の語と判定しない。

...
物価 ぶっか 物価 名詞 6 普通名詞 1 * 0 * 0 ...
指数 しすう 指数 名詞 6 普通名詞 1 * 0 * 0 ...
の の の 助詞 9 接続助詞 3 * 0 * 0 ...
...

判定対象とする語「指数」の品詞が「名詞」であり、直前の語「物価」の品詞も「名詞」であるため、文節の先頭の語と判定しない。

8. 判定対象とする語の品詞が「名詞」であり、かつその直前の語の品詞が「接頭辞」である場合は、対象とする語を文節の先頭の語と判定しない。

...
被 ひ 被 接頭辞 13 名詞接頭辞 1 * 0 * 0 ...
保険 ほけん 保険 名詞 6 普通名詞 1 * 0 * 0 ...
者 しゃ 者 接尾辞 14 名詞性名詞接尾辞 2 * 0 * 0 ...
...

判定対象とする語「保険」の品詞が「名詞」であり、直前の語「被」の品詞が「接頭辞」であるため、文節の先頭の語と判定しない。

9. 判定対象とする語の品詞が「動詞」であり、かつその直前の語の品詞が「名詞」である場合は、対象とする語を文節の先頭の語と判定しない。

...
規定 きてい 規定 名詞 6 サ変名詞 2 * 0 * 0 ...
する する する 動詞 2 * 0 サ変動詞 16 基本形 2 ...
標準 ひょうじゅん 標準 名詞 6 普通名詞 1 * 0 * 0 ...
...

判定対象とする語「する」の品詞が「動詞」であり、直前の語「規定」の品詞が「名詞」であるため、文頭の先頭の語と判定しない。

10. 判定対象とする語の品詞が「名詞」「接頭辞」「動詞」「形容詞」「連体詞」「副詞」のいずれかである場合、文節の先頭の語と判定する。

11. それ以外は文節の先頭の語ではないとする。

3.3.5 アライメントに基づく句と句の類似度計算

PB と PF_1 を決定した後、その全ての句の組み合わせについて類似度を計算し、類似度が最大の組み合わせを後方並列句 pb、前方並列句 pf_1 と決定する。

$$(pb, pf_1) = \underset{pb_j \in PB,\ pf_{1k} \in PF_1}{\arg\max} sim_p(pb_j, pf_{1k}) \tag{3.2}$$

ここで $sim_p(a, b)$ は句の類似度を表す。

本研究では、句と句同士のアライメントに基づいて句の類似度を計算する。アライメントによって対応付けられた単語が似ていれば、句の類似度も高くする。単語間類似度は、表記の一致や意味クラスの類似度で算出する。一方、対応先がない単語があれば句の類似度を低くする。この際、末尾に近い単語ほど低い類似度を与える。これは、主辞の単語に対応先がない場合に大きなペナルティを与えるためである。以下に句の類似度の計算方法を詳述する。

a, b を類似度を求める句、つまり単語の集合とする。よって、

$$a = wa_1 \cdots wa_n,\ b = wb_1 \cdots wb_m$$

とする。ここで句 a と b のアライメント $ALIGN$ を式 (3.3) のように定義する。

$$ALIGN = \{a_k\}, \quad \text{但し,} a_k = (wa_i, wb_j) \text{ or } (wa_i, \phi) \text{ or } (\phi, wb_j) \tag{3.3}$$

語と語の対応関係（アライメント）として考えられる組み合わせは、(wa_i, wb_j)、(wa_i, ϕ)、(ϕ, wb_j) の 3 つのパターンである。(wa_i, wb_j) は wa_i と wb_j に 1 対 1 の対応関係があるこ

とを表わす。(wa_i, ϕ) 及び (ϕ, wb_j) はそれぞれ wa_i, wb_j に対応する語がないことを表わす。一般に可能なアライメントは複数存在する。その中からスコアが最大となるアライメントを求め、そのときのスコアを句 a、b の類似度と定義する（式 3.4）。

$$sim_p(a, b) = \max_{ALIGN} \ score_A(ALIGN) \tag{3.4}$$

また、アライメントのスコア $score_A(ALIGN)$ は以下のように定義する.

$$score_A(ALIGN) = \frac{1}{|ALIGN|} \sum_{a_k} score_a(a_k) \tag{3.5}$$

ALIGN のスコアは語と語の対応関係 a_k に対するスコアの和 $\sum_{a_k} score_a(a_k)$ の平均値とする。

次に、$score_a(a_k)$ を以下のように定義する。

$$score_a(a_k) = \gamma \cdot s\text{-}word(a_k) + (1 - \gamma) \cdot s\text{-}skip(a_k) \tag{3.6}$$

$score_a(a_k)$ は、$s\text{-}word(a_k)$ と $s\text{-}skip(a_k)$ の重み付き和とする．$s\text{-}word(a_k)$ では、対応関係にある単語が互いに類似しているほど高いスコアを与えるようにする。$s\text{-}skip(a_k)$ は対応関係がない場合（(wa_i, ϕ) 及び (ϕ, wb_j)）に対するスコア（ペナルティ）を与えるようにする。値は $s\text{-}word(a_k)$、$s\text{-}skip(a_k)$ ともに 0 から 1 までの間で算出する。重み γ は予備実験により 0.6 とした。

次に $a_k = (wa_i, wb_j)$ のとき $(wa_i \neq \phi,\ wb_j \neq \phi)$，$s\text{-}word(a_k)$ と $s\text{-}skip(a_k)$ を以下のように定義する。

$$s\text{-}word(a_k) \ = \begin{cases} 1 & \text{if } wa_i = wb_j \\ 0.9 & \text{if } wa_i \text{と } wb_j \text{がともに数字} \\ sim_w(wa_i, wb_j) \times 0.6 + 0.2 & \text{if } sim_w(wa_i, wb_j) \text{が計算可能なとき} \\ 0.1 & \text{if } wa_i \text{と } wb_j \text{の品詞が同じ} \\ 0 & \text{otherwise} \end{cases} \tag{3.7}$$

$$s\text{-}skip(a_k) \ = \ 1 \tag{3.8}$$

式 (3.7) における $sim_w(wa_i, wb_j)$ は、wa_i と wb_j がともに日本語語彙大系に登録されている場合に求める単語間の類似度であり、その計算式は式 (3.1) である。但し、表記が一致するものや品詞が一致するだけの場合と差をつけるために、類似度が 0.2 から 0.8 までの間になるようにスケーリングを行っている。一方，$s\text{-}skip(a_k)$ は最大値の 1 とする。

$a_k = (wa_i, \phi)$ もしくは (ϕ, wb_j) のとき，$s\text{-}word(a_k)$ と $s\text{-}skip(a_k)$ を以下のように定義する。

$$s\text{-}word(a_k) = 0 \tag{3.9}$$

$$s\text{-}skip(a_k) = \begin{cases} (1 - \frac{i}{n}) & \text{if } (wa_i, \phi) \text{ のとき} \\ \\ (1 - \frac{j}{m}) & \text{if } (\phi, wb_j) \text{ のとき} \end{cases} \tag{3.10}$$

