社団法人　情報処理学会　研究報告
IPSJ SIG Technical Report

2003－ＮＬ－158　(13)
2003／11／7

接続助詞の結合順位に基づく複文の構文解析

市丸　夏樹†　　飛松　宏征†

† 九州大学システム情報科学研究院/府　〒812–8581 福岡市東区箱崎 6–10–1
E-mail: itimaru@is.kyushu-u.ac.jp, tobimatu@lang.is.kyushu-u.ac.jp

あらまし　本研究は，PCFG を用いた日本語の構文解析の特に複文や重文等の長い文に対する解析精度を向上させることを目的とする．従来の用例とシソーラスに基づく PCFG を用いて長い文を構文解析した場合，複数の従属句相互の係り受け関係の曖昧性により非常に多くの構文木候補が生じることがあった．そこで我々は接続助詞等のすべての機能語間の線形結合順位をテキストコーパスから自動的に獲得し最適化する手法を考案し，求められた結合順位を PCFG の生成規則の適用確率値の計算に組み込むことを試みた．その結果，南氏の従属句の分類に従った複数の構文木候補の中から，ほぼ唯一の最尤解を選び出すことが可能となり，EDR 日本語コーパスを用いた構文解析実験では約 75%の係り受けの正解率を得ることができた．

キーワード　構文解析，確率文脈自由文法，係り受け，接続助詞，結合順位，テキストコーパス．

A Method of Parsing Japanese Complex Sentences by Prioritizing Postpositional Particles

Natsuki ICHIMARU† and Hiroyuki TOBIMATSU†

† Graduate School of Information Science and Electrical Engineering, Kyushu University
Hakozaki 6–10–1, Higashi-ku, Fukuoka-shi, 812–8581 Japan
E-mail: itimaru@is.kyushu-u.ac.jp, tobimatu@lang.is.kyushu-u.ac.jp

Abstract　In parsing long Japanese complex sentences by example-based PCFG, the former parsers have produced many parse-tree candidates which have the same probability of occurrence because of ambiguity between modifier phrases. In this paper, we propose an algorithm to optimize the linear priority over all postpositional particles using text corpus. And the priority can be used to control the probability of rule application, and choose the best tree which has the most natural syntax structure. As a result of an experiment using EDR Japanese corpus, the success rate of choosing the correct combination between postpositional particles and phrases turns out to be about 75%.

Key words　PCFG, parsing, priority, postpositional particles, compound sentence, text corpus.

1. はじめに

日本語文は，連用修飾句に修飾された動詞がまた従属句を成すという入れ子構造を持っている．それらの従属句の相互の係り受け関係を特定することは計算機にとって容易ではないため，1 文中に複数の従属句を内蔵する複文や重文を構文解析する際には多量の解候補が生じてしまうことがあった．我々の研究室では用例とシソーラスに基づく PCFG を用いて構文解析を行う手法 [11] についての研究を行ってきたが，係り側と受け側の従属句の組み合わせは多種多様であり，従来の文法では構文木を絞り込むことができなかった．

そこで，この問題に対処するために南氏による従属句の分類 [18], [19] を用いることを考える．南氏の説によると，日本語の従属句は文節末尾の接続助詞等によって A 類, B 類, C 類の 3 分類に分けられる．A 類の句を B 類や C 類の句が修飾することはできず，同様に B 類の句が C 類の句を要素とすることは許されない．これらの制約によって従属句の係り受け関係を制限することによって，解候補の中からその文の最も自然な解釈だと思われるような解を選びだすことができる．南氏の分類を採用した先行研究としては，NTT の白井氏らによる係り受け解析システム [2]，京都大学の依存構造解析プログラム KNP [1] などが存在し，いずれにおいても日本語文の係り受け解析に接続助詞の分類を組み込むことによって，1 文単位の解析で 90%以上という非常に高い正解率が得られている．そこで本稿では，南氏の分類を細分化し最適化した上で PCFG に組み込むことを考える．

南氏の3分類に従って構文解析を行えば，隣接する接続助詞の分類が異なっている場合には，従属句間の係り受け構造が一意的に決まることになる．しかし，A類同士，B類同士，C類同士といった同じ分類に属する接続助詞が連続する部分では，どちらを先に結合させるかを決定できないため曖昧性を完全に絞り込むことはできない．ところが実際の構文解析では，正しい唯一の構文木が要求される場合が多いため，複数の候補を出力するにしても，結局は各々の構文木に対して何らかの評価値を付け，最も確らしいものを選び出す必要がある．そこで我々は，全ての接続助詞あるいは全ての機能語間の結合順位をテキストコーパスから求めることを考え，結合順位の最適化を効率的に行うアルゴリズムを開発した．

