係り受けの複雑さの指標に基づく文の書き換え候補の生成と推敲支援への応用

横林，博
九州大学システム情報科学研究院知能システム学部門

菅沼，明
九州大学システム情報科学研究院知能システム学部門

谷口，倫一郎
九州大学システム情報科学研究院知能システム学部門

出版情報：火の国情報シンポジウム，pp.200-207，2003-03
バージョン：accepted
権利関係：

係り受けの複雑さの指標に基づく文の書き換え候補の生成と推敲支援への応用

横林　博，菅沼 明，谷口 倫一郎
九州大学大学院システム情報科学府
〒816–8580 福岡県春日市春日公園6–1
TEL: 092–583–7618
E-mail: {yokoba,suga,rin}@limu.is.kyushu-u.ac.jp

あらまし　文章の推敲支援を行うツールは多数存在するが、その多くは推敲支援に役立つ情報を提示するのみである。それらの情報を用いれば推敲作業自体は楽になるが、文章をどのように書き換えるかは書き手自身が考える必要がある。我々の研究室では、本稿に先立って係り受け解析過程モデルを作成した。このモデルは係り受けの複雑さの指標を元に文の複雑さを判定し、係り受けの複雑な文を抽出することができる。しかしどのように書き換えれば文の複雑さが減るかについては何も提示しない。そこで、この指標が下がるように、構文情報のみを用いて文の書き換え候補を作成することを考えた。複雑さの指標が下がれば、その文は書き換える前よりも複雑さが減っていると言える。さらに、この書き換え候補による推敲支援方法について述べる。
キーワード　係り受け解析、書き換え候補の生成、推敲支援、文書処理

Generating Candidates for Rewriting Based on an Indicator of Complex Dependency and It's Application to a Writing Tool

Hiroshi Yokobayashi, Akira Suganuma, Rin'ichiro Taniguchi
Department of Intelligent Systems, Kyushu University
6–1, Kasuga-kouen, Kasuga, Fukuoka, 816–8580 Japan
TEL：092–583–7618
E-mail: {yokoba,suga,rin}@limu.is.kyushu-u.ac.jp

Abstract　Many writing tools for Japanese documents only present a point of improving a sentence to a writer. He has to consider by himself how to rewrite a text. Our laboratory has proposed the model which imitates human dependency analysis for a Japanese sentence. The model can extract a complex sentence based on the indicator of complex dependency. It presents, however, nothing about the method of decreasing the complexity of the sentence. In this paper, we describe a generating method of candidates for rewriting based on the indecator. We apply,furthermore, the method to a writing tool.
Key words　Dependency structure analysis, Generating candidates for rewriting, Writing tool, Text processing

1. はじめに

近年、個人用PCが急速に普及し、個人が簡単にPCを扱えるようになった。これらのPCは、ワープロソフトを標準搭載し、その結果個人で容易にワープロソフトを扱えるようになった。

そして文章作成支援機能の一つとして、文章校正支援機能がワープロソフトに含まれるようになった。校正支援機能を用いると、校正作業に役に立つ情報が提示され、書き手はその情報を用いて効率よく校正作業を行うことができる。校正作業の次に書き手が行う作業としては推敲作業がある。推敲作業では、文法的には間違ってはいないが読みづらい文や曖昧な表現を含む文を発見し、書き手の意図が読み手に伝わりやすい文に書き直す作業を行う。ただし推敲作業に役立つ情報を提示するようなツールはまだ少ないようである。

本研究室では本研究に先立って、係り受け解析過程モデルを作成した[1]。これは人間の係り受け解析過程を模したものである。このモデルによって、文の係り受けの複雑さの指標を得ることができる。

係り受けの複雑さの指標を用いることによって、読みづらい文を抽出し、推敲を要領よく行うことが可能になる。しかし、係り受け解析過程モデルは構文情報を元に複雑な文を抽出するのみで、具体的な推敲作業は書き手の判断に委ねられている。また文を書き直した後に、係り受けの複雑さの指標をすぐに再計算できる仕組みがないため、訂正後の文が訂正前に比べて読みやすくなっているかどうかは分かりづらい。

