

日本語文章の推敲支援を目的とした並列構造の指摘

菅沼 明 宮崎 晋[†]

九州大学 大学院システム情報科学研究科

[†] 現在, 富士通情報通信システムズ

日本語文章中における並列構造とは、1 文中に同等の機能を持つ複数個の文節列が並んだ構造である。並列構造を含む文は、どの文節列とどの文節列が並列であるのかという認識において曖昧性を含みやすく、このため文の書き手と読み手の間に食い違いが生じることがある。本研究は、述語並列の並列構造を推定し、書き手と読み手の間に食い違いが生じやすい並列構造を文章推敲の立場から指摘することを目的としている。本稿では、ユーザに煩わしさを感じさせない待ち時間で処理を行うために、字面の情報を主に利用した述語並列の構造推定法について述べている。

Extracting Method of Coordinate Structures in Japanese Documents for a Writing Tool

Akira SUGANUMA and Susumu MIYAZAKI

Graduate School of Information Science and Electrical Engineering

Kyushu University

6-1 Kasuga-koen, Kasuga-shi, Fukuoka-ken, 816 Japan

e-mail : suga@is.kyushu-u.ac.jp

In Japanese documents, sentences with some coordinate structures are often ambiguous when two components of the coordinate structure are recognized. It is often the case that the reader cannot recognize the coordinated components which the writer intends. As a result, some communication gaps occur between the writer and the reader. In this paper, we describe the method for extracting coordinate structures consisting of clauses and estimating the coordinated components. Our method analyzes a Japanese document with textual information mainly, so that the writer can comfortably find information of coordinate structures in a few seconds.

1. はじめに

文章を推敲する際には、書き手が文章を読み進めながら問題となる箇所を探し出し、書き直すという作業を繰り返す。書き手は文章を書く時に、読み手に正しく伝わると思い込んでしまうことがある。この状況では、書き手が自分自身の文章を推敲する場合、間違いを間違いだと気が付かない可能性がある。他人であればすぐに気が付くようなことでも、本人には、なかなか気が付きにくいことがある。そのため、客観的な目で問題となる箇所を探し出す過程を計算機で支援できれば、人間は計算機が指摘したものを吟味して推敲を行えばよいので、人間の作業を軽減できる。

日本語文章中における並列構造とは、1文中に同等の機能を持つ複数個の文節列が並んだ構造である。並列構造を含む文は、どの文節列とどの文節列が並列であるのかという認識において曖昧性を含みやすく、このため並列構造を誤って認識してしまうことがある。これは、文章推敲の立場から考えると、文の書き手と読み手の間に食い違いが生じることにつながる。

本研究の目的は、文章推敲の立場から考えて並列構造を推定し、書き手と読み手の間に、曖昧さによる食い違いが生じやすい箇所を書き手に指摘することである。この目的に沿って、読み手が受け取りやすい並列構造を推定する方法を構築する。また、書き手に指摘する際に、食い違いが生じると考えられる箇所を示すだけではなく、どのように食い違いが生じると考えられるのかを示すようにできれば、より良いと考える。このために、曖昧な並列構造をどのように書き手に提示すれば良いか考える。今回は、推定した並列要素を書き手に示し把握してもらうことで、曖昧さを書き手に理解してもらうことを考えた。

2. 日本語文中の並列構造

並列構造とは、1文中に同等の機能を持つ複数個の文節列を並べた構造である。並列構造を含む文は、どの文節列とどの文節列が並列しているかという並列要素の認識において曖昧性を含みやすい。そのため並列構造を誤って認識してしまうことが多く、後の構文解析や意味解析

