自然言語処理 121－8
（1997. 9. 11）

係り受け情報を利用したパーザの効率化とロバスト解析への応用

今一 修，藤尾 正和，山根 洋平，宮田 高志，松本 裕治
奈良先端科学技術大学院大学 情報科学研究科
{osamu-im,masaka-h,youhei-y,takashi,matsu}@is.aist-nara.ac.jp

曖昧性の取り扱いは自然言語処理における重要な問題の一つである．統語解析を行なう際にも曖昧性によって処理時間が増大するため，曖昧性をうまく解消することが必要になってくる．本稿では，そのための情報として統計的に得られた係り受け情報を利用する．各係り受け関係には，そのもっともらしさに応じて確率が付与されている．係り受け情報を利用するために，チャート法のアルゴリズムに修正を加え，統語解析の効率化を図った．講談社和英辞典の例文100文について実験したところ，生成される弧の数が約3分の1，解析時間が約8分の1になり，係り受け情報を用いることにより統語解析が効率的に行なえることがわかった．また，係り受け情報をロバスト解析に利用する方法についても述べる．

[キーワード] 係り受け構造，チャート解析，効率化，ロバスト解析

An Efficient Parsing Method using Dependency Information and its Application to Robust Parsing

Osamu IMAICHI, Masakazu FUJIO, Youhei YAMANE,
Takashi MIYATA, Yuji MATSUMOTO
Graduate School of Information Science, Nara Institute of Science and Technology

Ambiguity resolution is one of the important problems in natural language processing. Since the processing time increases according to ambiguity, it is necessary to resolve ambiguity appropriately. In this paper, we use probabilistic dependency information as information for resolving ambiguity. Probability is assigned to each dependency relation according to its plausibility. In order to use this information, we modify the Chart parsing algorithm. The result of the experiment shows that the number of edges in the Chart decreases to one third, and the processing time to one eighth. We also describe its application to robust parsing.

[keywords] dependency structure, Chart parsing, efficient parsing, robust parsing

1 はじめに

曖昧性をどのように取り扱うかは自然言語処理，特に，統語解析において重要な問題である．統語的曖昧性は，語間の係り受け関係が一意に定まらず複数の候補が考えられる場合に発生するが，統語解析では，そのような曖昧性が増えるにしたがって，解析時間が増えていってしまう．構文的曖昧性の解消は，候補となる係り受け関係の中から，適切な係り受け関係を選択する問題と考えることができる．したがって，統語解析を効率的に行なうためには，もっともらしい係り受け関係から順番に処理を行

$$
\begin{aligned}
vp &\rightarrow 、np, vp \\
np &\rightarrow n, p \\
n &\rightarrow vp, n \\
vp &\rightarrow v \\
v &\rightarrow \text{壊れた} \mid \text{修理した} \\
n &\rightarrow \text{車} \mid \text{太郎} \\
p &\rightarrow \text{は} \mid \text{を}
\end{aligned}
$$

図 1: 文法規則

なっていく必要がある．

統語解析における優先処理の枠組みの代表的なものに確率文脈自由文法 [Charniak93] を用いた統語解析がある．確率文脈自由文法では各文法規則に確率を付与しており，その確率を最大にするような解析結果を優先的に得ることができる．しかし，その確率は文法規則ごとに定義されており，領域依存性の高いテキストを処理する場合には，その確率をその文法で再学習する必要がある．

係り受け確率などの統計情報と制約としての文法規則を独立に考えると，確率文脈自由文法における問題点を克服することができる．つまり，領域依存性の高いテキストに順応したパーザが必要な場合には，文法に修正を加える必要はなく，係り受け確率を再学習するだけでよい．このように，統計情報と制約情報を分けて考えることはシステムの保守・管理の点で自由度が高く，見通しのよいものとなっている．本稿では，統計情報を文法制約の適用順序を制御するための情報として利用することにより，チャート法 [Kay80] による統語解析の効率化を図る．