本稿では推敲支援に役立つ情報として、文の係り受けに関する情報に着目し、これを用いて情報を提示する方法について述べている。係り受けの複雑さの指標による推敲支援だけでなく、文の具体的な書き換え候補の作成を考える。具体的には、指標が低くなるように、構文情報を用いて修飾節の入れ替えや長い文の分割を行って推敲支援へ応用する方法を提案する。さらに、係り受け解析過程モデルを利用した推敲支援システムも構築した。係り受けの複雑さの指標を元に文を書き直した後、指標を再計算して推敲前の文の指標と比べることも可能である。

2. 係り受け解析過程モデル

2.1 人間の係り受け解析過程

一般的に、人間が文を理解するときには、形態素解析、構文解析といった作業を行っていると考えられる。本論文では、この構文解析の文法として、係り受け文法を用いる。

上記の作業を人間が行うためには、ある種の記憶機構が存在している必要がある。この記憶領域を用いたモデルとして、村田らが文献[2]で用いたモデルがある。このモデルでは、文を理解するときに短期記憶に格納する必要があるものは、係り先が未決定な文節であると考えている。またその短期記憶の容量については7 ± 2程度のチャンクであると提唱されている。そこで、文理解の過程において使用される短期記憶の容量も、7 ± 2程度であると仮定することができる。

これらのことから、人間の係り受け解析過程を模したモデルを作成して文を処理したときに、短期記憶の容量を超えるような文は、モデルの解析能力を超えた文であるということが言える。そして、そのような文はわかりにくい文である可能性が高いと考えられる。

我々の研究室では、このモデルを計算機上に構築し、7 ± 2を超える文を提示するシステムを作成した。次節ではこの係り受け解析モデルの概要について説明する。

2.2 係り受け解析過程モデルの概要

係り受け解析過程モデルは、大きく分けて、入力処理部、係り受け判断部、短期記憶スタックの三つの部分から成る。

2.2.1 入力処理部

入力処理部は、入力文に形態素情報を与え、文を文節に区切る。そして、文の先頭の文節から順に係り受け判断部に文節を渡す。

2.2.2 係り受け判断部

係り受け判断部は、後述する短期記憶スタックから取り出した文節と、入力文節との間の係り受けの有無を判断する。係り受け判断部では文の係り受け情報を用いて、文の係り受け解析を行い、文を読む動作を擬似的に再現する。

2.2.3 短期記憶スタック

係り受け解析過程モデルでは、短期記憶スタックと呼んでいるスタック型の短期記憶を持っている。短期記憶スタックの記憶単位をブロックと呼び、データの出し入れはブロック単位で行う。短期記憶スタックへ1ブロック保存する動作を push、短期記憶スタックから1ブロック取り出す動作を pop と呼ぶ。

文を処理していく過程で、係り先が未決定な文節は、スタックに push する。その後スタック中の文節と係り受け関係が成立する文節が表れると、その二つの文節を結合する。この作業を文の最後まで繰り返し、使用したブロックの最大段数を係り受けの複雑さの指標として使用する。

2.3 動作アルゴリズム

係り受け解析過程モデルでは、文が入力されると、文頭から係り受け解析を行っていく。その時のスタック操作の基本動作アルゴリズムを以下に示す。

操作1

まずは入力処理を行う。入力処理では、入力された文を文節ごとに区切り、全ての文節を未処理文節列に入れる。その後操作2に進む。

操作2

未処理文節列の最初から1つ文節を取り出し、係り受け判断部に送る。取り出した文節は未処理文節列からは取り除く。操作1の後、文頭の文節から係り受け判断部に送る。この文節を入力ブロックと呼ぶ。この後操作3に進む。

操作3

短期記憶スタックから1ブロック pop する。これを poppedブロックと呼ぶ。短期記憶スタックが空で、入力ブロックが文末ブロックであったら、処理を終了する。そうでない場合は、操作4に進む。

操作4

入力ブロックと poppedブロックとの間で係り受け関係があるかの判断を行う。係り受けが成立した場合、それらを結合し、それを入力ブロックとして操作3に戻る。係り受けが成立しない場合は、poppedブロックと入力ブロックをそれぞれスタックに push し、操作2に戻る。

前述のとおり、人間の短期記憶容量は7±2程度であると言われている。つまり、文を読む際にこの数よりも多くのことを覚えなければならないときに、人間は文の理解に困難を感じると考えられる。係り受け解析過程モデルにおいても、短期記憶スタックの使用容量が7±2を越えるような必要があるとき、モデルの処理能力を越えた文であると考えることができる。そこで、文を処理するときに使用した短期記憶スタックの段数の最大値を係り受けの複雑さの指標とする。