式 (3.10) において、n、m は句 a、b の単語数を表わす。すなわち，$s\text{-}skip(a_k)$ は、末尾に近い位置で対応関係を持つ単語がない場合により低いスコア（ペナルティ）を与えるようにする。これは日本語の主辞は句の一番後ろにあるため、主辞に近い位置での $s\text{-}skip(a_k)$ に対しペナルティを大きくするのが妥当と考えたためである。一方、$s\text{-}word(a_k)$ は最低値の 0 とする。

さらに，法令文の特徴を考慮し，以下のヒューリスティクスを導入する。

法令文の特徴を考慮したヒューリスティクス

- 「第」「条」「項」「号」という語は法令文書において条文番号を表わす語である。条文番号は同じレベルの条文番号と一致すると解釈するのが普通であるため、これら４つの語に対しては、表記の一致のみを考慮する。つまり、「第」ならば「第」と、「項」ならば「項」と、表記が一致した時のみスコアを１とし、「条」と「号」などの組み合わせの時にはスコアを与えないこととする。特に、これらの語と他の語の組み合わせについては、日本語語彙大系による類似度 (式 (3.7) の３行目) をスコアの計算に用いない。これは不自然な単語の組に対して 0 より大きいスコアを与えないようにするためである。

- 「同」「前」「次」という単語が「条」「項」「号」の直前に出現するときは、「第」「条」「項」「号」及び数字の連続で構成される複数の語と対応付け、それに対し最大のスコアを与える。例えば、『第七条第一項第二号』と『同項第三号』という 2 つの句に対しては、「第七条第一」と「同」を対応付け、(第, 同)、 (七, 同)、 (条, 同)、(第, 同)、(一, 同) の 5 つの対応関係があるとみなす。このとき、それぞれ最大のスコア１をもつとみなし、それらの和である５をスコアとする。

また、式 (3.11) に示すように、全く同一の句の組のスコアは 0 とする。これは、同一の句が並列の関係にあることは基本的に起こり得ないからである。

$$sim_p(a, b) = 0 \qquad \text{if } a = b \tag{3.11}$$

以下にこの例外規則が適用される具体例を示す。

...厚生年金保険法第四十七条若しくは第四十七条の二の規定による...

この例文に対し、pf の候補が「第四十七条」、pb の候補も「第四十七条」となることがある。これらは同一の句であるため、アライメントに基づく類似度は最大の 1 となる。しかし、同じ句が並列関係にあるという解釈は通常ではありえない。そこで、同一の句が前方もしくは後方並列句として選ばれないようにするために、式 (3.11) を用いてスコアを 0 とする。類似度が最大となる句の組は、pf は「第四十七条」、pb は「第四十七条の二」となり、正しい解を得ることができる。

3.3.6 ２番目以降の前方並列句の検出

後方並列句 pb 及び前方並列句 pf_1 を検出後、２番目以降の前方並列句の範囲の候補を検出する。いま、後方並列句 pb と $i-1$ 個の前方並列句 $pf_l(1 \leq l \leq i-1)$ が検出されているとする。そのとき、i 番目の前方並列句 pf_i の検出を試みる。その際、以下の条件を満たす場合のみ検出を試みる。

- 既に検出された前方句 pf_{i-1} の直前が読点である。
- その読点の直前の語 w_y と $head$ の品詞が同じである。
- ($head$ の品詞が名詞のとき) $head$ と w_y の意味的類似度 (式 (3.1)) が 0.4 より大きい。

これらの条件を満たさない場合は並列構造解析を終了する。条件を満たしている場合、前方並列句の候補 $PF_i = \{...pf_{ik}[x,y]...\}$ を検出する。pf_{ik} の終点は常に y とする。一方 x は 3.3.4 項の手法と同様の処理を行う。この候補の中から既に検出されている前方並列句 pf_l 及び後方並列句 pb との類似度の和の最大のものを pf_i とする。

その計算式を (式 (3.12)) に示す。

$$pf_i = \mathbf{arg\ max}_{pf_{ik} \in PF_i} \sum_{l=1}^{i-1} (sim_p(pf_{ik}, pf_l) + sim_p(pf_{ik}, pb)) \tag{3.12}$$

3.3.7 解析例

提案手法による並列構造解析の例を示す。

この法律において、「保険料免除期間」とは、保険料全額免除期間、 保険料四分の三免除期間、保険料半額免除期間及び 保険料四分の一免除期間を合算した期間をいう。

まず最初に並列キー key として「及び」を検出する。

句 a :	保険	料	半額	免除	期間
	|	|	|	|	|
句 b :	保険	料	四分の一	免除	期間
	a_1	a_2	a_3	a_4	a_5
$s\text{-}word(a_k)$	1	1	0	1	1
$s\text{-}skip(a_k)$	1	1	1	1	1
$score_a(a_k)$	1	1	0.4	1	1

$$score_A = \frac{(1+1+0.4+1+1)}{5} = 0.8800$$

図 3.1: アライメントに基づく句の類似度の計算例 1

この法律において、「保険料免除期間」とは、保険料全額免除期間、 保険料四分の三免除期間、保険料半額免除期間**及び**(key) 保険料四分の一免除期間を合算した期間をいう。

前方並列句の主辞 $Head$ は「期間」となる。次に後方並列句の候補の検出を行う。後方並列句の候補は以下のとおりである。

後方並列句 (pb) の候補

- 保険料四分の一免除期間
- 保険料四分の一免除期間を合算した期間

次に前方並列句の候補を検出する。

前方並列句 (pf_1) の候補

- 保険料半額免除期間

これらの全ての組み合わせに対して句と句のアライメントに基づく類似度計算をする。この例では可能な句の組み合わせは 2 つである。それぞれについて句の類似度を求める。「保険料四分の一免除期間」と「保険料半額免除期間」という 2 つの句に対するアライメントとそのスコアの計算例を図 3.1 に、「保険料四分の一免除期間を合算した期間」と「保険料半額免除期間」の計算例を図 3.2 に示す。

図 3.1 と図 3.2 を比べてみると、「保険料四分の一免除期間」と「保険料半額免除期間」の組み合わせの方が、「保険料四分の一免除期間を合算した期間」と「保険料半額免除期間」の組み合わせより類似度が高いことがわかる。よって、「保険料四分の一免除期間」と「保険料半額免除期間」の組み合わせを後方、及び前方並列句として同定する。

句 a :	保険	料	ϕ	ϕ	半額	ϕ	免除	ϕ	期間
	|	|	|	|	|	|	|	|	|
句 b :	保険	料	四分の一	免除	期間	を	合算	した	期間
	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9
$s\text{-}word(a_k)$	1	1	0	0	0.44	0	0.44	0	1
$s\text{-}skip(a_k)$	1	1	0.667	0.5556	1	0.3333	1	0.1111	1
$score_a(a_k)$	1	1	0.2667	0.2222	0.6640	0.1333	0.6640	0.0444	1

$$score_A = \frac{(1+1+0.2667+0.2222+0.6640+0.1333+0.6640+0.0444+1)}{9} = 0.5549$$

図 3.2: アライメントに基づく句の類似度の計算例 2

この法律において、「保険料免除期間」とは、保険料全額免除期間、
保険料四分の三免除期間、保険料半額免除期間(pf_1) **及び**(key)
保険料四分の一免除期間(pb) を合算した期間をいう。

pf_1 の直前を探索すると、読点なので、更に前方に並列句があるかを確認する。読点の直前の語が「期間」であり、$Head$ との類似度が閾値の 0.4 を越えるので、更に前方並列句 pf_2 があると判断する。pf_2 の候補を以下に示す。

