構文解析木を対象とするデータ解析法の研究
EDR コーパス文例を用いた助詞の分析

雄山 真弓　　岡田 孝
oyama@kgupyr.kwansei.ac.jp　okada@kgupyr.kwansei.ac.jp

関西学院大学　情報処理研究センター

1 はじめに

文のデータ解析においては、文を文字列、単語の並び、構文解析木、意味構造などの各レベルから多面的に解析することが必要である。しかしながらこれまでの研究では、文字列以外のデータを準備することが困難であったこと、構造を有するデータに対する適当な分析法がなかったこと、等の理由により、構文解析木や意味構造の解析はほとんど行われてこなかった。しかし、近年多数のコーパスが利用可能となってきており[1]、なかでも EDR コーパスは意味構造までを含めたすべてのレベルでの情報を備えている[2]。このようなコーパスを利用して、文のデータ解析を実行することにより、統計的なものも含めた新たな文法的知見が得られ言語学の発展に寄与することが期待できる。

我々は、既に文に含まれる特定の単語または構文木の中間節点を *viewpoint* として定めれば、構文木のトポロジー的な性質のみを使用して、拡張された ID3 法による文のデータ解析が可能であることを発表した[3]。またこの方法を、トポロジー的な性質ばかりでなく、各ノードに付随する通常の属性を含めた一般的な形で定義した[4]。本報告は、これまでと同様に構文解析木を対象とするが、より知識の探索範囲を広げた方法を考案し実際のシステムを作成した。

日本語文法において、「が」と「は」の問題は多くの注目を集めてきた[5]。そこで、新しい方法を EDR 日本語コーパス中の「が」と「は」を含む例文に適用したところ、「が」と「は」の使用法に関するいくつかの既知の文法的知見を再確認できたばかりでなく、統計的な知見をも得ることができた。

2 EDR 日本語コーパスデータ

本報告では言語データとして EDR 日本語コーパスを使用する。個々の文に付加されている情報の中では、構文木の情報と形態素の情報を利用する。構文木情報の例を図1に示す。形態素情報で注意する点は、「動詞」、「形容詞」等の品詞が語幹に対して割り当てられ、語尾には「語尾」という品詞が割り当てられている。また構文木の内部節点には、下位の各節点の合成関係を示す4種の関係子：修飾合成（M）、統合合成（S）、数合成（N）、複合語合成（I）のいずれかが割り当てられている。

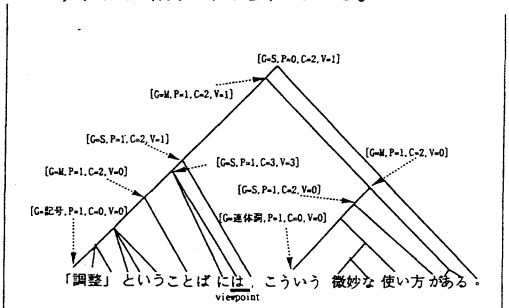

図1． EDR コーパスにおける構文木の例
G: 合成関係子、P: 親節点数、C: 子節点数、V: viewpoint への経路が通る隣接節点
0: 親節点, n: 左から n 番目の子節点

本報告では、コーパス中で助詞「が」または「は」を含み、かつ複合語合成を有しない最初の 1000 文を選んで使用した。これらの助詞に対応する葉節点が *viewpoint* となり、その内容の「が」または「は」が識別されるべきクラスラベルとなる。1文中に複数の「が」、「は」が出現する場合には、同一の文を *viewpoint* の異なる複数の事例として取り扱った。その結果解析対象としてはちょうど 1200 事例を対象とすることとなった。

3 解析法

事例として与えられる各構文木には *viewpoint* となる葉節点が存在する。本報告で提案する解析法では、*viewpoint* の周辺にあるいくつかの連結したノードの集合を *field-of-view* と呼び、*field-of-view* 内の各節点の属性を利用して実際の知識候補の探索を行う。そこで問題は、(1) 如何にして *field-of-view* を生成し選択するか、(2) 与えられた *field-of-view* からどのような方法で知識候補の探索を行うか、(3) 得られた知識候補からいかに実際に有用な知識を得るか、の3点に絞られる。以下この順に方法の概要を述べる。