そして，求められた機能語間の結合順位をPCFGの生成規則の適用確率の計算過程に組み込むことによって，導出を制御する文法を作成した．この文法では，コーパスから収集した機能語の左右の結合性に従って結合方向を制限することによって，同じ接続助詞が連続した場合にも入力文に対する導出木をほぼ一意に決定できるようになった．また，結合順位と結合性を制約として用いた場合に20%程度受理できない文が生じたため，その対策として制約を若干緩めることを試み，ほぼ100%の入力文を受理できるようにした．その結果，構文的に最も自然な構造に対して高い優先度を与えるような優先付けを行いながら，南氏の制約に必ずしも従わない一般の文を受理することができるモデルを構築することができた．

EDR日本語コーパス[16]から求めた結合順位を用いて提案文法を構築し，構文解析実験を行ったところ，3分類に読点を付加した6分類を用いた場合と比べて係り受けの正解率が約6ポイント向上し，未学習の試験データに対して約75%の正解率が得られることがわかった．

2. 接続助詞の結合順位

2.1 南氏の従属句の分類

本研究では，国立国語研究所の南不二夫氏（現大阪外語大教授）による日本語の構造に関する説，とりわけ従属句と接続助詞の分類に着目し，従属句間の係り受け関係の絞り込みに用いることを考える．

南氏の説[18], [19]によると，日本語の接続助詞は，「ながら」「つつ」などイベントや状況を記述するA類，「ので」「のに」など客観的な判断による原因・理由などを表すB類，「が」など書き手の主観的な判断による逆説表現などを表すC類という3つに分類することができる．句の末尾の接続助詞の分類に応じて，その句の構成要素になることができる語句はある程度限定され，A類の従属句にB類やC類の連用修飾句が係ることは非常にまれであり，同様にB類の従属句はC類の句を要素とすることは少ない．これによって日本語の従属句は接続助詞の結合順位によって定められた階層構造を成すことになる．

南氏の従属句の分類によると，C類よりB類，B類よりA類が先に結合するわけであるから，これをA類，B類，C類に属する接続助詞間の結合順位として捉えることができる．

表1 南氏による接続助詞の3分類．

Table 1 The categories of Japanese postpositional particles by Minami.

A類	B類		C類
ながら (継続)	て	たら	が
つつ	と	なら	から
て	ながら (逆接)	ても	けれど
	ので	ず	し
	のに	ないで	て
	ば		

A類の接続助詞 $<$ B類の接続助詞 $<$ C類の接続助詞.

2.2 佐伯氏の格助詞の語順に関する研究

述語が持つ格要素についても，主格は目的格より前に現れやすいといった，統計的にみて用いられることが多い語順というものが存在する．無論この条件を満たさない文も多くの場合許容されるが，この結合順位に従った構造を持つ文の方が，より自然で座りのよい語順として感じられるようである[19]．

これを格助詞が動詞と結合する順位として捉えれば，格助詞の結合順位を次のように表すことができる．

（時間）に＞（場所）で＞が＞に＞を.

3. テキストコーパスを用いた結合順位の最適化

3.1 接続助詞の分類の細分化

まずここで，接続助詞の「ので」と「ても」について考える．南氏の分類ではどちらもB類に分類されているが，たとえば次の例文では「ても」の方が先に結合しそうに思われる．

彼は若いので、激しい労働で疲れても、すぐ回復する.

EDRコーパスを見てみると，「ので」と「ても」が同一文中に現れる17文のうち，16文は「ても」が先に結合するものである．従って，「ても」の方を「ので」よりも先に結合させれば，この組み合わせに関してはおよそ94%程度の正解率が見込まれる．このように，南氏の分類では同じカテゴリに分類されていても，どちらかというと先に結合されることが多いものが存在する．

構文木を含んだテキストコーパスには実際の文中で生じる結合関係が多く含まれているため，それを用いて様々な接続助詞間の実際の結合の出現頻度を獲得することができる．コーパス中で先に結合することが多い方を先に結合させれば，統計的にはある程度の正解率が得られるのではないかと思われる．そこで我々は，結合順位の有効性の調査を兼ねて，接続助詞を1語1分類にまで細分化し，全ての接続助詞間に結合順位を設定することにした．

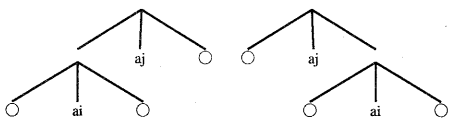

図 1　優先関係 $a_i < a_j$ を支持する構文木の例.
Fig. 1　Sample trees that support the priority $a_i < a_j$.