前方並列句 pf_2 の候補

- 保険料四分の三免除期間

検出された pf_2 の範囲の候補と既に同定された pf_1 と pb に対して類似度計算を行いその和を算出する (式 (3.12))。この場合、

- 「保険料四分の三免除期間」と「保険料半額免除期間 (pf_1)」
- 「保険料四分の三免除期間」と「保険料四分の一免除期間 (pb)」

の計算を行い、その和を算出する。一般に、前方並列句の候補は複数あると考えられるため、この計算で最大の値を算出した組み合わせを 2 番目の前方並列句 pf_2 として同定する。ここでは pf_2 は「保険料四分の三免除期間」となる。

この法律において、「保険料免除期間」とは、保険料全額免除期間、
保険料四分の三免除期間(pf_2)、保険料半額免除期間(pf_1) **及び**(key)
保険料四分の一免除期間(pb) を合算した期間をいう。

pf_2 の直前を探索すると、読点なので、更に前方の並列句があるかを確認する。読点の直前の語が「期間」であり、$Head$ との類似度が閾値の 0.4 を越えるので、更に前方並列

句 (pf_3) があると判断する。pf_3 の範囲の候補を示す。

前方並列句 pf_3 の候補

- 保険料全額免除期間

この場合、候補は１つしかないため、これを３番目の前方並列句とする。

> この法律において、「保険料免除期間」とは、保険料全額免除期間(pf_3)、
> 保険料四分の三免除期間(pf_2)、保険料半額免除期間(pf_1) **及び**(key)
> 保険料四分の一免除期間(pb) を合算した期間をいう。

pf_3 の直前を探索すると、読点なので、更に前方の並列句があるかを確認する。読点の直前が「は」であり、$Head$ との意味的類似度が 0.4 未満となるため、ここで探索を終了する。最終的に「保険料全額免除期間、保険料四分の三免除期間、保険料半額免除期間及び保険料四分の一免除期間」を並列構造として出力する。

3.4　階層的な並列構造解析の手法

本節では階層的な並列構造を検出する方法について述べる。本研究ではまず下位の並列構造を検出し、その後上位の並列構造を検出するといったように、ボトムアップかつ逐次的に並列構造を検出する。まず、一文内に複数の並列キーがある場合、それらを処理する順序を決める。その方法は 3.4.1 項で述べる。次に 3.3 節で説明した手法に若干の変更を加え、先に決められた並列キーの順序にしたがって並列構造を１つずつ検出する。変更の詳細を 3.4.2、3.4.3、3.4.4 項で述べる。最後に 3.4.5 項で階層的な並列構造の解析例を示す。

3.4.1　並列キーの検出順序

法令文における並列構造の特徴として、並列キーの並べ方に厳格なルールがあることが挙げられる。その特徴の詳細を以下に示す。

- 「若しくは」「又は」は選択的接続を表わす。いわゆる論理積の役割を果たす。
- 「及び」「並びに」は併合的接続を表わす。いわゆる論理和の役割を果たす。

これらの間に優先度が規定されており、またこれらの語の並べ方も規定されている。選択的接続の場合は最も外側の並列には「又は」を、それより内側の並列には「若しくは」を使用する。併合的接続の場合は最も内側の並列には「及び」を、それより外側の並列には「並びに」を使用する。法令文書では基本的に、このルールに基づいて階層的な並列構造を表現する。よってこれらの並列キーの優先度は、

及び、若しくは　＞　並びに、又は

となる。

本研究では、並列キーとして更に「と」「や」「かつ」「その他」を考慮する。これらの語は全て、併合的接続を表現する。これらの語の使われ方を実際の法令文書で確認してみると、最も上位の並列構造を構成すると解釈するのが自然だということがわかった。よって、これらの並列キーも含めた優先度を次のように定義する。

及び、若しくは　＞　並びに、又は　＞　と、や、かつ、その他

「及び」「若しくは」は優先度１である。よって一番最初にこれらの並列キーによって表される並列構造を検出する。「並びに」「又は」は優先度２である。よって二番目にこれらの並列キーを含む並列構造を検出する。「と」「や」「かつ」「その他」は優先度３である。よって最後にこれらの並列キーを処理する。同じ優先度を持つ並列キーが複数ある場合、文の先頭に近い位置の並列キーを含む並列構造から最初に検出する。上記のルールにしたがって、文中に複数の並列キーがあるとき、処理を行う順序を一意に決め、$key^1, key^2, ...key^z$ と表記する。また、処理を行う順序 z を並列キーのキー番号と呼ぶ。

3.4.2　後方並列句の検出方法の変更

3.3.3 項で後方並列句の候補を検出する手法について述べたが、階層的並列構造を取り扱うためにこのアルゴリズムを以下のように変更する。既にキー番号が小さい並列キーに対する下位の並列構造が同定されている場合、その並列構造の範囲内は後方並列句の終点としない。また、下位の並列構造の終点は後方並列句の終点の候補に必ず加える。尚、既に同定されている並列構造の範囲外は、通常の処理と同様に終点の候補を検出する。終点となる単語を後方へ探索するときに、読点、句点及び他の並列キーがあればそこで探索を止める。以下に例を示す。

... 第四項**並びに**第百五条第一項**及び**第四項の規定により 市町村が処理することとされている事務並びに...

既に優先度１の「及び」に対する並列構造が検出されているとする。その範囲を丸括弧で示す。

... 第四項**並びに**第百五条（第一項**及び**第四項）の規定により 市町村が処理することとされている事務並びに...

次に優先度２の「並びに」の後方並列句の候補を検出する。主辞 $Head$ は「並びに」の直前の語の「項」であり、その品詞は「名詞」である。よって探索する際は、文節の最後の自立語であり、かつ品詞が「名詞」のものを探索する。このとき、既に検出されている並列構造の範囲内（第一項及び第四項）は、終点の候補としない。また、既に検出されて

いる並列構造の最後の語の品詞が主辞の品詞と同じ（この場合は「名詞」）であれば終点の候補とする。「並びに」に対する後方並列句の終点の候補を下線で示す。

... 第四項**並びに**第百五 条（第一項**及び**第四 項）の 規定 により 市町村 が 処理 する こと とされている 事務 並びに...

この例では「事務」の後の「並びに」という並列キーに到達した時点で、探索を終了する。

3.4.3 前方並列句の検出方法の変更

3.3.4 項で前方並列句の候補を検出する手法について述べたが、階層的並列構造を取り扱うために、このアルゴリズムを以下のように変更する。既にキー番号が小さい並列キーに対する下位ので並列構造が同定されている場合、その並列構造の範囲内は前方並列句の始点の候補としない。また、下位の並列構造の始点は前方並列句の始点の候補に必ず加える。尚、既に同定されている並列構造の範囲外は、通常の処理と同様に始点の候補を検出する。始点となる単語を前方に探索するときに、読点、他の並列キー及び文頭に到達したらそこで探索を止める。以下に例を示す。

- 第十二条第一項**及び**第四項**並びに**...