3—1 *field-of-view* の生成と選択

この過程の基本を、図2の *field-of-view*: F を例にとって解説する。この F を部分木として含む事例集合 S が解析対象となったとき、まず ID3 法により知識候補を探索し、対応する事例の集合 $S_{knowledge}$ を確定する。事例集合 S 中で $S_{knowledge}$ に含まれる事例が少数であれば、残りの事例中に別の知識候補が隠されている可能性がある。そこで F を展開して新たな *field-of-view* を生成し、それらの事例群に対して ID3 法による知識候補の発見を続行する。なお、新たな *field-of-view* に対応する事例集合からは $S_{knowledge}$ に含まれる事例群は除去しておく。

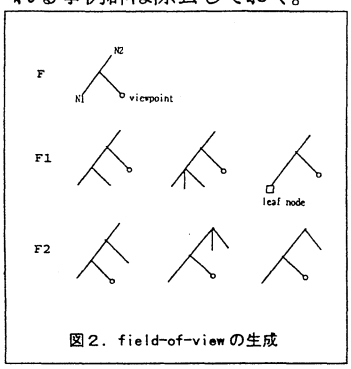

図2. field-of-view の生成

ところで、F を展開する場合、*field-of-view* の端に位置する展開可能な節点は N1, N2 の2個である。

これらを展開すると例えば図2中の *field-of-view* 群 F1, F2 が出現し、いずれの *field-of-view* 群を選択するかが問題となる。これらの各 *field-of-view* に対応する事例数を集計し、何らかの規範により最適の展開を選択することとなる。この規範としては、(1) *field-of-view* 毎に分類される事例群で「が」、「は」の識別に対する情報量が多いこと、(2) 知識の発見が期待できない少数事例群への分割が少ないもの、の2種を組み合わせることが適当であろう。今回の報告では、事例数が 100 以下になるような分割は無意味と考え、後者の基準のみを用いて展開すべき edge_node の選択を行った。

図3に、基本的なアルゴリズムを示す。ここで各 Fi は *field-of-view* を、また Si は Fi を部分木として含む事例の集合を表し、F_candidates は探索対象として保持されている *field-of-view* のリストを示す。また、func_eval(F_i, S_i)は[F_i, S_i]からの知識候補の発見可能性を評価する関数、expand_F は *field-of-view* の端に位

```
1.  F0 := viewpoint; S0 := all instances
2.  F_candidates := F0
3.  loop until null(F_candidates)
4.    select Fi with maximum func_eval(Fi, Si) from F_candidates
5.    Given Fi & Si, classify Si by ID3
6.    Set Sknowledge using some criterion for tree pruning
7.    Si := Si - Sknowledge
8.    for each edge_nodej in Fi
9.      ([Fi,Si], [Fj+1, Sj+1],.. ) := expand_F([Fi, Si], edge_nodej)
10.   end for
11.   select edge_nodej with maximum sum(func_eval(Fj, Sj))
12.   delete Fi from & append (Fj, Fj+1,.. ) to F_candidates
13. end loop
```

図3.　アルゴリズム

置する節点 edge_node の拡張により新たな *field-of-view* 候補を生成する関数である。

上記の過程により帰納的に F_i が生成され、それらの中でもっとも知識の発見されることが期待される F_i, S_i に対して、順に ID3 法による解析が進行する。

3－2　知識候補の発見

field-of-view と対応する事例群が与えられたとき、すべての事例は *field-of-view* の範囲内において等価なトポロジーを有することになる。したがって field-of-view 内のすべての節点に付随するすべての属性を、通常の属性／属性値の対として表現できる。そのため、多くの知識発見手法が適用可能であるが、ここでは最も簡単と考えられる ID3 法を使用する。

属性としては節点の属性として表現できるならどのようなものでも利用可能であるが、ここでは各節点上の文法的属性１種（G: 葉節点の品詞名または中間節点の合成関係子）とトポロジカルな属性３種（P: 上位節点の数、C: 下位節点の数、V: 当該節点から *viewpoint* への経路が通る隣接節点）を使用した。図１の構文木中のいくつかの節点には、これら４種の属性値が付されている。生成された分類木からの知識選択の基準としては、「が」、「は」の識別率が75%以上でかつ事例数が 10 以上とした。

3－3　知識候補から有用な知識への洗練

知識の候補が得られても、そこで使われている表現は通常の日本語文法で使用されるものとは異なっている。また探索過程自体が発見的なものであるため、不要な属性値が混入している可能性や、反対により精細な分類を行うべき可能性も存在する。実際、次節で示すように、ID3 法により分類された結果の属性値だけでは、解釈は困難であった。