3. 文の書き換え候補の作成

係り受け解析過程モデルを使用すれば、係り受けの複雑な文を抽出することが可能である。ただし、モデルはどの文が複雑であるかを指摘するのみで、その文をどのように書き換えれば読みやすくなるかについてはまったく関与しない。よって指摘された文を読みやすくするには、文のどの部分が複雑になっていて、どのように書き換えれば文が読みやすくなるかを書き手が考える必要がある。

そこで書き手の推敲作業の負担を少しでも減らせるように、モデルよって抽出された文を対象に、推敲作業に役立つ情報として書き換え候補文を作成することを考える。

3.1 修飾節の入れ替え

日本語は、修飾句・修飾節前置型の言語である。係り受け解析過程モデルではこの修飾節に着目し、修飾節の係り受けが複雑な文を抽出することを目的としていた。文の修飾節が複雑になればなるほど、その文は読みづらくなると考えられる。

修飾節・修飾句にはいくつかの修飾語が含まれており、それらの修飾語ごとに、修飾語の長さや修飾語が含んでいる文節の数が異なることが多い。故にこれらの修飾語の並びによって文の読みやすさが変化することが予想される。文献[3]には、修飾語の順序に関して、長い修飾語は先に、短い修飾語は後にした方が、その文が読みやすくなると記述されている。そこでこのルールを利用して、修飾節の語順を入れ替えることによって文を読みやすくすることを考える。

3.1.1 係り受けの複雑さの指標と文の読みやすさの関係

文の読みやすさの目安としては、係り受けの複雑さの指標がある。そこで修飾語の順序と係

り受けの複雑さの指標を調べることにする。修飾語として長い文節が後ろに来ている簡単な例を例文1に示す。括弧内は係り受けの複雑さの指標である。

例文1) 上空に花子の風船が針の先のような小さな点として現れた。(4)

これを修飾語の長い順に入れ替えると例文2になる。

例文2) 針の先のような小さな点として花子の風船が上空に現れた。(3)

例文1では「上空に」「花子の風船が」「針の先のような小さな点として」が修飾語であり、係り受けの複雑さの指標は4である。この3つの修飾語の中で「針の先のような小さな点として」が一番文節数が多い。そこで例文2のように、この文節が一番最初に来るように文節を並び替えると、結果的に係り受けの複雑さの指標が減ることが分かる。

3.1.2 修飾節の切り出し

長い修飾節を前に、短い修飾節を後にすると係り受けの複雑さの指標が低くなることが分かった。そこでこのルールを利用して、指標が低くなるように文を書き換えることを考える。そのためにはまず修飾節単位で文を分ける必要がある。

そこで、各文節を連文節としてまとめていく作業を行う。連文節とは、二つ以上の連続した文節が一まとまりとなって、一つの文節と同様の働きをするものである。文頭から文節を検査していき、以下の2つの条件を元に連文節を判定していく。連続する文節間で係り受け関係が成立せず、かつ下の条件を満たさない場合は、そこで連文節が一旦切れるものとする。連文節の判定には各文節の係り受け情報と係りの資格[4]を用いる。

条件1: 検査対象文節が連体修飾要素の修飾語に係っているときは、その修飾語と連文節になる

条件2: 検査対象文節に連体修飾要素の修飾語が存在するときは、その修飾語と連文節になる

例えば図1では、「上空に」は連体修飾要素ではなく、次の文節である「花子の」を修飾しない

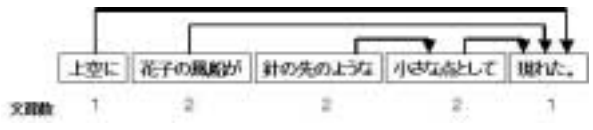

図 1: 修飾節の切り出し

ため、「上空に」は連文節にはならず単独で修飾語になる。次の文節「花子の」は係りの資格が連体修飾要素で、かつ次の文節に係るため、「花子の」と「風船が」が連文節になる。同様に「針の」と「先のような」は連文節になるが、「先のような」は「点として」に係るためにここで区切られる。「小さな」と「点として」は1つの連文節になる。図1は最終的に5つの修飾句に分けられることになる。