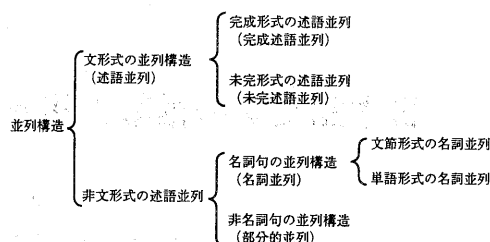


図 1: 本研究における並列構造の分類

にも解析誤りが生じてしまう。また、並列構造を含む文を文章推敲の立場から考えても、文の書き手と読み手の間に食い違いが生じやすい。

2.1 並列構造の分類

日本語における並列構造は、並列の単位となっている文節列によって分類することができる。本研究では図1のように分類する。

文形式とは、末尾に述部を備え、まとまった叙述機能を持つ文節列である。この文形式の文節列を並列要素として持つ並列構造を述語並列と呼ぶ。また、述語並列は、構成面に関して完成形式の述語並列と未完形式の述語並列に分けることができる。完成形式とは述部が用言として成り立っている節であり、これらを並列要素として持つ述語並列を完成述語並列と呼ぶ。逆に、未完形式とは述部が用言として成り立っていない節であり、これらを並列要素として持つ述語並列を未完述語並列と呼ぶ。未完形式は、サ変動詞の「する」が省略されたサ変名詞がサ変動詞の役割をする場合に多く見られる。

非文形式とは、文形式から述部を含む文末側の単語列を取り去って残った単語列のうち、文形式でない単語列である。また、非文形式の並列構造は、構成面に関して名詞句になるものと、そうでないものとに分けることができる。名詞句または単独の名詞を並列要素として持つ並列構造を名詞並列と呼ぶ。非名詞句とは、節の述部を除いた文節列の一部分であり、これを並列要素として持つ並列構造を部分的並列と呼ぶ。

本稿では、述語並列の並列構造の推定法に関して述べる。名詞並列、部分的並列に関しては、文献[2]を参照されたい。述語並列の例を図2に挙げる。(1)と(2)が完成述語並列であり、(3)

- (1) 待ち合わせ方式においては、「ロック要求が拒否され」、かつ「デッドロック要求が検出されない」時、…
- (2) 「コミュニケーションは管理組織の中で最も重要な基本的な事項である」が、「すぐれた管理は良いコミュニケーションのみで作られるものではない」。
- (3) 「第1日と次の日の半日で計画内容を吟味」、「あとの半日で明確に対象を選定」、「最後に経済計算する手順を示す」。
- (4) アルゴリズムを「調査」、「適用した」結果を…

図 2: 述語並列の例

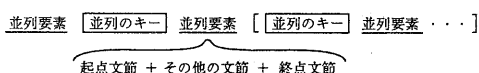


図 3: 並列構造の構成

と (4) が未完述語形式である。

2.2 並列構造の構成要素

文中に存在する並列構造は、例えば、図3のような構造を持つ。並列構造の存在を示す語を並列のキーと呼ぶ。並列のキーを挟んで並立する2つの文節列を並列要素と呼び、前側を前置要素、後側を後置要素と呼ぶ。また、並列要素の最初と最後の文節を起点文節、終点文節と呼ぶ。図3にあるように前置要素の終点文節は並列のキーの文節であり、後置要素の起点文節は並列のキーの文節の直後の文節となる。

図4は、名詞並列の構成要素の例である。図において、並列のキーは「および」、前置要素は「東部工業における作業組織の発展」、後置要素は「西部産業における職能組織の発展」である。また、並列要素の最初と最後の文節をそれぞれ起点文節、終点文節と呼ぶ。前置要素における起点文節は「東部工業における」、終点文節「発展および」であり、後置要素における起点文節は「西部産業における」、終点文節は「発展をとおして」である。

2.3 並列構造の曖昧性

自然言語処理技術において、並列構造を解析する際の大きな問題点は、並列要素の決定にある。この問題は、並列要素の候補が複数生じ、一意に並列構造を決定できないための曖昧性が

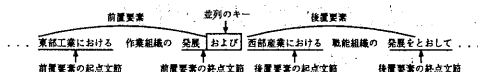


図 4: 並列構造の構成要素の例 (名詞並列)