そこで、知識候補の表現と分類された事例を照合し、通常の文法的知識へと洗練する作業を行った。また、任意の知識表現パターンを入力としてコーパスデータを検索するプログラムを開発し、種々の仮説の正当性をチェックした。

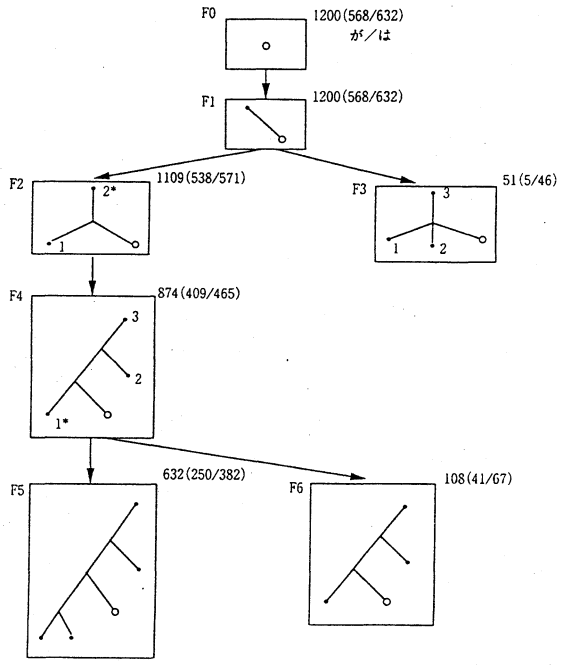

図4.　field-of-view の展開
○：viewpoint 節点、　＊：展開に用いた節点

4　結果と議論

利用した *field-of-view* とその生成過程を図4に示す。ここで各 *field-of-view* の右上に示した数は、そこに属する事例数とその中での「が」と「は」の内訳を表す。なお、図中の F3 は事例数が少なく、知識候補の探索に使用しなかったが、F1 から生成される多数の *field-of-view* の1例として図に含めた。

各 *field-of-view* における ID3 解析の実行結果から条件を満たす 10 種の知識候補を発見した。さらに、それぞれについて、全 1200 事例に対して再度検索を行って得た数との比較を行った。以上の結果から4つの特徴的なパターンが検出できた。

（1）助詞「で」、「に」に続く「は」
（2）読点直前の「は」
（3）文の2番目の節に現れる「が」
（4）文の先頭節に現れる「は」

5　まとめ

括弧付きの EDR コーパスを使用して、古くから国語学において議論され続けている「が」と「は」の問題に対して、構文木パターンを直接解析する方法を提案・適用した。未だ部分的な解析にすぎないが、その結果からいくつかの文法的知識を確認することができた。また、「は」、「が」の直後にある読点の有無や、「は」は文頭節に、「が」は2番目の節に現れやすいというような統計的な傾向も確認することができた。

これらの知見の多くは、すでに国語学の研究において広く知られているものである。しかしながら、「が」と「は」の用法に関する研究という単純な題材を与えるだけで、自動的に知識の候補を得ることができ、さらにそれらの解析を進めれば確定的な文法知識や用法に対する統計的傾向の知見を得られることになる。通常のコーパス検索ツールを利用するだけならば、細部に至るまですべての進め方を研究者が指示する必要があることと比べると、本報告で示した方法論は研究の発想支援を高度に進めたものといえよう。

今後、利用者インターフェース部分の整備をはじめ、より簡便に多くの知見を得ることができるようにシステムを発展させる予定である。また、多くの問題に適用して経験を蓄積し、各種規範の最適化を行うとともに、現在未活用である EDR コーパスの意味構造を利用した解析も行っていきたい。

参考文献

[1] 竹沢寿幸、末松博：音声・テキストコーパスとその構築技術、標準化動向、人工知能学会誌、Vol.10, 168-180 (1995).

[2] EDR: EDR 電子化辞書 1.5 判仕様説明書、EDR TR2-006、日本電子化辞書研究所 (1996).

[3] 雄山真弓、岡田孝、李貴峰： 構文解析木を対象とするデータ解析法の研究（1）方法論についての一考察、シンポジウム：人文科学における数量的分析、東京 (1996).

[4] Guifeng Li, Mayumi Oyama and Takashi Okada : Knowledge Discovery from Syntactic Trees, 1996 年度人工知能学会全国大会（第10回), 08-01, 東京 (1996).

[5] 金田一春彦、林大、柴田武編：日本語百科大事典、大修館書店 (1988).