3.1.3 指標に基づく修飾節の入れ替え

前節で作成した修飾句を並びかえて、係り受けの複雑さの指標が低くなるように文を作成する。修飾句の並び替えは以下の手順で行う。

手順1: 文頭から順に修飾句を検査し、近くに係る修飾句から順に結合していく。複数の修飾句が同じ文節に係る場合は、文節数の多い順に並び替えて、一つの修飾句にまとめる。

手順2: 手順1を文末の修飾句まで繰り返す。

図1を文頭から検査していくと、「針の先のような」が「小さな点として」に係っていることが分かる。「小さな点として」に係っている修飾句は「針の先のような」だけなので、この2つの修飾句を1つの修飾句にまとめる。その後「現れた。」に前の3つの修飾句が係っていることが分かる。そこでこの3つの修飾句を、文節数が多い順に並び替える。この場合は「針の先のような小さな点として」が4文節で一番文節数が多いので、この修飾句が最初に来ることになる。この手順で並び替えると最終的には例文2になる。

修飾句の並び替えに関しては、係助詞「は」を含む修飾句（以下主題要素と呼ぶ）は並び替えないことにする。その理由としては、日本語は「主題要素＋修飾節＋述部」という語順になりやすいからである。その根拠として、京都大学テキストコーパス[5]の15513文を用いて主題要素が修飾句の先頭に来るかどうかを調査した。その結果を表1に示す。この結果から、主題要素

表 1: 主題要素が先頭に来る文の数

主題要素の位置	文数
文の先頭	3514
接続要素 or 副詞の次	333
最初の修飾句と連文節になる	1857
合計	5704
主題要素を含む文	7998
主題要素が先頭に来る割合 (%)	71.3

は約7割の確率で修飾句の先頭に来ていることが分かる。

また接続詞、接続助詞を含む修飾句（以下接続要素と呼ぶ）も、主題要素を含む修飾句同様並び替えないことにする。接続要素を含む修飾句の前後の修飾句を入れ替えると、生成された文の意味が変化している可能性が高いからである。

3.2 長い文の分割

新聞記事や学術論文を読んでいると、時折やたらと長い文を発見することがある。長い文の種類としては、文中に節構造がいくつも含まれているものや、修飾節がだらだらと長いものなどが存在するが、無駄に長い文というのは読み手にとって苦痛である。そこで推敲支援として、長い文を機械的に2つに分けることを考える。

例文3は125文字、27文節で、京都大学テキストコーパスの中でも長い方の文である。文の長さに関しては感覚的な判断は可能だが、具体的な数値に基づく判断というのは案外難しい。

例文3) それぞれに大きな政治的変化であり、戦後政治に強烈なインパクトをもたらしたが、いまこの一年半を振り返ってみると、政治的大変化はあってもそれが在来の政治構造の深奥部を突き崩す政治の大変革に至っていないところに、国民のもどかしさと失望感の増幅がうかがえる。（10）

3.2.1 処理対象とする文の条件

本研究では文の読みづらさの目安として、係り受けの複雑さの指標を使用している。そこでまずは文の長さと係り受けの複雑さの指標との関係を調べた。調べるに当たって、前述のテキストコーパス15513文を調査の対象とした。その結果を図2に示す。

図2は平均文節数と係り受けの複雑さの指標との関係を示している。グラフから指標が高くなるほど文節数が増えているのが分かる。また指標7以上の文の平均文節数を表2に示す。