ら生じる。その曖昧性が文全体の構造の決定(構文解析)に支障をもたらす。以下では、述語並列の並列要素を決定する際の曖昧性について述べる。

述語並列においても、名詞並列と同様の曖昧性が存在する。副詞などが並列した用言に係っている場合、それが前側の用言だけに係っているのか、それとも並列している用言の両方に係っているのか曖昧性が存在する。さらに、述語並列特有の問題として、連用中止法から生じる曖昧性がある。連用中止法とは、「勉強を終え、外で遊ぶ。」のように、連用形で文を中止し、さらに文を続けていく方法である。この連用中止法は、名詞並列と同様に自然言語処理解析を困難にする表現の1つである。なぜなら、連用中止法で結ばれる前側の要素と後側の要素には、様々な関係があるからである。それらの関係が並列要素の決定に影響を及ぼす。

連用中止法で結ばれる文には、時間の進行に従う文と、時間の進行に関わりなく並列する文がある。並列または進行という形で、連用中止法の前後の要素の接続関係があるため、前側の要素と後側の要素の関係は不安定になる。したがって、連用中止法で結ばれた述語並列を解析するには、前後の要素の関係を正しく把握することが必要になる。例えば、技術雑誌(「日経コンピュータ」誌などの解説文約22万語)を調査した結果、連用中止法による接続995箇所のうち、300箇所程度が述語並列であったという報告がある¹⁾。このように、連用中止法で結ばれるすべての文が述語並列であるとは限らない。また、前後の要素のどの関係までが述語並列に含まれるか決定することは難しい。1つの基準として、「前側の要素と後側の要素を入れ換えても、文の意味が変わらない」がある。しかし、この基準を満たす連用中止法を、計算機に認識させることは難しい。

3. 述語並列の並列要素の推定

3.1 推定の概要

読み手が並列構造の曖昧さを感じた場合、意味的な類似もしくは対比する単語に注目して並列要素を決定していると考えられる。また、構造的な類似性から並列要素を決定する場合もある。本研究に先だって、名詞並列、部分的並列の並列要素を推定する手法の構築を行った。その手法では、並列要素を決定する表層的な手がかりと構造的類似性で並列要素を推定している。そのため、述語並列の推定においても、構造的類似性に基づいて推定を行なうことを考える。

3.2 推定の手順

本研究の推定法では、単語の意味などを考慮していないので、述語並列の並列構造の推定においても、推定の対象となる並列構造が限定される。例えば、未完述語並列はサ変名詞が名詞または動詞のどちらの働きをするかを決定できなければ解析することはできない。そのため本研究では、完全述語並列のみを対象として並列構造の推定を行う。

述語並列の並列構造を推定する手順は、前処理と本処理の2つに分けることができる。前処理は、名詞並列、部分的並列の並列要素を推定する際の前処理と同様^{[2][3]}で、3つの過程からなる。まず、主に字種情報を利用して、文を仮の文節に分割する。次に、仮の文節に文節の性質を付与し、それを用いて仮の文節の分割誤りを修正する。最後に、文節の性質を利用して文節間の係り受け関係を決定する。

前処理で求めた文節の性質と文節間の係り受け関係を用いて本処理を行う。この処理は並列のキーを含んでいる文に対してのみ行う。本処理は以下の3つの過程からなる。

- (1) 文章中の並列のキーを検索し、述語並列のキーに対して、各並列のキーの前後の文節列に対して並列構造になりうる最長の文節列を求める。
- (2) 述語並列の前置要素と後置要素となりうる候補を得て、並列要素を推定する。基本的な並列要素を決定する方法は、並列要素を

表 1: 述語並列のキー

述語並列のキー
[連用中止法]、(の)に対し(て)、が、だけで(は)なく、けれど(も)、および、または、ならびに、あるいは、かつ、もしくは、及び、又は、並びに、或は

決定する表層的な手がかりと、構造的類似性である。

- (3) 最長の文節列内に複数の並列のキーを含む場合の並列構造の推定を行なう。

3.3 並列要素の推定の処理

3.3.1 並列のキーの抽出

本研究では、文章中のすべての文に対して前処理を行ない、並列のキーを含む文に対してのみ並列構造の推定を行なう。並列のキーの抽出誤りは、そのまま並列構造の推定誤りとなる。また、推敲支援を目的として並列構造の推定を行なっているので、並列のキーをできるだけ取りこぼしなく、かつ精度良く抽出することが重要である。