チャート法による統語解析において，統語的曖昧性に起因する無駄な処理がどの程度行なわれているのかを，例 1 で考えてみる．

例 1: $_0$太郎$_1$は$_2$壊れた$_3$車$_4$を$_5$修理した$_6$

単語間の番号はその位置を示すものである．図 1 の文法規則とチャート法を用いて例 1 を解析すると，図 2 に示すような弧が生成される[1]．これらの弧の中で最終結果を生成するのに必要な弧は，図 2 の一番右の欄に○印を付与したものだけである．つまり，このような単純な文法と短い文においても，不要な弧が 38% (11/29) もあることになる．単語から生成される不活性弧を除けば，不要な弧の割合は 52% (11/21) にもなる．より複雑な文法規則を用いて解析を行なった場合には，不要な弧がさらに多数生成されることになり，処理速度と記憶容量の両方の側面で非効率的である．

[1]係り先位置欄の数字に意味については後述．

2 パーザの効率化

2.1 チャート法

チャート法では，**活性弧**と**不活性弧**と呼ばれる二つのデータ構造を用いて解析を行なう．不活性弧は完成した部分解析木に相当し，活性弧は未完成な部分解析木に相当する．図 2 の例では，(1) 番や (4) 番などが不活性弧あり，(2) 番や (5) 番などが活性弧である．活性弧は，完成した部分解析木になるために必要な要素を**未決定項**と呼ばれるデータ構造として含んでいる．図 2 では未決定項は [?] という記法で表されている．

チャート法で行なう操作には，以下の二つがある．

操作 (a) 活性弧と不活性弧の補完．

操作 (b) 不活性弧から活性弧への展開．

操作 (a) は，隣接する活性弧と不活性弧が存在する場合に，活性弧の最左未決定項を不活性弧で埋める操作である．例えば，(2) 番の活性弧と (3) 番の不活性弧が存在する場合には，(2) 番の活性弧の最左未決定項を (3) 番の不活性弧で埋めることにより新しい不活性弧 (4) 番が生成される．

操作 (b) は，不活性弧が生成された場合に，その不活性弧のカテゴリーを右辺の一番左の要素としてもつ文法規則を参照して活性弧を生成する操作である．例えば，図 2 では，(1) 番の不活性弧と図 1 の文法規則 $np \rightarrow n, p$ から，(2) 番の活性弧が生成される．

2.2 効率低下の原因

不要な弧の生成を抑制するためには，チャート法で実行される操作 (a) と操作 (b) について考える必要がある．図 2 を解析過程を見れば明らかなように，操作 (a) と操作 (b) の盲目的な適用が不要な弧

番号	位置			弧	
	開始	終了	係り先		
(1)	0	1		$[太郎]_n$	○
(2)	0	1	2	$[[太郎]_n[?]_p]_{np}$	○
(3)	1	2		$[は]_p$	○
(4)	0	2		$[[太郎]_n[は]_p]_{np}$	○
(5)	0	2	6,3	$[[[太郎]_n[は]_p]_{np}[?]_{vp}]_{vp}$	○
(6)	2	3		$[壊れた]_v$	○
(7)	2	3		$[壊れた]_{vp}$	○
(8)	2	3	4	$[[壊れた]_{vp}[?]_n]_n$	○
(9)	0	3		$[[[太郎]_n[は]_p]_{np}[壊れた]_{vp}]_{vp}$	×
(10)	0	3		$[[[[太郎]_n[は]_p]_{np}[壊れた]_{vp}]_{vp}[?]_n]_n$	×
(11)	3	4		$[車]_n$	○
(12)	3	4		$[[車]_n[?]_p]_{np}$	△
(13)	2	4		$[[壊れた]_{vp}[車]_n]_n$	○
(14)	2	4	5	$[[[壊れた]_{vp}[車]_n]_n[?]_p]_{np}$	○
(15)	0	4		$[[[[太郎]_n[は]_p]_{np}[壊れた]_{vp}]_{vp}[車]_n]_n$	×
(16)	0	4		$[[[[[太郎]_n[は]_p]_{np}[壊れた]_{vp}]_{vp}[車]_n]_n[?]_p]_{np}$	×
(17)	4	5		$[を]_p$	○
(18)	3	5		$[[車]_n[を]_p]_{np}$	△
(19)	3	5		$[[[車]_n[を]_p]_{np}[?]_{vp}]_{vp}$	△
(20)	2	5		$[[壊れた]_{vp}[車]_n]_n[を]_p]_{np}$	○
(21)	2	5	6	$[[[壊れた]_{vp}[車]_n]_n[を]_p]_{np}[?]_{vp}]_{vp}$	○
(22)	0	5		$[[[[太郎]_n[は]_p]_{np}[壊れた]_{vp}]_{vp}[車]_n]_n[を]_p]_{np}$	×
(23)	0	5		$[[[[[太郎]_n[は]_p]_{np}[壊れた]_{vp}]_{vp}[車]_n]_n[を]_p]_{np}[?]_{vp}]_{vp}$	×
(24)	5	6		$[修理した]_v$	○
(25)	5	6		$[修理した]_{vp}$	○
(26)	3	6		$[[[車]_n[を]_p]_{np}[修理した]_{vp}]_{vp}$	△
(27)	2	6		$[[[[壊れた]_{vp}[車]_n]_n[を]_p]_{np}[修理した]_{vp}]_{vp}$	○
(28)	0	6		$[[[太郎]_n[は]_p]_{np}[[[[壊れた]_{vp}[車]_n]_n[を]_p]_{np}[修理した]_{vp}]_{vp}]_{vp}$	○
(29)	0	6		$[[[[[[太郎]_n[は]_p]_{np}[壊れた]_{vp}]_{vp}[車]_n]_n[を]_p]_{np}[修理した]_{vp}]_{vp}$	×