  既に優先度 1 の「及び」に対する並列構造が検出されているとする。その範囲を丸括弧で示す。

- 第十二条（第一項**及び**第四項）**並びに**...

  次に優先度 2 の「並びに」の前方並列句の候補を検出する。既に検出されている並列構造の範囲内（第一項及び第四項）は、前方並列句の始点の候補としない。また、既に検出されている並列構造の先頭の語は始点の候補とする。「並びに」に対する前方並列句の始点の候補を下線で示す。

- 第 十二条（第 一項**及び**第四項）**並びに**...

  この例では文頭に到達したので探索を終了する。

3.4.4 句の類似度の計算方法の変更

3.3.5 項で述べたように、前方並列句と後方並列句の候補の全ての組について句の類似度を計算し、それが最大となる句の組を選択する。しかし、一方の句のみに下位の並列構造が含まれる場合は、3.3.5 項に示したアライメントに基づく類似度計算では、並列関係にある句の類似度が低く見積もられる。例えば以下の文について考察する。

被保険者でなかった者が第一号被保険者となった場合又は第二号被保険者若しくは第三号被保険者が第一号被保険者となった場合において、

並列キーが「又は」のときの前方及び後方並列句は

- pf = 被保険者でなかった者が第一号被保険者となった場合
- pb = 第二号被保険者若しくは第三号被保険者が第一号被保険者となった場合

である。しかし、後方並列句は下位の並列構造（第二号被保険者若しくは第三号被保険者）を含むため、後方並列句の長さは前方並列句よりも長い。そのため、両者の類似度が低く見積もられる可能性がある。

そこで、前方並列句及び後方並列句に下位の並列構造が含まれる場合、下位の並列構造をその後方並列句のみに置き換えて、類似度を計算する。上の例では後方並列句内の「第二号被保険者若しくは第三号被保険者」を「第三号被保険者」に置き換える。

- pf = 被保険者でなかった者が第一号被保険者となった場合
- pb = 第三号被保険者が第一号被保険者となった場合

この２つの句の類似度は、下位の並列構造を後方並列句に置き換える処理をしない場合と比べて高くなることが期待できる。また、前方及び後方並列句の候補の中に、深さ２以上の階層的な並列構造が含まれていれば、並列構造全体を後方並列句に置き換えるという処理を再帰的に繰り返す。

3.4.5 解析例

階層的並列構造の解析例を以下に示す。

第十二条第一項及び第四項並びに 第百五条第一項及び第四項の規定により 市町村が処理することとされている事務並びに 附則第九条の三の四の規定により市町村が処理することとされる事務 は、...

まず初めに優先度１の並列キーを文頭から探索する。「及び」(key^1) と「及び」(key^2) が検出される。

第十二条第一項**及び**(key^1) 第四項並びに 第百五条第一項**及び**(key^2) 第四項の規定により 市町村が処理することとされている事務並びに 附則第九条の三の四の規定により市町村が処理することとされる事務 は、...

次に、優先度２の並列キーを文頭から探索する。以下のように「並びに」(key^3) と「並びに」(key^4) が検出される。

第十二条第一項**及び**(key^1) 第四項**並びに**(key^3) 第百五条第一項**及び**(key^2) 第四項の規定により 市町村が処理することとされている事務**並びに**(key^4) 附則第九条の三の四の規定により市町村が処理することとされる事務 は、...

次に優先度３の並列キーを文頭から探索する。この例では優先度３の並列キーは存在しない。したがって、検出された並列キーとその処理の順序は以下のようになる。

「及び 」(key^1)**、「及び」**(key^2)**、「並びに」**(key^3)**、「並びに」**(key^4)

まず、「及び」(key^1) に対する後方並列句の範囲の候補と前方並列句の範囲の候補を検出する。

- 「及び」(key^1) に対する後方並列句の候補
  - 第四項
    ※ここでは「並びに」(key^3) の直前で探索を終了している。
- 「及び」(key^1) に対する前方並列句の候補
  - 第一項
  - 第十二条第一項

これらの句の全ての組み合わせについてアライメントに基づく類似度を計算し、前方並列句は「第一項」、後方並列句は「第四項」が選択され、（第一項及び第四項）

という並列構造が検出される。

第十二条 (第一項 (pf_1^1) **及び**(key^1) 第四項 (pb^1)) **並びに**(key^3) 第百五条第一項**及び**(key^2) 第四項の規定により 市町村が処理することとされている事務**並びに**(key^4) 附則第九条の三の四の規定により市町村が処理することとされる事務 は、...

次に、**及び**(key^2) に対する後方並列句の候補と前方並列句の候補を検出する。

- 「及び」(key^2) に対する後方並列句の候補
  - 第四項
  - 第十二条の規定
  - 第十二条の規定により市町村が処理
  - 第十二条の規定により市町村が処理すること

- 「及び」(key^2) に対する前方並列句の候補
  - 第一項
  - 第百五条第一項
    ※ここでは「並びに」(key^3) の直前で探索を終了している。

これらの語の全ての組み合わせについて類似度が最大となる句の組を求めると、(第一項及び第四項) という並列構造が検出される。

第十二条 (第一項 (pf_1^1) **及び**(key^1) 第四項 (pb_1^1)) **並びに**(key^3) 第百五条 (第一項 (pf_1^2) **及び**(key^2) 第四項 (pb_1^2)) の規定により 市町村が処理することとされている事務**並びに**(key^4) 附則第九条の三の四の規定により市町村が処理することとされる事務 は、...

次に、**並びに**(key^3) に対する後方並列句の候補と前方並列句の候補を検出する。下位の並列構造の範囲内は後方及び前方並列句の候補の境界になっていないことに注意していただきたい。

- 「並びに」(key^3) に対する後方並列句の候補

- 第百五条
- 第百五条第一項及び第四項
- 第百五条第一項及び第四項の規定
- 第百五条第一項及び第四項の規定により市町村が処理
- 第百五条第一項及び第四項の規定により市町村が処理すること

- 「並びに」(key^3) に対する前方並列句の候補
  - 第一項及び第四項
  - 第十二条第一項及び第四項

後方並列句の候補及び前方並列句の候補には、それぞれ既に同定されている下位の並列構造が存在しているため、これを後方並列句のみに置き換える。

- 「並びに」(key^3) に対する整形後の後方並列句の候補
  - 第百五条
  - 第百五条第四項　(←第百五条第一項及び第四項)
  - 第百五条第四項の規定　(←第百五条第一項及び第四項の規定)
  - 第百五条第四項の規定により市町村が処理　(←第百五条第一項及び第四項の規定により市町村が処理)
  - 第百五条第四項の規定により市町村が処理すること　(←第百五条第一項及び第四項の規定により市町村が処理すること)

- 「並びに」(key^3) に対する整形後の最終的な前方並列句の候補
  - 第四項　(第一項及び第四項)
  - 第十二条第四項　(第十二条第一項及び第四項)

これらの語の全ての組み合わせについて類似度が最大となる句の組を選択すると、(第百五条第一項及び第四項並びに第十二条第一項及び第四項）という並列構造が検出される。

(第十二条 (第一項 (pf_1^1) **及び**(key^1) 第四項 (pb^1))(pf_1^3)) **並びに**(key^3)
(第百五条 (第一項 (pf_1^2) **及び**(key^2) 第四項 (pb^2)) の規定により
市町村が処理することとされている事務(pb^3)) **並びに**(key^4)
附則第九条の三の四の規定により市町村が処理することとされる事務
は、...