3.2　テキストコーパス中の構文木からの結合数の獲得

テキストコーパス中の複数の接続助詞を含んだ構文木の中から図 1 のような形の包含関係にあるものを抽出し，各文中での推移的閉包を求めることによって，結合の包含関係を持つ a_i, a_j の組を収集した.

接続助詞の集合を A とするとき，$a_i \in A$ の方が $a_j \in A$ よりも先に結合しやすいこと，すなわち a_i の結合順位が a_j よりも高いことを次のように表すものとする.

$$a_i < a_j. \tag{3}$$

結合の優先関係 $a_i < a_j$ を支持する文の数を $M(a_i, a_j)$ とおくと，$M(a_i, a_j)$ は $a_i < a_j$ を支持する正例の文数であり，$M(a_j, a_i)$ はその逆を支持する反例の文数である．EDR 日本語コーパスから収集された $M(a_i, a_j)$ の値を表 2 に示す．表中では，右上三角行列部分が $a_i < a_j$ の正例，左下部分が反例となっている.

3.3　結合順位に対する評価値

いま，n 個の接続助詞 $a_i \in A$ からなる並び $L = (a_1, \cdots, a_n)$ が与えられたとして，最もこの正例が多くかつ反例が少なくなるような順番に並べ替えたい.

そのためにまず，$a_i < a_j$ の正例と反例の支持文数の差を $E_L(i,j)$ とし，結合順位の降順に並んだ $L = (a_1, \cdots, a_n)$ に対する評価値 $F(L)$ を，その合計を用いて定義する.

$$E_L(i,j) = M(a_i, a_j) - M(a_j, a_i). \tag{4}$$

$$F(L) = \sum_{i=1}^{n} \sum_{j=i+1}^{n} E_L(i,j). \tag{5}$$

評価値 $E_L(i,j)$ は個々の結合の順序関係 $a_i < a_j$ の確らしさを表し，$F(L)$ は L に含まれる順序関係が全体としてテキストコーパス中の構文木に合致する度合いを表す.

3.4　結合順位の最適化

コーパスとの適合度を表す評価値 $F(L)$ を最大にする結合順位 L_{max} を求めることを考える.

$$L_{max} = \arg\max_{L} F(L). \tag{6}$$

もし L_{max} を求めるために，全組み合わせを求めて生成検査法を用いるとすると，$O(n!)$ の計算量が必要となり現実的でない．そこで，ここでは完全性は保証せずに山登り法による近似解法を用いて準最適解を求めることにする.

並び $L = (a_1, \cdots, a_n)$ の 1 つの要素 a_i の位置を a_j の直前に移動したものを L' とすると，L' および評価値の差分値 $\Delta_L(i,j) = F(L') - F(L)$ は次のように求められる.

i)　$i < j$ の場合.

$$\Delta_L(i,j) = -2\sum_{k=i}^{j-1} E_L(i,k). \tag{7}$$

ii)　$i > j$ の場合.

$$\Delta_L(i,j) = 2\sum_{k=j}^{i-1} E_L(i,k). \tag{8}$$

以上を用いて，結合順位 L を最適化するアルゴリズムは次のようになる.

[結合順位の最適化アルゴリズム]
入力: 頻度順に列べられた接続助詞の並び $(a_1, \cdots, a_n)$.
出力: 評価値 $F(L)$ を最大にする結合順位 $L = L_{max}$.

```
begin
    L ← (a_1).
    for k ← 2 to n do begin
        L 中の F(L) を最大にする位置に a_k を挿入.
        repeat begin
            (i,j) ← arg max_{i,j} Δ_L(i,j).
            if Δ_L(i,j) = 0 then break.
            L 中の a_i を a_j の直前の位置に移動.
        end.
    end.
    return L.
end.
```