日本語の表現は多様であり、日本語における並列構造の存在を示す表現は多様である。そのため、これらの表現をすべて網羅することは難しい。本研究では、述語並列のキーとして、表1の表現を取りあげた。これら以外の表現が存在しても、それらは出現することが稀であると考え、抽出の対象から外した。

また、連用中止法には、連用形活用語の後に読点が来るものと来ないものがある。本手法では、読点を伴わない連用中止法は「連用修飾要素/用言」と判断してしまう。これを区別するためには、意味まで考慮しなければならない。一方、機械可読の日本語文章約67万文字に対して連用中止法の調査を行なうと、読点が付くものが2,230個、読点が付かないものが115個(4.9%)であった^[4]。また、読点を伴わない連用中止法のほとんどが次の述語に結び付く^[5]。この2点により、本研究では、読点を伴わない連用中止法は並列のキーとはしない。

また、名詞並列・部分的並列のキーにもなりうるキーについては、述語並列のキーであることを区別しなければならない。例えば「かつ」

は、すべての種類の並列のキーとなりうる。述語並列の場合、これら並列のキーの直前の表現が用言になる。したがって、並列のキーが接続している品詞が用言であれば、それは述語並列のキーとして処理を進める。

3.3.2 述語並列の最長範囲の決定

並列構造の並列構造を推定する際、並列のキーの前後で、やみくもに多くの並列要素の候補を求めても、処理の無駄であり意味がない。そこで、並列構造の要素となりうる最長の文節列を求め、候補を絞り込むことが重要になる。この最長の文節列の範囲のことを並列構造の最長範囲と呼ぶ。この最長範囲は、あくまでも最長の候補ということであり、実際の並列要素というわけではない。前置要素側の最長範囲を前方最長範囲と呼び、後置要素側の最長範囲を後方最長範囲と呼ぶ。

述語並列の最長範囲は、節となりうる最長の文節列を決定する。ところが、節となりうる最長の文節列は、文頭から文末までのすべての文節が当てはまってしまう。そこで、文節の中に、意味的に区切りの性質が強い文節(区切り文節と呼ぶ)があることを利用する。区切り文節には、条件を示す述語(「～すれば」など)、因果関係を示す述語(「～するので」など)、文末の述語を含む文節がある。区切り文節が存在する場合は、1つの文として意味的に完結することが多い。そこで、区切り文節を利用して述語並列の最長範囲を制限する。

前方最長範囲: 並列のキー文節から文頭に遡って検査する。検査対象文節が「区切り文節」または「接続詞」であれば、その後の文節までを前方最長範囲とする。それらがなければ、文頭の文節までを前方最長範囲とする。

後方最長範囲: 並列のキー文節から文末に向かって検査する。検査対象文節が「区切り文節」であれば、その文節までを後方最長範囲とする。または「接続詞」であれば、その文節の前までを後方最長範囲とする。それらがなければ、文末の文節までを後方最長範囲とする。また、検査対象文節が述語並列のキーである場合は、検査対象文節を

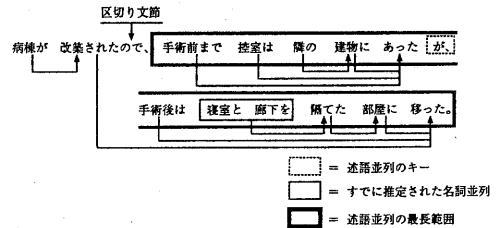


図 5: 述語並列の最長範囲の例

次に移し、複数の述語並列が含まれることを考慮して推定する。

図5は、述語並列の最長範囲を推定する例である。述語並列のキーは「が」である。前方最長範囲は、条件を示す区切り文節「～ので」が存在するので、「手術前まで～あったが、」になる。後方最長範囲は、区切り文節が存在しないので、「手術後は～移った。」になる。

3.3.3 表層的な手がかり

並列構造を含む文には、読み手が並列構造を容易に決定できる表層的な表現がある場合がある。この手がかりを表層的な手がかりと呼ぶ。述語並列の並列要素の推定においては、以下の2種類の手がかりを用いる。