次に, **並びに**(key^4) に対する後方並列句の候補と前方並列句の候補を検出する。

- 「並びに」(key^4) に対する後方並列句の候補
  - 附則第九条の三の四の規定により市町村が処理することとされる事務
    ※これは読点に到達したため探索を終了したことと、**並びに**(key^4) の前方並列句の主辞である「事務」と表記が完全一致した語のみを後方並列句の終点としたためである。

- 「並びに」(key^4) に対する前方並列句の候補
  - 事務
  - されている事務
  - 処理することとされている事務
  - 市町村が処理することとされている事務
  - より市町村が処理することとされている事務
  - 規定により市町村が処理することとされている事務
  - 第十二条第一項及び第四項並びに第百五条第一項及び第四項の規定により市町村が処理することとされている事務

  前方並列句の候補には、既に検出されている並列構造が存在しているため、これを並列構造の後方並列句に置き換える。まず (pf_1^3, key^3, pb^3) という並列構造を pb^3 に置き換え、更に pb^3 内には (pf_1^2, key^2, pb^2) という並列構造があるので、これも pb^2 に置き換える。

- 「並びに」(key^4) に対する整形後の前方並列句の候補
  - 事務
  - されている事務
  - 処理することとされている事務
  - 市町村が処理することとされている事務
  - より市町村が処理することとされている事務
  - 規定により市町村が処理することとされている事務
  - 第百五条第四項の規定により市町村が処理することとされている事務　(←第十二条第一項及び第四項並びに第百五条第一項及び第四項の規定により市町村が処理することとされている事務)

  これらの句の全ての組み合わせの中から類似度が最大になるものを選択し、(第十二条第一項及び第四項並びに第百五条第一項及び第四項の規定により市町村が処理す

ることとされている事務並びに附則第九条の三の四の規定により市町村が処理することとされる事務) という並列構造が得られる。

(（第十二条（第一項(pf_1^1) **及び**(key^1)第四項(pb^1)）(pf_1^3)） **並びに**(key^3)
（第百五条（第一項(pf_1^2) **及び**(key^2)第四項(pb^2)）(pb^3)） の規定により
市町村が処理することとされている事務(pf_1^4)) **並びに**(key^4)
(附則第九条の三の四の規定により市町村が処理することとされる事務(pb^4))
は、...

これは、正しい解析結果である。

第4章　評価

本章では、提案手法の評価実験について述べる。

4.1　実験データ

ここでは実験に用いたデータについて述べる。

4.1.1　評価用コーパス

国民年金法の法令文を評価用コーパスとして使用した。国民年金法の法令文の先頭から２００文を開発用コーパスとして使用し、２０１から３００文までを評価用コーパスとして使用した。これら３００文には全て人手で正解データを付与した。また、開発用データ、評価用データにおける並列構造の数は、階層的な並列構造を構成するものも含めて、それぞれ１８８、６８である。

以下に正解データの例を示し、説明をする。

- 正解データの例１

...
父母 ふぼ 父母 名詞 6 普通名詞 1 * 0 * 0 ”代表表記:父母/ふぼ カテゴリ:人 ドメイン:家庭・暮らし COORD:01,forward,3”
、 、 、 特殊 1 読点 2 * 0 * 0 NIL
孫 まご 孫 名詞 6 普通名詞 1 * 0 * 0 ”代表表記:孫/まご 漢字読み:訓 カテゴリ:人 ドメイン:家庭・暮らし COORD:01,forward,4”
、 、 、 特殊 1 読点 2 * 0 * 0 NIL
祖父母 そふぼ 祖父母 名詞 6 普通名詞 1 * 0 * 0 ”代表表記:祖父母/そふぼ カテゴリ:人 ドメイン:家庭・暮らし COORD:01,forward,5”
又は または 又は 助詞 9 接続助詞 3 * 0 * 0 ”COORD:01,key”
兄弟 きょうだい 兄弟 名詞 6 普通名詞 1 * 0 * 0 ”代表表記:兄弟/きょうだい カテゴリ:人 ドメイン:家庭・暮らし COORD:01,backward,1”
姉妹 しまい 姉妹 名詞 6 普通名詞 1 * 0 * 0 ”代表表記:姉妹/しまい カテゴリ:人 ドメイン:家庭・暮らし COORD:01,backward,1”

であって であって だ 判定詞 4 * 0 判定詞 25 デアル列タ系連用テ形 26 NIL
...

この例では、「又は」が並列キーであることが、「又は」の行末の COORD:01,key というダグで確認できる。01 は並列構造の ID である。このように COORD... と書かれたものが正解データのタグとなる。「又は」の直前の語の「祖父母」を見てみると COORD:01,forward,5 とタグ付けされている。これは、01 番目の並列構造の 5 番目の前方並列句という意味である。同様に「孫」は 4 番目の前方並列句、「父母」は 3 番目の前方並列句ということになる。また、後方並列句に関しては、「又は」の直後の語の「兄弟」の行末に COORD:01,backward,1 とタグ付けされている。これは、01 番目の並列キーの 1 番目の後方並列句という意味である。但し、本研究では基本的に後方並列句は一つだけしかとらないとしている。そしてその次の語である「姉妹」も行末に COORD:01,backward,1 とタグ付けされている。その次の語の「であって」の行末は NIL なので後方並列句ではないことが確認できる。よって、後方並列句の範囲は「兄弟」「姉妹」ということになる。

次に、階層的な並列構造の場合の正解データの例を示し、説明をする。

- 正解データの例 2

  ...
  一 いち 一 名詞 6 数詞 7 * 0 * 0 ”COORD:01,forward,1;02,backward,1;04,forward,1”
  項 こう 項 接尾辞 14 名詞性名詞助数辞 3 * 0 * 0 ”代表表記:項/こう 内容語 カテゴリ:数量 COORD:01,forward,1;02,backward,1;04,forward,1”
  及び および 及び 助詞 9 接続助詞 3 * 0 * 0 ”COORD:01,forward,1;02,backward,1;04,key”
  第 だい 第 接頭辞 13 名詞接頭辞 1 * 0 * 0 ”代表表記:第/だい COORD:01,forward,1;02,backward,1;04,backward,1”
  四 よん 四 名詞 6 数詞 7 * 0 * 0 ”COORD:01,forward,1;02,backward,1;04,backward,1”
  ...