接続助詞の並び L の初期値の与え方としては，一度に与えてしまうと収束までに大変時間がかかるため，最も頻度が大きいもの 1 つ $L_1 = (a_1)$ から始めて，頻度順に 1 つずつ最適な位置に挿入することによって結合順位を保ったまま徐々に増加させる．追加によって既存の接続助詞の順位の変更が必要となる場合は，山登り法によって修正するようになっているが，ほとんどの場合修正の必要は生じないため，時間がかかる山登り法の部分が実際に駆動されることはまれである．そのため，このアルゴリズムを用いると，224 分類を 3 分弱で最適化することができる.(注1)

3.5　最適化された結合順位の分析

以上の方法により，EDR 日本語コーパスから接続助詞，連体形，語幹 (ε 連用形) で終わる文節間の結合順位を求めたとこ

(注1)：このアルゴリズムでは同時に 2 つ移動しなければ最大値に到達できない場合を考慮していないため，厳密に言うと必ずしも $F(L)$ を最大にする解が得られる保証はない．しかし，この場合の評価値の変動は比較的なだらかであり，ここではむしろ評価値の最大値付近で同じ評価値を持つ並びが多く存在するという問題が生じたため，各接続助詞を動かしてみて評価値が変動しない範囲を求め，その中心の位置を与えることによって，より安定した結合順位が得られるようにした.

表 2　接続助詞間の結合関係 $a_i < a_j$ を支持する文数 $M(a_i, a_j)$.

Table 2　$M(a_i, a_j)$, the number of sentences that support the priority between postpositional particles $a_i < a_j$, presented in optimized order.

$a_i \backslash a_j$	連用形	語幹	つつ	で	て	ながら	ば	ても	つつ、	から	のに	て、	ながら、	語幹、	が	で、	ので	連用形、	ものの、	ば、	し、	けど、	ても、	ので、	のに、	から、	けれど、	が、
連用形	—	83	2	36	1293	52	159	60	8	26	9	997	83	739	21	32	49	2259	37	298	68	5	83	282	49	81	8	971
語幹	75	—	0	4	99	3	12	12	1	3	0	67	3	73	3	3	4	201	8	21	10	0	8	15	7	8	0	83
つつ	1	0	—	0	3	0	0	0	0	0	0	1	0	1	0	0	0	5	0	0	0	0	0	0	0	0	0	4
で	22	0	0	—	33	0	4	2	0	1	0	26	1	34	2	0	0	116	0	1	2	1	2	19	1	6	0	33
て	422	41	2	21	—	18	72	26	3	17	10	567	63	970	18	28	49	2624	29	241	70	10	67	290	42	90	5	1205
ながら	13	3	0	1	17	—	0	1	0	0	0	24	0	22	0	1	2	70	2	2	1	0	2	8	0	2	0	19
ば	16	2	0	0	29	0	—	6	0	2	0	20	0	37	1	0	1	116	1	7	21	0	18	31	5	9	0	84
ても	13	0	0	0	11	0	4	—	0	4	0	17	0	14	0	1	4	83	0	8	15	1	11	24	0	12	0	36
つつ、	0	0	0	0	0	0	0	0	—	0	0	1	1	3	0	0	0	3	0	1	0	0	0	0	0	0	0	0
から	5	0	0	1	8	1	0	1	0	—	1	19	0	1	0	1	1	16	4	3	2	2	2	1	5	0	0	11
のに	1	0	0	0	2	0	0	0	0	0	—	9	0	5	0	0	0	23	0	0	0	0	0	2	0	1	0	7
て、	60	9	1	6	157	6	16	5	1	9	2	—	12	145	6	2	7	386	10	49	12	4	19	59	9	17	4	267
ながら、	3	0	0	0	11	1	0	0	0	0	0	8	—	14	0	1	0	26	0	1	0	0	1	0	0	0	0	5
語幹、	47	12	0	6	128	7	16	4	1	3	0	126	13	—	9	5	6	337	5	45	5	0	5	18	7	7	0	154
が	5	0	0	0	3	0	0	0	0	0	0	6	0	3	—	0	2	22	0	1	0	0	0	2	0	1	0	3
で、	3	1	0	0	6	0	1	0	1	0	0	2	0	4	0	—	0	20	0	1	1	0	3	3	1	4	0	6
ので	1	1	0	0	4	0	1	0	0	0	0	8	1	5	0	0	—	30	0	3	1	0	1	1	0	0	0	50
連用形、	136	13	2	15	304	9	34	10	1	9	2	287	22	251	15	4	26	—	23	125	13	2	29	99	18	34	2	513
ものの、	1	0	0	0	1	0	0	1	0	0	0	6	0	1	0	0	0	19	—	0	1	0	0	0	0	2	0	3
ば、	11	0	0	2	9	0	4	5	0	1	0	11	0	15	1	1	1	75	0	—	10	0	15	10	0	13	1	64
し、	2	0	0	0	6	0	5	1	0	0	0	9	0	5	0	1	1	7	0	9	—	1	4	4	2	7	0	13
けど、	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	—	1	0	0	0	0	0
ても、	7	1	0	0	5	0	4	4	0	0	0	5	0	4	1	0	1	26	0	9	2	0	—	16	1	2	0	12
ので、	10	1	0	2	37	2	2	2	0	0	0	20	1	4	0	0	0	36	0	1	2	0	1	—	2	1	2	39
のに、	1	0	0	0	7	0	1	0	0	1	0	7	0	0	0	1	1	13	0	1	0	0	0	2	—	1	0	2
から、	2	0	0	0	13	0	1	2	0	1	0	3	0	3	0	0	0	6	0	3	1	0	2	0	0	—	0	11
けれど、	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	—	0
が、	19	3	0	3	30	0	2	0	0	0	0	31	0	45	1	2	5	196	0	5	2	0	1	20	0	3	0	—