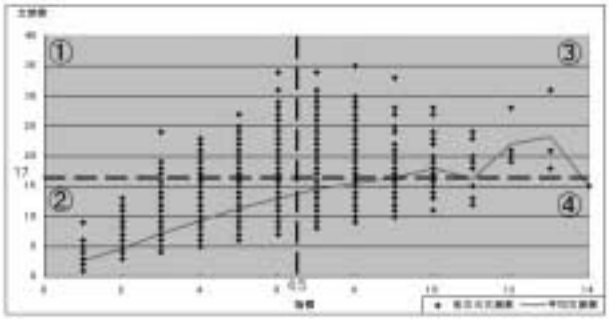

図 2: 文の平均文節数と係り受けの複雑さの指標の関係

表 2: 指標7以上の文の平均文節数

	1文当たりの平均文節数
文全体	13.5
指標7以上	17.4
指標9以上	18.5

表2を見ると、指標7以上の文の平均文節数が17.4になっているのが分かる。そこで指標が7、文節数が17という条件でグラフを区切ると図2のようになる。図2の①の領域に入る文は長い文であるが指標は高くない、つまり文の長さが読みづらさにそれほど影響を与えていない文の集まりである。このような文は名詞句の羅列等によって文が長くなっている場合が多く、読みづらさは感じないと思われるため書き換えの対象にはしない。④の領域に入る文は指標は高いが短いという文の集まりである。これらは、文の長さとは異なる要因で係り受けの複雑さの指標が高くなっていると考えられるので、書き換えの対象からは外す。そこで、今回は③の領域の文を処理の対象とする。

3.2.2 区切る場所の設定

以上の条件を元にして文を区切ることにする。ただし長い文を闇雲に切ればよいというわけではない。長い文には主に以下の2つの種類が存在すると考えられる。

- 複数の節構造が接続詞等によって接続されて、長くなっている文
- 連続した修飾節がだらだらと続いている文

この2つは文法的にまったく異なるもので、この2つの原因を切り離さずに文の分割を考える

ことは不可能である。故に、それぞれの場合に対して文を区切ることを考えることにする。

3.3 接続要素による長い文の分割

文章を書いているときに、何か複数のことを一つの文のみで丁寧に説明しようとして、結果的にその文が長くなりすぎて読み手が読みづらいと感じることがある。このようなときは一文のみで説明しようとするのではなく、複数の文に分けて説明したほうが文のテンポが良くなり、読み手も理解しやすくなる。

ただし文を区切るときには、任意の箇所で切ればよいというわけではない。当然ながら文の意味的な区切りを考慮する必要がある。そこで、複数の節を持つ文に対しては、

- 接続要素 ＋ 読点

を満たす文節で区切ることにする。例えば例文4では「したいものだが」で区切ることになる。

例文4) ことしこそポスト冷戦、バブル崩壊を踏まえ、二十一世紀を見据えた新しい日本の在り方に展望を切り開く年にしたいものだが、肝心の政界の方は依然として行き着くあてのない漂流を続けている。

書き換え候補を作成するときは、本来ならば接続助詞の変わりに接続詞を使用するのが理想的である。ただし、一つの接続助詞が複数の意味を持つ場合が多く、的確な接続詞をあてがうのは難しい。そこで現時点ではシステムが分割できると判断した箇所で文を一旦切って、残りを別の文として書き手に提示する。どのように文をつなぐかは書き手の判断にゆだねることにする。

3.4 逆茂木型の文の書き換え候補の生成

前述のとおり、日本語は修飾句・修飾節前置型の言語である。この修飾句・修飾節が長い文、または修飾節の中の言葉にさらに修飾節が係るような文は読みづらくなる。このような文を文献[6]では逆茂木型の文と呼んでおり、書き手の意図が読み手に伝わりづらく、読みづらい文になりやすいと述べている。

逆茂木型の文を係り受け解析過程モデルで処理すると、出力される係り受けの複雑さの指標はそうでない文に比べて大きくなることが多い。枝である修飾節が多く存在するほど、スタックに積むブロックの数も多くなるからである。そこで枝である修飾節を抽出し、それを別の文に分けることによって、文を読みやすくすることを考える。

図 3: 逆茂木型の枝の修飾節の検出例

逆茂木型の文の修飾節の検出は、各文節の係りの資格を用いて行う。まず文頭から文節を検査していき、係りの資格が「連体修飾要素/用言」である文節を探索する。見つかった場合は、その文節から文の前方に検査していく。係り先が先ほど検出した文節を飛び越える文節が現れると、その一つ手前の文節までを修飾節の範囲とする。

例えば図3では、文頭から検索していくと「持ってきた」の係りの資格が「連体修飾要素/用言」であることが分かる。そこから前方に検索していくと、「彼は」が「持ってきた」を飛び越えて「読んだ。」に係っているのが分かるので、「太郎が」から「持ってきた」までを逆茂木型の枝の修飾節であると判断する。