この例では、「及び」が並列キーとなっており、COORD:01,forward,1;02,backward,1;04,key とタグ付けされていることから、04 番目の並列構造の並列キーということが確認できる。また、01 番目の並列構造の１つ目の前方並列句の一部であり、かつ 02 番目の並列構造の１つ目の後方並列句の一部であることが確認できる。「及び」の前後に出現する語も複数の並列構造の前方並列句若しくは後方並列句の一部としてタグ付けされている。階層的な並列構造はこのようにタグ付けされている。尚、この並列構造の ID は、3.4.1 節で述べたような並列キーの処理の順序とは異なる。

4.2 実験方法

ここでは実験方法について述べる。提案手法を実装したシステムと既存の構文解析器である KNP ver. 3.01[1] との比較実験を行う。但し、KNP の出力においては、本研究で並列キーと定義していないものでも、並列キーとみなし、並列構造を構成している場合がある。図 4 は実際の KNP の出力例である。

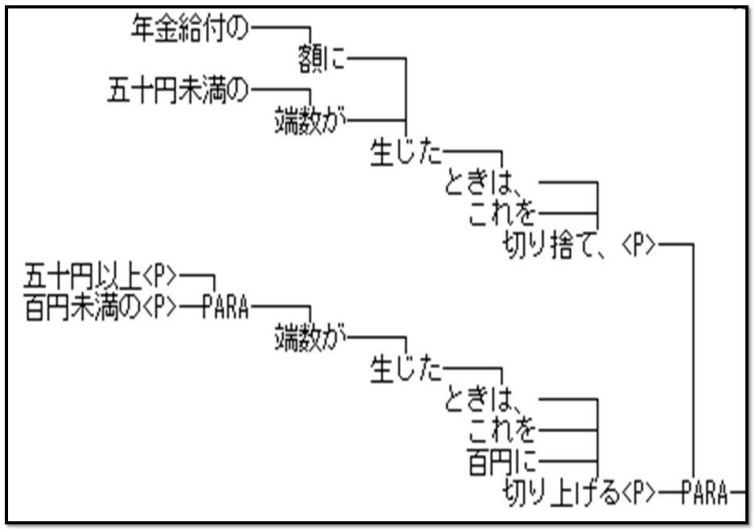

図 4.1: KNP の出力の例

この例では、動詞句並列として、「切り捨て、」の節と「切り上げる」の節で並列関係を構成している。(記号 ⟨P⟩ で並列関係を表わしている。) しかし、この場合においては、本研究で取り扱っている並列キーは使用されていない。よって、このような並列構造に関しては評価の対象から除外する。同様に、「切り上げる」の節内の「五十円以上」と「百円未満の」も並列関係を構成しているが、この例でも本研究で扱う並列キーは使用されていないため、評価の対象とはしていない。

4.2.1 評価尺度について

本研究で扱う評価尺度としては、精度、再現率、F 値を使用する。これらは検出された並列構造に対して正解データと比較した際に、検出された並列句の範囲が完全一致している場合と、片方がもう片方を包含している部分一致の場合とを考慮する。以下に精度、再現率、F 値の説明をする。

- **精度** ($Precision$)**：システムの出力における、正しい要素の割合**

$Precision =$

$$\frac{\text{システムが出力する正解の要素数}}{\text{システムが出力する正解の要素数} \quad + \quad \text{システムが出力する不正解の要素数}} \tag{4.1}$$

一般に、「システムが出力する正解の要素」は$True\text{-}positive(TP)$と表現される。「システムの出力する不正解の要素」は、$False\text{-}positive(FP)$と表現される。よって、精度は以下のようにも定義される。

$$Precision = \frac{TP}{TP + FP} \tag{4.2}$$

精度は検索エラーを評価する尺度である。

- **再現率**$(Recall)$**：要素がシステムによって正しく検出された割合**

$Recall =$

$$\frac{\text{システムが出力する正解の要素数}}{\text{システムが出力する正解の要素数} \quad + \quad \text{システムによって検出されない要素数}} \tag{4.3}$$

一般に、「システムによって検出されない要素」は、$False\text{-}negative(FN)$と表現される。よって、再現率は以下のようにも定義される。

$$Recall = \frac{TP}{TP + FN} \tag{4.4}$$

再現率は検索における漏れを表わす評価尺度である

- **F 値**$(F\text{-}measure)$**：精度と再現率を１つにまとめたもの**
精度と再現率の調和平均として定義される。

$$F\text{-}measure = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{4.5}$$

上記の定義における要素とは、「並列構造」「並列キー」「前方並列句」「後方並列句」のいずれかである。すなわち、並列構造全体がどれだけ正しくできているかを評価するとともに、並列構造を構成する並列キー、前方並列句、後方並列句の解析結果も個別に評価する。また、システムが出力した要素が正解の要素と範囲が完全に一致しているときを正解とみなす場合と、システムの出力する要素と正解の要素の範囲が交差しない（一方の範囲がもう一方を包含している）ときを正解とみなす場合の両方で評価する。前者を「完全一致」による評価、後者を「部分一致」による評価と呼ぶ。

4.2.2 実験結果

本項では実験結果について述べる。提案手法と KNP との精度、再現率、F 値の実験結果を表 4.1 に示す。表中の P は精度を、R は再現率を、F は F 値をそれぞれ表わす。また、開発データと評価データにおける結果、及び正解の基準を「完全一致」にしたときと「部分一致」にしたときの結果を示している。

4.3 考察

ここでは得られた実験結果について考察する。「完全一致」の並列構造解析の F 値は、開発データで 0.62、評価データで 0.50 であることから、提案手法の解析精度は十分に高いとはいえない。しかし、「部分一致」に対しては開発データで 0.97、評価データで 0.87 であることから、十分な数値が得られた。今後は完全一致の解析精度を向上させるための手法を考案する必要がある。並列キーの F 値は、開発データで 0.99、評価データで 0.93 であることから比較的高い値を得ることができた。この解析誤りは主に「と」を解析する際に生じている。すなわち、並列関係を表わさない格助詞の「と」を誤って並列キーとして検出するときがあった。前方・後方並列句の「完全一致」の F 値は開発データで 0.78、評価データで 0.65、「部分一致」の F 値は開発データで 0.98、評価データで 0.92 であった。いずれも並列キーよりも値が低いが、前方並列句・後方並列句の範囲の同定は並列キーよりもはるかに難しいと言える。前方並列句と後方並列句の F 値を比較すると、開発データでは前方並列句の F 値の方が 0.05 ほど高いが、評価データでは両者の F 値に大きな差はない。提案手法と KNP を比較すると、全ての項目で提案手法が KNP を上回った。この結果は、提案手法が法令文書の並列構造の特徴を考慮できたことを示している。また、開発データと評価データを比べてみると、評価データの結果が全ての項目において開発データにおける評価値以下である。このことからまだ十分に法令文書の特徴を考慮しきれていないことがわかる。

表 4.1: 実験結果

		開発データ		評価データ	
		提案手法	KNP	提案手法	KNP
並列構造 (完全一致)	P	0.62	0.49	0.47	0.27
	R	0.62	0.36	0.54	0.26
	F	0.62	0.41	0.50	0.26
並列キー (完全一致)	P	0.99	0.96	0.87	0.84
	R	0.99	0.70	0.99	0.81
	F	0.99	0.81	0.93	0.82
後方並列句 (完全一致)	P	0.81	0.71	0.60	0.40
	R	0.81	0.52	0.68	0.39
	F	0.81	0.60	0.64	0.40
前方並列句 (完全一致)	P	0.77	0.76	0.62	0.59
	R	0.75	0.46	0.69	0.48
	F	0.76	0.58	0.65	0.53
前方・後方並列句 (完全一致)	P	0.79	0.74	0.61	0.50
	R	0.78	0.49	0.69	0.44
	F	0.78	0.59	0.65	0.47
並列構造 (部分一致)	P	0.97	0.79	0.82	0.79
	R	0.97	0.58	0.93	0.77
	F	0.97	0.67	0.87	0.78
後方並列句 (部分一致)	P	0.99	0.96	0.87	0.84
	R	0.99	0.70	0.99	0.81
	F	0.99	0.81	0.93	0.82
前方並列句 (部分一致)	P	0.99	0.96	0.86	0.82
	R	0.97	0.96	0.69	0.67
	F	0.98	0.72	0.91	0.74
前方・後方並列句 (部分一致)	P	0.99	0.96	0.87	0.83
	R	0.98	0.63	0.97	0.73
	F	0.99	0.76	0.92	0.78