表 3　評価値 $F(L_{max})$ の最適化の結果.

Table 3　The result of optimizing $F(L_{max})$.

L	分類数	正例文数	負例文数	$F(L_{max})$
接続助詞のみ	28 分類	19,033	3,411	15,622
全ての助詞等	224 分類	3,336,536	617,186	2,719,350

ろ，前述の表 2 のような結果が得られた．この表は最適化された状態のものであり，図中の接続助詞は結合順位の高い順に並べられている．なお，その際の評価値 $F(L_{max})$ は表 3 上段のようになった．正例数は負例数より圧倒的に多く，有効な結合順位が求まっているものと思われる．

こうして求められた接続助詞列について，次のようなことが観察された．

- 南氏の 3 分類は最適化された結合順位の中でも保存されており，3 分類間の優先関係も，若干の入れ替わりはあるものの概ね保存されている．
- 読点について，南氏の 3 分類に属する接続助詞 a, b, c に読点「、」が後接したものを $a_、 < b_、 < c_、$ と置くと，それらの間には次のような結合順位が成り立つ傾向が見られる．

$$a < b < c \lesssim a_、 < b_、 < c_、. \tag{9}$$

南氏の分類は小説や随筆について調査されたものであったが，辞典や新聞・雑誌記事からなる EDR コーパスについても統計的には確かに成り立っているようである．一方，読点については従来手法 [2] で用いられている順序とは異なった順序が求まっている．

なお，この結合順位に対する反例が特に多かったのは，南氏の分類で A, B, C の全てに含まれる多義を持つ接続助詞「て」，「で」，および連用形や，「煮 ても 焼い ても 食えない」のように並列句を成す接続助詞であった．

4. PCFG への結合順位の導入

接続助詞だけでなく全ての助詞と活用語の活用形を含めた計 224 語の機能語について，同様に結合順位を最適化したところ，表 3 下段のような結果が得られた．この結合順位 L_{max} を用例とシソーラスに基づく PCFG に組み込むことを考える．

4.1 用例とシソーラスに基づく PCFG

ここでは，機能語間の制約の取り扱いを容易にするため，結合子となる機能語 f を中心とする 3 分木の形式に正規化された構文木を成す文法の生成規則を示す．

非終端記号に含まれる品詞を c, c'，主辞を h, h'，機能語を f, f'，主辞側が H で修飾句側が M という値を持つ係り受けに関するフラグを k，結合子 f からみて左側が L で右側が R という値をとる方向を表すフラグを d とし，最大の結合子が f である句を表す非終端記号を $c(h, f)$，機能語 f の d 方向側にのみ結合可能な非終端記号を $c(h, f, d, k)$，付属語 f_i によって代表される付属語列を導出する非終端記号を $F(f_i)$，単語 w_i に相当する概念を持つ主辞を h_i，自立語または接辞からなる終端記号列を $w_1 \cdots w_{n'}$，付属語または活用語尾からなる終端記号列を $f_1 \cdots f_{n''}$ とおくと，提案文法は次のように表される．