図 2: チャート法による例 1 の解析過程

$$
\begin{array}{lll}
1 & \rightarrow & 2 \\
2 & \stackrel{1}{\rightarrow} & 3 \\
2 & \stackrel{9}{\rightarrow} & 6 \\
3 & \rightarrow & 4 \\
4 & \rightarrow & 5 \\
5 & \rightarrow & 6
\end{array}
$$

図 3: 係り受け情報

の生成につながっている．このような過剰生成を抑制するために，どの句がどの句に係り得るかに関する係り受け情報を利用することが考えられる．

例えば，図 3 のような単語単位の係り受け情報が利用できるとする．両辺の数字は各単語に付与された番号で，例 1 における単語の右側の番号に相当する．例えば，例 1 では,「太郎」が 1 番,「は」が 2 番，などである．矢印の上の数値は，その式の左辺の単語が右辺の単語にどれくらいの確率で係っているかを示している．確率が 1 の場合は，その数値は省略してある．

このような係り受け情報が利用できると，操作 (a) や操作 (b) を実際に行なうかどうかの判断をそれを基に行なうことができる．つまり，操作 (a) や操作 (b) が実行可能な状況になった場合に，係り受け情報を参照することによって，実際に操作 (a) や操作 (b) を実行するかどうかを決定するのである．図 3 の係り受け情報を利用すると，図 2 の解析過程においては，不要な操作 (a) を禁止することにより×印の弧の生成を抑制することができ，また，不要な操作 (b) を禁止することにより△印の弧の生成が抑制できる．

ここでは説明のために単語間の係り受け関係が利用可能であるとした．実際には，藤尾らが提案している統計的手法を用いた係り受け解析 [藤尾 97] を利用する．彼らのシステムは，入力文を形態素解析した結果を用いて入力文を文節単位に区切り，文節間の係り受け関係をそのもっともらしさに応じた確率を付与して出力する．文節間の係り受け関係を統計的に学習することによって，精度の向上や分野依存性の高いテキストに順応可能な係り受け解析を実現している．このような文節間の係り受け関係を利用する場合には，文節内の各単語間の係り受け確率は 1 と考えて単語間の係り受けに展開することができる．したがって，本稿の枠組みで必要となるのは，どの文節がどの文節に係り得るかという文節間の係り受け確率だけである．

2.3 効率化の方法

2.3.1 操作 (a) の制御

活性弧と不活性弧の補完を行なうことは，係り受け関係を成立させることと等価である．したがって，係り受け情報を利用することにより，操作 (a) の不要な適用を抑制することができる．つまり，活性弧にその係り先がどれかを記述しておき[2]，補完の際には，その活性弧の最左未決定項を埋める不活性弧がその係り先かどうかをチェックすればよい．