4.3.1 KNP との比較

KNP との比較の結果、提案手法は KNP よりも完全一致、部分一致の両方で精度、再現率、F 値の全てで上回っていることがわかる。大きな要因としては以下の 3 つがある。

1. 法令文書で頻出する並列キーを、KNP では並列キーと認識しないことが多い。
2. KNP では法令文書に対して更に前方並列句の検出が出来ていないことが多い。
3. 法令文書に特有の階層的な並列構造を構成するルールに対応できていないことである。

以下に例を示す。太字は並列キーを、下線は並列句を表わす。

1. **KNP で並列キーを認識しない例**

(a) 「その他」の例

KNP の出力
（... 給付その他の老齢...）

提案手法の出力
（... 給付**その他**(key) の老齢...）

(b) 「並びに」の例

KNP の出力
（... 事項並びに氏名...）

提案手法の出力
（... 事項**並びに**(key) 氏名...）

(a) では「その他」を並列キーとして検出しないのに対し、提案手法では検出できている。開発用コーパスの２００文を対象に確認したところ、KNP が「その他」を並列キーと認識している例は見当たらなかった。「その他」が並列関係を表わすのは法令文特有の特徴であるといえる。

(b) では、KNP は「並びに」を並列キーとは認識しないのに対し、提案手法では検出できている。KNP で検出に失敗しているのは、JUMAN では「並びに」が「並び」と「に」の二語に分割されていることが原因であると考えられる。開発用コーパスの２００文を対象に確認したところ、「並び」と「に」が二語に分かれているときにこれらをを並列キーと認識している例は見当たらなかった。

2. **KNP での２つ以上の前方並列句の検出失敗の例**

KNP の出力
（... 保険料全額免除期間、保険料四分の三免除期間、保険料半額免除期間(pf_1) **及び**保険料...）

提案手法の出力
（...保険料全額免除期間(pf_3)、保険料四分の三免除期間(pf_2)、保険料半額免除期間(pf_1) **及び**保険料...）

下線を引いた範囲（「保険料半額免除期間」）が前方並列句に当たるが、この例では

更に前方並列句（「保険料四分の三免除期間」）が存在するので、その範囲も検出すべきである。しかし、KNP では検出できていない。このように KNP では法令文書においては２つ目以降の前方並列句の検出で失敗することが多いことが確認された。

3. **KNP での階層的な並列構造での検出失敗の例**

   KNP の出力
   （...、障害(pf_1^1) **若しくは**(key^1)死亡$(pb^1)(pf_1^2)$ **又は**(key^2)これらの直接の原因(pb^2) となった...）

   提案手法の出力
   （...、障害(pf_1^1) **若しくは**(key^1)死亡$(pb^1)(pf_1^2)$ **又は**(key^2)これら(pb^2) の直接の原因となった...）

   KNP では「障害若しくは死亡」と「死亡又はこれらの直接の原因」が並列構造として検出されているが、これらには上位-下位の関係はない。よって、階層的な並列構造の検出に失敗していると言える。

4.3.2 考察

本項では、提案手法の解析誤りの原因について考察する。提案手法では、２つ目以降の前方並列句の同定に失敗している場合が多く見受けられた。その原因としては、前方並列句と後方並列句の長さのが大きく異なるために、１つ目の前方並列句が正確に検出できていないことが挙げられる。例えば以下のような例である。

- **提案手法によって解析された並列構造**
  第一項第一号に規定する給付が、恩給法による増加恩給、同法第七十五条第一項第二号に規定する扶助料(pf_1) **その他**(key)政令で定めるこれら(pb) に準ずる給付であって、...

- **正解の並列構造**
  第一項第一号に規定する給付が、恩給法による増加恩給(pf_2)、同法第七十五条第一項第二号に規定する扶助料(pf_1) **その他**(key)政令で定めるこれらに準ずる給付(pb) であって、...

２つ目の前方並列句同定には、１つ目の前方並列句が完全に同定されていなければならないという条件がある。正解では「同法第七十五条第一項第二号に規定する扶助料」と「政令で定めるこれらに準ずる給付」が並列関係にあるが、句の長さが大きく異なるため、長さがほぼ等しい「規定する扶助料」と「政令で定めるこれら」との類似度の方が高くなり、解析に失敗している。また、「規定する扶助料」の前には読点がないので、２つ目以

降の前方並列句は存在しないと判定している。よって、１つ目の前方並列句を正確に同定できるように、句の長さを考慮した方法を考える必要がある。

また、動詞による並列においては、動詞節が並列関係にあるケースも多く確認できた。節同士の並列だと必然的に長くなるので、前方並列句の候補の数も増えることから、並列構造の検出が困難になると考えられる。節の場合には主語の次にくる助詞が一つの手がかりとなる可能性が高い。例えば、

...、名目手取り賃金変動率が一以上となり、**かつ**、調整率が一以下となるとき...

という例文の場合、前方並列節の主語である「名目手取り賃金変動率」の次の語である助詞が「が」である。後方並列節の主語である「調整率」の次の語である助詞が「が」である。この「が」に着目した処理をすることで、節の同定が可能になると考えられる。まれではあるが、並列節の範囲内に読点を含む場合も存在する。提案手法では、読点が出現した時点で前方並列句や後方並列句の探索を打ち切るため、並列句が読点を含む場合は必ず解析に失敗する。節における並列構造解析では、読点を跨ぐことも考慮する必要がある。長さのバランスを考慮するために、節の並列と句の並列とを区別して考える必要があるだろう。

また、現在の提案手法では取り扱っていないが、指示語や係り受け関係も並列構造解析の際に考慮すべきである。以下に具体例を示す。

- 係り受け関係を考慮しないことが原因のとき

  提案手法の出力
  (被保険者は、厚生労働省令の定めるところにより、その 資格の 取得(pf_1^1) **及び**(key^1) 喪失$(pb^1)(pf_1^2)$ **並びに**(key^2)種別の変更(pb^2) に関する事項...)

  この解釈では、「その」は「取得」と「喪失」に係る。

  係り受け関係：「その」→「取得」、「その」→「喪失」

  正解
  (被保険者は、厚生労働省令の定めるところにより、その資格の 取得(pf_1^1) **及び**(key^1) 喪失$(pb^1)(pb^2)$ **並びに**(key^2)種別の変更(pb^2) に関する事項...)