$$c(h, f) \xrightarrow{p_1} c(h, f, L, H)\ F(f)\ c'(h', f, R, M). \tag{10}$$

$$c(h, f) \xrightarrow{p_2} c'(h', f, L, M)\ F(f)\ c(h, f, R, H). \tag{11}$$

$$c(h, f, d, H) \xrightarrow{p_3} c(h, f'). \tag{12}$$

$$c(h, f, d, M) \xrightarrow{p_4} c(h', f'). \tag{13}$$

$$c(h_i, f_0) \xrightarrow{p_5} w_1 \cdots w_i \cdots w_{n'}. \tag{14}$$

$$F(f_i) \xrightarrow{p_6} f_1 \cdots f_i \cdots f_{n''}. \tag{15}$$

機能語によって生成規則数がむやみに増加することを防止するため，ルール (10)，(11) に現れる機能語は右辺第 2 項から終端記号に直結した f のみとなっている．ルール (12) では主辞側を汎化せず，直接 h を継承する．一方，ルール (13) は，品詞 c の汎化された主辞 h から，同じ品詞でシソーラス中の子孫

表 4　提案文法による構文解析の実験結果.

Table 4　The results of parsing experiment by the proposed grammar.

入力文タイプ	入力文数	3 分類+読点の 6 分類 受理された文の数	出力された構文木数	係り受けの正解率	提案手法 a. 制約版 受理された文の数	出力された構文木数	係り受けの正解率	提案手法 b. 制約緩和版 受理された文の数	出力された構文木数	係り受けの正解率
未学習	985	960	1,246	68.6%	782	782	74.6%	982	999	74.8%
学習済	988	969	1,312	76.4%	754	754	76.9%	985	1,009	80.3%

ノードに相当する h' を主辞とする句を導出する．ルール (14) の h_i，ルール (15) の f_i としては，品詞間の優先順位 [7] により右辺を代表する語を 1 つ選ぶ．ルール (14) には便宜上，終端であることを表す記号 f_0 を与える．

4.2　生成規則の適用確率の設定

各生成規則の適用確率 p_1, p_2, p_5, p_6 は最尤推定法 [4] により求められる．ルール (12) の適用確率 p_3 を厳密に確率的に求めることは難しいため，機能語 f' によって先に結合された部分木を f によって結合することが許容されるか否かを表すフラグ $u_d(f, f')$ を使用する．ルール (13) の適用確率 p_4 の方は，まずシソーラス中の上位ノード h から子孫ノード h' を導出する確率 $p(h \stackrel{*}{\Rightarrow} h')$ と結合フラグ $u_d(f, f')$ とに分離して，各々独立であると仮定する．

$$p_3 = u_d(f, f'). \tag{16}$$

$$p_4 = p(h \stackrel{*}{\Rightarrow} h') \cdot u_d(f, f'). \tag{17}$$

シソーラスノード h の子孫に属する単語数を $|h\downarrow|$ で表すものとすると，全単語の導出木を低頻度で与えて最尤推定することにより，導出確率 $p(h \stackrel{*}{\Rightarrow} h')$ は近似的に次のように求められる．

$$p(h \stackrel{*}{\Rightarrow} h') = \prod_{\substack{h \stackrel{*}{\Rightarrow} h_i \\ h_j \stackrel{*}{\Rightarrow} h'}} p(h_i \rightarrow h_j) \tag{18}$$

$$\approx |h\downarrow| \,/\, |h'\downarrow|. \tag{19}$$

一方，結合フラグ $u_d(f, f')$ は場合分けにより次のように表される．

a)　結合順位を制約として用いる場合

ここで，同じ結合子 f が連続して現れる場合の曖昧性を完全に絞り込むためにテキストコーパスから収集した左結合を支持する文の数を $L(f)$，右結合を支持する文の数を $R(f)$ とすると，結合子 f, f' 間の結合順位と左右の結合性が共に成り立つ際に 1，さもなければ 0 となる結合フラグ $u_d(f, f')$ は次のように定義される．

i)　$f \neq f'$ の場合は，$f > f'$ のときのみ結合を許す．

$$u_d(f, f') = \begin{cases} 1 & if\ f > f', \\ 0 & if\ f < f'. \end{cases} \tag{20}$$

ii)　$f = f'$ で $L(f) \geq R(f)$ の場合，f は左結合とする．

$$u_L(f, f') = 1, \qquad u_R(f, f') = 0. \tag{21}$$

iii)　$f = f'$ で $L(f) < R(f)$ の場合，f は右結合とする．

$$u_L(f, f') = 0, \qquad u_R(f, f') = 1. \tag{22}$$

b)　結合順位の制約を緩和した場合

最適化した結合順位による制約は本来絶対に満たさなければならない制約ではないため，これを制約として用いてしまうと，制約に従わないあまり模範的でない文は受理できないということになってしまう．