チャート法では，以下の二つの条件が満たされていれば補完操作が行なわれる．

条件 (A) 活性弧の終了位置と不活性弧の開始位置が一致すること．

条件 (B) 不活性弧のカテゴリーが活性弧の最左未決定項のカテゴリーと一致すること．

係り受け情報を利用するためには，これに以下の条件を加えればよい．

条件 (C) 活性弧の係り先位置と不活性弧の終了位置が一致すること．

例えば，図 2 において，(2) 番の活性弧と (3) 番の不活性弧との間の補完は上記の条件 (A) から (C) がすべて満たされているので補完操作が実行される．一方，(5) 番の活性弧と (8) 番の不活性弧との間の補完は活性弧の係り先と不活性弧の終了位置が異なっているため条件 (C) を満足することができずに補完操作は実行されない[3]．これは，通常のチャート法が無駄に生成してしまう弧を係り受け情報を用いることによって，その生成を抑制した例である．

2.3.2 操作 (b) の制御

不要な操作 (b) がどのような状況で生じるのかを考えてみる．例えば，連続する三つの語 A，B，C

[2]図 2 では係り先位置欄にその活性弧がどの弧に係るかを記述してある．

[3]後述するが，係り先位置が複数存在する場合は確率の高いものから採用するため，この場合は補完が実行されない．

があり，AがBに係り，BがCに係るとする．このような係り受け関係が与えられた場合，AとBから部分解析木Dをつくり，DとCから部分解析木Eをつくる必要がある．ここで，BとCから部分解析木Fをつくってしまうと，その部分解析木は後に利用されず無駄である．

逆に，BとCからつくられた部分解析木Fが無駄にならないのは，どんな場合を考えてみると，AがCに係り，BがCに係る，というように，Bに係ってくる語がないような場合であることがわかる．

このことをチャート法の上で考えてみると，ある不活性弧を活性弧に展開すべきかどうかは，その不活性弧に係ってくる要素がない場合，つまり，前接する活性弧と補完できない場合であることがわかる．例えば，図2における，(11) 番の不活性弧について考えてみる．係り受け情報を参照することにより，(8) 番の活性弧が (11) 番の不活性弧に係っていることがわかるので，(11) 番の不活性弧を展開して (12) 番の活性弧を生成する必要はない．

操作 (b) を適用するための条件は以下のようになる．

条件 (D) ある不活性弧を活性弧に展開してもよいのは，その不活性弧に係ってくる活性弧が存在しない場合である．

このように，係り受け情報を参照することによって，操作 (b) の無駄な適用を避けることができるのである．

2.4 次候補の利用

操作 (a) と操作 (b) の制御を行なうことによって無駄な弧を生成することなく解析を行なうことができるが，係り受け情報は必ずしも正しいとは限らず，統語解析で用いる文法と矛盾する場合がある．つまり，係り受け情報が示している係り受け関係が，文法規則に付与された制約条件に違反しているために，その関係を樹立することができない場合がある．

統計的に得られた係り受け関係は，もっともらしさに応じて確率が付与されている．パーザは確率の高い係り受け関係から順番に採用して，解析を行なっていき，活性弧と不活性弧の補完操作に失敗する場合には，前接する活性弧の係り先の次候補を採用して処理を行なっていく．

表 1: すべての文の実験結果

	A	B	B/A
弧の数	117	367	3.1
処理時間 (msec)	219	1847	8.4
解析結果の数	1.4	7.7	5.5

表 2: 形態素数 3 から 7 の文の実験結果

	A	B	B/A
弧の数	80	172	2.2
処理時間 (msec)	145	420	2.9
解析結果の数	1.1	2.9	2.6

つまり，係り受け情報でもっともらしいと判断された係り受け関係を文法規則を用いて成立させていき，文法規則がその係り受け関係を禁止している場合には，次にもっともらしい係り受け関係を採用して処理を行なう．係り受け情報にしたがって文法規則を適用していくことにより，無駄な弧を生成することなく解析を進めていくことができる．これはチャート法におけるアジェンダ制御に係り受け情報を利用することに相当する．

3 実験

前節で述べたチャート法の効率化のアルゴリズムを構文解析システム SAX [松本 94] 上に実装し，SAX 付属の日本語文法 (文法規則数約 150) を用いて，講談社和英辞典 [清水 79] の例文 100 文 (一文当たりの平均文字数 14.1，平均形態素数 7.5) の解析を行なった．形態素解析には茶筌 [松本 97] を使用した．