書き換え候補を生成するときは、幹の部分を1つの文として生成し、抽出した修飾節を別の文とする。そのため、抽出した修飾節が複数存在するときは、生成される文も3つ以上存在することになる。なお実際に文を処理するときは、3.2.1節の条件を加味して行う。

4. 評価

修飾節の並び替えと長い文の分割に関する評価実験を行った。

評価実験の対象とした文は、九州大学工学部電気情報工学科2001年卒業論文（466文）とした。括弧記号を含む文は正しく処理できないため、前もって取り除いてある。

4.1 修飾節の入れ替えの評価

上記の466文の中で、処理の対象となったのは86文存在した。これらの文の係り受けの複雑さの指標を表3 に示す。また修飾節を入れ替え

表 3: 修飾節入れ替えの評価対象文の指標

係り受けの複雑さの指標	文の数	係り受けの複雑さの指標	文の数
10	3	6	16
9	2	5	19
8	11	4	12
7	20	3	3

表 4: 修飾節入れ替え後の指標の増減

入れ替え後の指標の増減	文の数	入れ替え後の指標の増減	文の数
−4	1	−1	28
−3	4	0	31
−2	17	1	5

た後の文の係り受けの複雑さの指標の増減と文の数の関係を表4に示す。評価した文の中から例を挙げる。

例文5) 並列構造は、図のように、機能面に関して文形式の並列構造と非文形式の並列構造に分けることができる。(8)

例文6) 並列構造は、文形式の並列構造と非文形式の並列構造に図のように、機能面に関して分けることができる。(4)

例文5は文自体は短いが、「文形式の...並列構造に」が述語の直前に存在するため、それまでのブロックが蓄積される分、指標が大きくなっている。そこでこの修飾句を他の修飾句よりも前に出すと、例文6のように指標が下がることになる。

上で挙げた例文では読点を含めた修飾句をそのまま入れ替えて文を作成している。これに関してはシステム側で修正することが困難なため、読点の打ち直しなど、細かい修正は書き手に任せることになる。また修飾節の入れ替えは係り受け情報を用いて行っている。この処理では、係り受け情報が正しいことを前提にしているため、誤った係り受け情報が出力されると生成された文の意味が異なっている可能性が高くなる。

4.2 長い文の分割の評価

4.2.1 接続要素による長い文の分割

466文の中で処理の対象となったのは11文である。これらの中で、分割した後の長い方の文の指標の増減を表5に示す。評価した文の中から例を挙げる。

表 5: 接続要素で分割した後の文の指標の増減

書き換え後の指標の増減	文の数	書き換え後の指標の増減	文の数
−4	1	−2	3
−3	1	−1	6

例文7) 並列構造解析において重要な点は、後置要素の終点文節まで読み進めて、そこでどの文節列とどの文節列がはじめて並列になっているかがわかることである。(8)

例文8) そこではじめてどの文節列とどの文節列が並列になっているかがわかることである。(5)

例文7は指標が8、文節数が17の文である。この文では、文中に接続要素「読み進めて」が存在するため、そこで区切ることになる。例文8は例文7を2つに分けた後の、長い方の文であるが、分ける前に比べて指標が3下がっている。

4.2.2 逆茂木型の文の分割

466文の中で処理の対象となったのは28文である。これらの文から逆茂木型の枝葉の修飾節を取り除いた文の指標の増減を表6に示す。評価した文の中から例を挙げる。

例文9) チェコスロバキアの全国的情報システムの作成を目標とした総合的長期計画の一環としてとりあげられている情報科学における近代的方法の適用について述べ，大きな国立情報機関で達成されつつある主要方向を示す。(11)

例文10) 情報科学における近代的方法の適用について述べ，大きな国立情報機関で達成されつつある主要方向を示す。(4)

例文11) チェコスロバキアの全国的情報システムの作成を目標とした総合的長期計画の一環としてとりあげられている。(7)

例文9は「情報科学に」に係る修飾節が長いため、文が読みづらくなっている。このような場合はその修飾節を分けて、別の文としたほうが読み手に意図が伝わりやすくなる。そこで「チェコスロバキアの」から「とりあげられている」までが一つの大きな修飾節となっているため、この修飾節を抜き出して例文11のように一つの文とする。そうすると文の幹の部分が例文10になり、書き換える前に比べて指標が大幅に下がっていることが分かる。