  この解釈では、「その」は「資格」に係る。

  係り受け関係：「その」→「資格」

  一般に、このような係り受け関係も考慮し、どちらの係り受け関係が尤もらしいかを判定しないと、正しい解釈の並列構造を検出できない。特に指示詞については、

指示詞が指すものを同定し、係り受け関係を考慮する必要がある。この例の場合、「その」は「被保険者」を指す。したがって、提案手法が出力する並列構造、及び正解の並列構造の解釈は以下のようになる。

提案手法の解釈
「被保険者」の「取得」、「被保険者」の「喪失」

正解の解釈
「被保険者」の「資格」

後者の方がもっともらしいといえる。このように、並列構造に関連した語の係り受け関係や指示語の指す対象を同定することで、並列構造解析の性能を向上させることができる。

更に、並列キーではない「と」を誤って並列キーの「と」と認識してしまう例が評価データでは頻出した。

- **並列キーではない「と」を誤って検出する例**

  ... その者の死亡の当時その 子(pf_1^1) **と**(key^1)生計(pb_1^1) を同じくしていたもの...

  JUMAN ではこの「と」は「接続助詞」として解析されるために提案手法で並列キーとして検出されるが、本来は「格助詞」として解析されるべきである。「と」は並列関係を表わすときとそうでない（格助詞として働く）ときがあり、両者を正確に識別する必要がある。

第5章　おわりに

本論文では、法令文書の特徴を考慮した並列構造解析の手法を提案した。本章では、本研究で得た知見と今後の課題について述べる。

5.1　まとめ

提案手法のうち特に重要な要素技術を以下に挙げる。

1. 前方並列句、後方並列句の範囲の同定
   (a) 法令文書の特徴を踏まえ、後方又は前方並列句の終点又は始点の候補となる語を厳格に規定した。
2. 句の類似度計算
   (a) 法令文書の特徴である「条」「項」「号」などの語を考慮して計算を行った。
   (b) 同じ句同士の類似度は最低値を返すことにした。
3. 階層的な並列構造解析
   (a) 階層的な並列構造解析をするために、下位の並列構造から上位の並列構造へ順次解析するための方法として、並列キーの検出に優先順位を設けた。
   (b) 句と句の長さのバランスを考慮するために、上位の並列構造を解析する際は、下位の並列構造全体を考慮するのではなく、後方並列句のみを考慮した。

1 (a) では後方及び前方並列句の候補を決める際に、厳格にルールを定めたことで明らかに正しくない句の候補を除外し、句の類似度計算における計算量を減少させた。この処理では同時に JUMAN の解析誤りにも対応しているため、解析精度向上に貢献した。2 (a) では「条」「項」「号」といった語を含む並列句の同定処理であるが、これらの語を含むことで句の長さのバランスが崩れることが頻繁に生じていた。法令文書ではこれらの語は条文番号を表わすことが多いことから、これらの語に特化した手法を提案した。2 (b) では並列構造の類似度を計算する際に、前方並列句と後方並列句が全く同じ場所は類似度が最大となる。しかし、一般に、同じ句同士が並列の関係になることは考えられないため、その場合はスコアを最低値とする例外処理を行った。これにより、同一の句が並列関係に

なるような並列構造が検出されることを回避することができた。3 (a) では階層的な並列構造に対応するために並列キーの検出の順序を決めた。下位の並列構造をつくる並列キーを先頭から逐次検出することで、ボトムアップ式に下位の並列構造から並列構造を同定することができた。3 (b) では上位の並列構造を解析する際に生じる、句の長さのアンバランスに対応するための手法を提案した。これは、上位の並列構造の前方か後方のどちらか一方に下位の並列構造が含まれている場合、上位の並列句同士の長さのバランスが崩れるためである。更に、一般的に下位の並列構造に複数の前方並列句が含まれていることもあり、この場合は更に長さのバランスが崩れる。これに対し、下位の並列構造をその後方並列句のみに置き換えて句の類似度を計算した。この処理により、より正確に並列構造が検出できるようになった。

5.2 今後の課題

本研究では法令文書の特徴を考慮した並列構造解析を行い、その精度を向上させることを目標としてきた。結果として KNP より大幅に F 値を向上させることができた。しかしながら、並列構造全体での「完全一致」の F 値は約 0.50 であるため、決して高い数値とは言えない。以下に問題点を挙げる。

1. 前方並列句と後方並列句の長さが大きく異なるときに解析に失敗する。

2. 係り受け関係や指示語の係り先を考慮していない。

大きな問題点としてはこの 2 点が挙げられる。並列句の長さのバランスがとれていない時に解析に失敗することが多いが、前方並列句の同定に失敗すると、更に前方にも並列句がある場合に対応できない。また、今後は係り受け関係や指示語の照応先を考慮した手法を提案し、並列構造の同定をより正確なものにする必要がある。

謝辞

本研究を進めるに当たって、白井清昭准教授、島津明教授、中村誠助教、Nguyen Minh Le 助教には数多くのご教示を頂きました。また、白井研究室・島津研究室の皆様方には、本研究に関する貴重なご支援を頂きました。この場を借りて感謝申し上げます。

参考文献

[1] 黒橋禎夫, 長尾眞、並列構造の検出に基づく長い日本語文の構文解析, 情報処理, Vol. 1, No. 1, pp35–57, 1994.

[2] Kawahara, D.,Kurohashi, S., Probabilistic Coordination Disambiguation in a Fully-Lexicalized Japanese Parser, EMNLP-CoNLL, pp306–314, 2007.

[3] 河原大輔, 黒橋禎夫、Web から獲得した大規模格フレームに基づく構文・格解析の統合的確率モデル, 言語処理学会 第 12 回年次大会, pp1111–1114, 2006.

[4] Hara, K.,Shimbo, M.,Okuma, H.,Matsumoto, Y., Coordinate Structure Analysis with Global Structural Constraints and Alignment-Based Local Features, Proceedings of the 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP, pp967–975, Suntec, Singapore,2-7 August 2009.

[5] 加藤竜太, 小川康弘, 戸山勝彦, 構文情報タグ付き法律文コーパスにおける並列表現の分析とタグ付け誤りの修正, 言語処理学会第１６回年次大会講演論文集, pp490–493, 2010.3.

[6] Daniel M. Bikel. Multilingual statistical parsing engine version 0.9.9c., http://www.cis.upenn.edu/ dbikel/sowtware.html., 2005.

[7] Eugene Charniak and Mark Johnson. Coarse-to-fine n-best parsing and MaxEnt discriminative reranking, Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics(ACL 2005), pages 173-180, Ann Arbor, Michigan, USA, 2005.

[8] Masashi Shimbo and Kazuo Hara. A discriminative learning model for coordinate conjunctions, Proceedings of Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNNL 2007), pages 610-619, Prague, Czech Republic, 2007.

[9] Sadao Kurohashi, Toshihisa Nakamura, Yuji Matsumoto and Makoto Nagao. Improvements of Japanese Morphological Analyzer JUMAN, In Proceedings of The International Workshop on Sharable Natural Language Resources, pp.22-28 1994.8.

[10] 上田章, 笠井真一, 条例規則の読み方・作り方, 学陽書房, 2003.

[11] 池原悟, 宮崎正弘, 白井諭, 横尾照男, 中岩浩巳, 小倉健太郎, 大山芳史, 林良彦. 日本語語彙大系, 岩波書店, 1999.