そこで，結合順位の制約を守らない場合にもその導出を許容し，適用確率が 0 にならないようにするため，$f < f'$ のとき接続助詞間の順位の差 $f' - f$ に応じて指数関数的に減衰する微少な値を与えることを試みた．この場合の結合フラグ $u'_d(f, f')$ は次のように表される．

$$u'_d(f, f') = \begin{cases} 1 & if\ f > f', \\ 1/\alpha^{f'-f} & if\ f < f'. \end{cases} \tag{23}$$

この α の値を大きくすると制約が強くなり，徐々に式 (20) に近い値をとるようになる．実験的によると $\alpha = 2$ 程度が適当なようである．$f = f'$ の場合は a) と同様である．

以上のように制約を緩和することによって，構文的により自然な構造に対して高い優先度を与えつつ，南氏の制約に必ずしも従わない一般の文も受理できるようになった．

4.3　解析実験

ここでは簡単のため，入力文は文節構造解析済であり，かつ語義の曖昧性も解消済として，EDR 日本語コーパス中の形態素

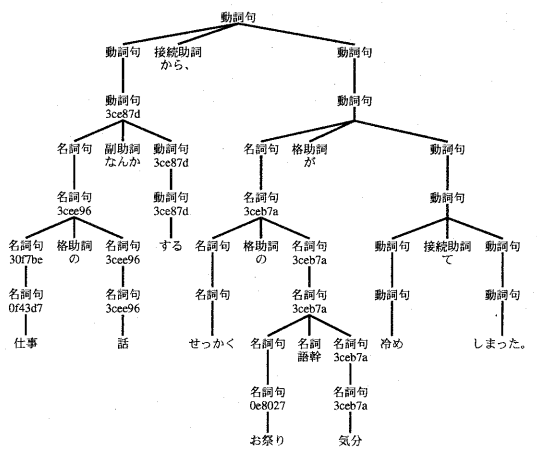

図 2　提案文法の導出木の例.

Fig. 2　A tree example by the proposed grammar.

を入力とする構文解析実験を行った．学習データとしてはコーパスから後述の 1,000 文を除いたものを用い，助詞の品詞を細分化した上で結合順位の最適化にも使用した．試験データとしては学習データに含まれない 1,000 文と，学習済みデータの中から 1,000 文を用意し，正解データとしては，コーパスに含まれる構文木を提案文法に沿った 3 分木の形式に正規化したものを用いた．

試験データのうち頻度 10 未満の機能語を含まない文に対して構文解析実験を行ったところ，表 4 のような結果が得られた．結合順位を制約として用いた場合には，20%程度の不受理文が生じたのに対し，制約を緩和した場合にはほぼ 100%の入力文を受理できるようになった．いずれもほぼ一意の最尤解が得られている．

表中の係り受け正解率とは，末尾とその 1 つ前の機能語を除き，各機能語からみて被修飾句側の head が正解構文木と等しくなった割合である．南氏の 3 分類とそれに読点を付加した 6 分類を用いた場合と比較すると，約 6 ポイントの向上がみられた．未学習の試験データには約 75%，学習済のデータに対しては約 80%の正解率が得られている．

係り受けの正解率の値はまだ十分だとは言えないが，その主な理由は今のところ，EDR コーパスの粗い品詞分類をそのまま用いている部分が多いということと，係り受けルールを獲得するための学習データの量が十分ではないためであると考えられる．

提案文法による導出木の例を図 2 に示す．この構文木は 3 分木形式となっているが，より標準的な形式への変換も可能である．

5. おわりに

テキストコーパスから獲得した結合関係によって接続助詞等の結合順位を最適化するアルゴリズムを開発し，その結合順位を組み込んだ PCFG の構成法を提案し，実験によりその有効性を確認した．今後は品詞を細分化してルールを精錬し，学習データ量を増加させることによって，さらなる正解率の向上を目指す．