表 6: 逆茂木型の枝を除去した文の指標の増減

書き換え後の指標の増減	文の数	書き換え後の指標の増減	文の数
−7	1	−3	1
−6	2	−2	4
−5	2	−1	4
−4	4	0	10

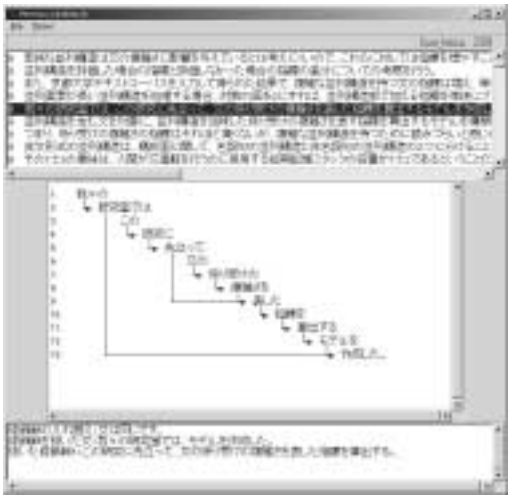

図 4: 推敲支援情報を提示するインターフェース

5. 推敲支援への応用

係り受け解析過程モデルでは、7±2を基準にして文の読みづらさを判断することができる。このモデルでは構文情報のみを用いて文の複雑さを判定するため、モデルが抽出したすべての文が読みづらいとは考えにくいが、提示した文の中から書き手に推敲を促すことは有意義であると考える。

そこで係り受け解析過程モデルをベースにして、本稿で提案した機能を用いて、推敲支援システムの構築を行った。システム構成は以下の通りである。

- 形態素解析:「茶筌」[7]
- 係り受け解析:「CaboCha」
- 読みにくさの判定:係り受け解析過程モデル
- 推敲支援情報の提示:自作インターフェース

係り受けの複雑さの指標を書き手に提示するインターフェースを図4に示す。インターフェースは3つのウィンドウから成る。上から順に、文とその係り受けの複雑さの指標の一覧、構文情報、メッセージを表示する欄である。一番上のウィンドウでは、入力したテキストファイルの各文の係り受けの複雑さの指標を算出した後、指標の高い文から順に提示する。一番上のウィンドウで文を一つ選択すると、下2つのウィンドウに瞬時に解析結果を提示する。

本稿では推敲支援方法として、修飾節の入れ替えと長い文の分割を提案した。これらを推敲支援へ応用するときは、1つの文に対して両方の方法を組み合わせて結果を1つだけ出力するやり方と、両方の方法を別々に割り当ててそれぞれの結果を出力するやり方が考えられる。現状では後者のやり方で推敲対象文を評価して、出力されたそれぞれの結果を提示する。推敲作業自体は書き手の判断にゆだねることにする。

6. おわりに

本稿では、構文情報を用いて文の書き換え候補を作成する方法を提案した。またそれらを推敲支援に応用する方法についても提案した。

今後の課題としては、読みづらい文の一つであると考えられる、曖昧な係り先を含む文の検出と指摘を考えている。また推敲支援情報を提示するインターフェースの改良を考えている。また本稿の推敲支援システムの客観的な評価を行うことも今後の課題である。

参考文献

[1] 小池康弘, 菅沼明, 谷口倫一郎, “係り受け解析過程モデルを用いた係り受けの複雑さの指標の作成とその推敲支援への応用”, 第14回情報処理学会九州支部研究会報告, pp.370–377 (2000).

[2] 村田真樹, 内元清貴, 馬青, 井佐原均, “日本語文と英語文における統語構造認識とマジカルナンバー7±2”, 自然言語処理, vol.6, No.7, pp.61–71 (1999).

[3] 本多勝一, “日本語の作文技術”, 朝日文庫, (1982).

[4] 菅沼明, 山村広臣, 牛島和夫, “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).

[5] 黒橋禎夫, 齋藤由衣子, 坂口昌子, “コーパス作成の作業基準”, 京都大学テキストコーパス. (1997).

[6] 木下是雄, “理科系の作文技術”, 中公新書, (1981).

[7] 松本裕治, 北内啓, 山下達雄, 平野善隆, “日本語形態素解析システム『茶筌』version2.0 使用説明書”, 奈良先端科学技術大学院大学松本研究室 (1999).