起点文節の類似性: 前置要素の起点文節を推定する場合に用いる。起点文節の類似性とは、後置要素の起点文節と類似している前置要素の起点文節が存在することである。この起点文節の類似性として、「同じ文字または同じ文字列を含む」というものがある。そこで、後置要素の起点文節と同じ「受けの資格」を持つ文節で、平仮名を除く2文字以上の文字列が一致する文節があった場合、その文節を前置要素の起点文節とする。

後置表現: 後置要素の終点文節を推定する場合に用いる。後置表現とは、いくつか列挙した事柄を総括する、または、それにより他にも同様の事柄が存在することを暗示する表現である。一般的には並列部分の直後に置かれるため有効な手がかりとなりうる。後置表現には「など、などといった、といった」などがある。そこで、「受けの資格」が

用言であるものについて後置表現があるか調べ、そのような文節があった場合、その文節を後置要素の終点文節とする。

3.3.4 構造的類似性

構造的類似性に基づいて並列構造を推定する手順では、まず、述語並列の前置要素と後置要素を決定するために、比較対象候補を作成する。述語並列の最長範囲はすでに得ているので、その範囲内で比較対象候補を獲得する。また、述語並列の推定を扱っているので、比較対象候補は節となりうる文節列である。

述語並列は一般に長い文節列を並列要素として持つため、格要素を修飾する連体修飾成分が並列要素に係ることは稀である。また、述語並列を含む文は長くなるため、多くの並列要素の候補が存在する。このことから、構造的類似性に基づいて並列要素を決定する場合は、終点文節に係る格要素を比較単位とすることにする。比較対象候補の作成は、以下のとおりである。

- 前置要素の比較対象候補は、述語並列の前方最長範囲までの、終点文節に係る格要素を順に付け加えた文節列である。
- 後置要素の比較対象候補は、受けの資格が用言である文節を終点文節として持つ文節列である。ただし、終点文節の受けの資格が用言であっても、その文節を飛び越えて他の文節に係る文節が存在する場合は、後置要素の比較対象候補としない。

前置要素、後置要素の比較対象候補を作成する際に、終点文節に係る格要素に係る連体修飾節は、1つの文節列にまとめる。また、まとめた文節の「文節の性質」は格要素の「文節の性質」を引き継ぐものとする。

3.3.5 述語並列の並列要素の決定

以下の手順で述語並列の並列要素を決定する。

処理 1: 最長範囲内に、述語並列の並列要素を決定する表層的な手がかりがあれば、並列要素とする。

図 5 の文では、後置要素の起点文節「手術後は」と 2 文字同じ文字列を含む文節「手術前ま

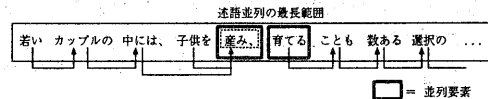


図 6: 述語並列の推定例

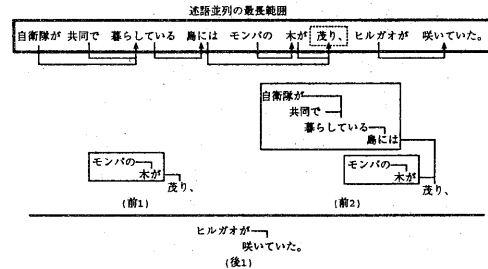


図 7: 比較対象候補の作成例

で」が存在するので、起点文節の類似性により、前置要素は「手術前まで～あったが」になる。

処理 2: 並列のキーの直前・直後の文節が、同じ品詞の用言であれば、直前の文節を前置要素、直後の文節を後置要素にする。

図 6 の文では、述語並列のキーの直前・直後の品詞が共に動詞であるので、前置要素は「産み」、後置要素は「育てる」になる。

処理 3: 最長範囲内で、述語並列の比較対象候補を作成し、構造的類似性に基づいて並列要素を決定する。構造的に完全に一致する候補が見つかった場合、その候補を採用する。完全に一致する候補が見つからなかった場合は部分的に一致した数が最大のものを採用する。