この文法では，導出木の生起確率をどの程度結合順位を満たしているかを表す指標とみなすことができるため，将来的には，自然な語順で文を生成するシステムや，文体の評価により読みやすい語順への書き換えを促すような推敲支援システム等への応用が考えられる．

謝　辞

この研究の初期の段階においてご指導戴きました，日高達 現九州大学名誉教授に感謝の意を表します．なお，この研究の一部は平成 14–15 年度科学研究費補助金 (課題番号 14780294) により行われました．

文　献

[1] 黒橋禎夫. 結構やるな，KNP. 情報処理, Vol. 41, No. 11, pp. 1215–1220, 2000.

[2] 白井諭, 池原悟, 横尾昭男, 木村淳子. 階層的認識構造に着目した日本語従属節間の係り受け解析の方法とその精度. 情報処理学会論文誌, Vol. 36, No. 10, pp. 2353–2361, 1995.

[3] 松ヶ下大輔. 連用修飾述語句の話題の転換力を考慮した文脈自由文法の提案. 九州大学大学院システム情報科学研究科修士論文, 2000.

[4] 日高達. 確率文法. 情報処理, Vol. 36, No. 2, pp. 169–176, 1995.

[5] 三井士和. 結合力に基づく述語句の係り受け解析. 九州大学工学部電気情報工学科卒業論文, 2001.

[6] 三井士和, 市丸夏樹, 日高達. 接続助詞の線形結合順位に基づく構文解析. 第 54 回九州支部連合大会講演論文集, p. 637. 電気関係学会, 2001.

[7] 松岡稔公. 述語文節間の係り受けの曖昧性を解消するための日本語係り受け文脈自由文法の提案. 九州大学大学院システム情報科学研究科修士論文, 1999.

[8] 市丸夏樹, 中村貞吾, 日高達. 名詞シソーラスを用いた派生語の処理. 技術研究報告 [言語理解とコミュニケーション] NLC 92-17, pp. 39–46. 電子情報通信学会, 1992.

[9] Natsuki Ichimaru, Teigo Nakamura, Yoshiaki Miyamoto, and Toru Hitaka. Example-based stochastic analysis of japanese derivative words. *Natural Language Processing Pacific Rim Symposium '93*, pp. 368–371, 1993.

[10] 市丸夏樹, 中村貞吾, 宮本義昭, 日高達. 用例に基づく派生語の確率的解析. 自然言語処理研究会研究報告 93-NL-97, pp. 21–28. 情報処理学会, 1993.

[11] 市丸夏樹, 中村貞吾, 宮本義昭, 日高達. シソーラスと確率文法による派生語解析. 情報処理学会論文誌, Vol. 36, No. 4, pp. 849–858, 1995.

[12] 市丸夏樹, 日高達. シソーラスを利用した複合語の仮名漢字変換のための確率複合語文法. 九州大学工学集報, Vol. 68, No. 6, pp. 557–564, 1995.

[13] 市丸夏樹, 日高達. 複合語の仮名漢字変換のための解析アルゴリズム. 九州大学工学集報, Vol. 69, No. 3, 1996.

[14] 市丸夏樹, 中村貞吾, 日高達. PCFG による派生語処理手法の比較と検討. 九州大学システム情報科学研究科 研究科報告, Vol. 4, No. 1, 1999.

[15] 市丸夏樹, 日高達. 述語句の分類を導入した PCFG による構文解析. 第 53 回九州支部連合大会講演論文集, p. 620. 電気関係学会, 2000.

[16] 日本電子化辞書研究所. EDR 電子化辞書 version 2. CDROM, 1999.

[17] 飛松宏征. 述語句間の結合力に基づく係り受け解析の精度の向上. 九州大学工学部電気情報工学科卒業論文, 2001.

[18] 南不二夫. 現代日本語の構造. 大修館書店, 1974.

[19] 南不二夫. 現代日本語文法の輪郭. 大修館書店, 1993.

[20] 長尾真（編）. 自然言語処理. 岩波講座 ソフトウェア科学 15. 岩波書店, 1996.

[21] 松井裕二. 日本語係り受け文脈自由文法 ～文節における統語制約の導入～. 九州大学大学院システム情報科学研究科修士論文, 1998.