まず、述語並列の前置要素と後置要素を決定するために、比較対象候補を作成する。図 5 の文では、後置要素の最長範囲内には用言が 2 つある。しかし、文節「手術後は」が文節「移った」に係っているので、後置要素の候補は「手術後は～移った」の 1 つだけである。

図 7 の文に対する比較対象候補の作成について説明する。述語並列のキー文節は「茂り、」である。前置要素の比較対象候補の終点文節は述語並列のキーであるので、その文節に係る文節「島には」と「木が」に係る文節は 1 つの文節

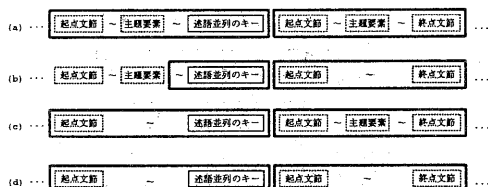


図 8: 処理 4 における並列要素の決定方法

列にまとめる。よって、前置要素の比較対象候補は、(前 1) と (前 2) の 2 つである。後置要素の比較対象候補は 1 つしかない。前置要素と後置要素の比較対象候補を比較すると、(前 1) と (後 1) の構造が完全に一致するので、前置要素は「モンパの木が茂り」、後置要素は「ヒルガオが咲いていた」である。

処理 4: 並列要素を決定する表層的な手がかりもなく、構造的な一致でも並列要素を決定できなかった場合、その時点での最長範囲を利用して並列要素を決定する。このとき、前置要素と後置要素の最長範囲内の主題要素の有無で、以下の規則を適用し、並列要素を決定する。

- 前置要素と後置要素が共に「主題要素」を含む場合：前置要素と後置要素共に最長範囲を並列要素とする。(図 8-(a) 参照)
- 前置要素は「主題要素」を含むが、後置要素は「主題要素」を含まない場合：主題要素の後の文節から述語並列のキー文節までを前置要素とする。また後方最長範囲を後置要素とする。(図 8-(b) 参照)
- 前置要素は「主題要素」を含まないが、後置要素は「主題要素」を含む場合：前置要素と後置要素共に最長範囲を並列要素とする。(図 8-(c) 参照)
- 前置要素と後置要素が共に「主題要素」を含まない場合：前置要素と後置要素共に最長範囲を並列要素とする。(図 8-(d) 参照)

4. 評価

本研究における、述語並列の並列要素の推定方法を計算機上で実装し、文章に適用して推定

表 2: 並列のキーの抽出結果

	キー	キーでない	合計
処理によって抽出	289	8	297
処理で抽出できない	5	—	—
合計	294	—	—

適合率 97% 再現率 98%

表 3: 並列のキーを含む文

並列キーを含む文の数				
1 個含む	2 個含む	3 個含む	4 個以上	合計
233	27	2	1	263

を行なった。また、並列のキーを抽出した結果も評価する。並列のキーの抽出結果については、前処理の段階での結果を元に、人手で判断した。使用した文章は、九州大学内で行なわれている情報工学演習 (第一・第二) の抄録である。この抄録のうち、85 件、約 63,500 字、1,228 文に対して本手法による推定を行なった。

4.1 並列のキーの抽出

表 2 に述語並列のキーの抽出結果を示す。表に示した値から、適合率は $\frac{289}{297} = 97\%$ であり、再現率は $\frac{289}{294} = 98\%$ である。適合率、再現率ともに高く、十分な精度である。抽出できなかった誤りは、サ変動詞が語幹だけで連用中止法として使用してあるものが、前処理の段階でサ変名詞と判断されていたものであった。これは、未完述語並列に分類され、本稿では抽出対象としていなかったものである。並列のキーでないにも拘らず誤って抽出してしまったものは、前処理の段階で、文節分割でミスがあった部分で生じたものであった。

表 3 に述語並列のキーを含む文の数を示す。この表は、処理により抽出されたキーを含む文について示したものである。抽出された並列のキーは 297 個であるが、1 文中に複数含まれることもあるため、文の数としては 263 文と少なくなる。この表より、1 文に述語並列のキーが複数含まれることは少ないことが分る。

4.2 述語並列の並列要素の推定結果

表 4 に述語並列の並列要素の推定結果を示す。構造の類似性により並列構造を推定しているこ

表 4: 並列要素の推定結果

	2 個	3 個	4 個以上	完全	他	合計
「○」	28	19	2	0	26	75
「△」	50	17	20	0	55	142
「×」	2	1	5	0	38	46
合計	80	37	27	0	119	263

とから、構造が一致した個数と推定の関係を示す。表中の「完全」は構造の完全一致で、「他」は構造以外で並列要素を推定した場合を示している。「○」は本手法でうまく推定ができているもの、「△」は推定された構造でも良いもの、「×」は推定に失敗しているものを示している。

今回調査した文章中では、構造が完全一致したものはなかった。また、格要素に係るものをまとめてから構造を比較するという手法により、部分的に一致した個数は、せいぜい数個である。しかし、2～3 個と一致した場合には、推定した結果は正解、もしくは、そう読むことができると思われるものとなっている。また、文が長くなった場合には、一致する数がたまたま増えてしまうことがあり、この場合は多く一致していても逆に推定に失敗することがある。

本研究では、推敲支援が目的であるから、読み手が読み取る可能性のある候補として提示することができるという点で、「推定された構造でも良いもの」も正しく推定できているとしてよい。このことから、推敲支援を目的とした精度は $\frac{75+142}{263} = 83\%$ となる。

また、述語並列の並列要素の推定結果と、推定した文の述語並列の分りやすさについて調査した。その結果を表 5 に示す。表中の「○」「△」「×」については表 4 と同じである。また、「●」は簡単に読み取れるもの、「▲」は良く読めば分るもの、「？」は分りにくいものを示している。この読み易さは、書き手と同じ並列要素とは限らなくても、意味まで考えると、すんなりと並列要素が把握できる文を読み易いとした。

5. おわりに

述語並列の並列要素を決定するために、構造的類似性、表層的な手がかり、規則を利用する方法について述べた。この推定方法を文章に対

表 5: 並列要素の推定結果 2

	「●」	「▲」	「？」	合計
「○」	73	2	0	75
「△」	74	64	4	142
「×」	10	17	19	46
合計	157	83	23	263

して使用し、評価した結果、推敲支援を目的とした精度は 83% 程度となった。

推定結果より、構造が一致しない場合が $\frac{119}{263} = 45\%$ と多い。しかし、この結果のなかには、「～に行く」と「～へ行く」といったように、本来一致しても良いものが、処理上は一致しなかったものとして含まれている。このようなものへ対応を取り入れることができれば、より精度があがると思われる。

また、本稿では、文章中にある並列構造はすべて書き手に指摘している。しかし、並列要素がはっきり分かる並列構造も指摘してしまう(過剰指摘)。過剰指摘が多いと書き手は煩わしく感じる。そのため、指摘候補を絞り込む手法を構築することが今後の課題である。

参考文献

- [1] 首藤, 吉村, 津田. “日本語技術文における並列構造”. 情報処理学会論文誌, Vol. 27, No. 2, pp. 183–190, 1986.
- [2] 菅沼, 山村, 牛島. “日本語文における名詞句の並列構造の推定およびその推敲支援への適用”. 情報処理学会論文誌, Vol. 38, No. 7, pp. 1296–1307, 1997.
- [3] 菅沼, 笠原晋, 牛島. “字面解析による用言の活用形推定について”. 情報処理学会論文誌, Vol. 37, No. 6, pp. 1007–1016, 1996.
- [4] 縄田, 菅沼, 牛島. “日本語文章推敲支援ツール『推敲』における連用中止法の抽出について”. 第 47 回情報処理学会全国大会, 第 3 巻, pp. 229–230, 1995.
- [5] 黒橋, 長尾. “長い日本語文における並列構造の推定”. 情報処理学会論文誌, Vol. 33, No. 8, pp. 1022–1031, 1992.