SAX 付属の日本語文法で解析できた 66 文[4]について，解析中に生成された弧の数，処理時間，解析結果の数を表にしたものが表 1 から表 4，解析中に生成された弧の数と一文の長さ (形態素数) の関係をグラフに表したものが図 4，解析時間と一文の長さ (形態素数) の関係をグラフにしたものが図 5 である．図 4 と図 5 では，一文の長さ (形態素数) ご

[4]解析失敗のうち形態素解析に起因するもの 12 文，統語解析に起因するものが 22 文であった．

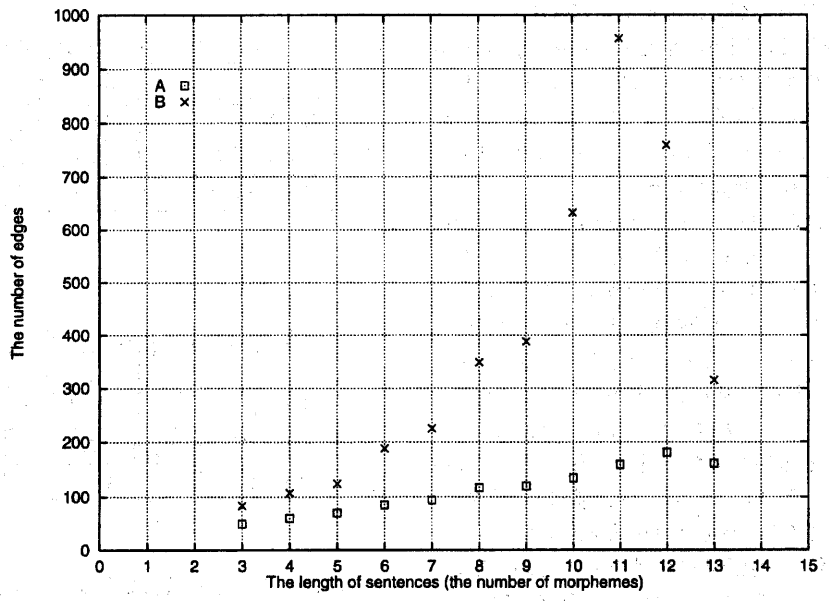

図 4: 一文の長さ (形態素数) と生成された弧の数の関係

表 3: 形態素数 6 から 10 の文の実験結果

	A	B	B/A
弧の数	104	300	2.9
処理時間 (msec)	194	1040	5.4
解析結果の数	1.1	5.5	5.0

表 4: 形態素数 9 から 13 の文の実験結果

	A	B	B/A
弧の数	143	642	4.5
処理時間 (msec)	277	3310	11.9
解析結果の数	1.1	12.5	11.4

とに平均をとっている．表中の A が係り受け情報を利用して効率化した場合で，B が係り受け情報を利用しなかった場合である．表 1 はすべての結果 (66 文) を集計したもの，表 2 は入力文の形態素数が 3 から 7 の文 (38 文) を解析した結果を集計したもの，表 3 は入力文の形態素数が 6 から 10 の文 (47 文) を解析した結果を集計したもの，表 4 は入力文の形態素数が 9 から 13 の文 (14 文) を解析した結果を集計したものである．なお，本実験では係り受け情報によって，最大どの程度効率が向上するかを調べるために，係り受け情報の正解データを人手で作成した．

SAX 付属の日本語文法は，制約条件を付加していない単純な文脈自由文法で記述されているために，比較的短い文に対してもかなりの曖昧性を示している．しかし，係り受け情報を利用することにより，弧の数は 32%，処理速度は 8.4 倍になる (表 1)．比較的短い文に対しては，弧の数が 45%，処理速度は 2.9 倍であるのに対して (表 2)，比較的長い文に対しては，弧の数が 22%，処理速度は 11.9 倍である (表 4)．これから，一文の長さが長くなればなるほど，係り受け情報を利用することによるパーザの効率化の度合が大きくなっているのがわかる．

4 ロバスト解析への応用

本稿で示した手法では，係り受け情報を利用して統語解析を誘導していくわけであるが，係り受け情

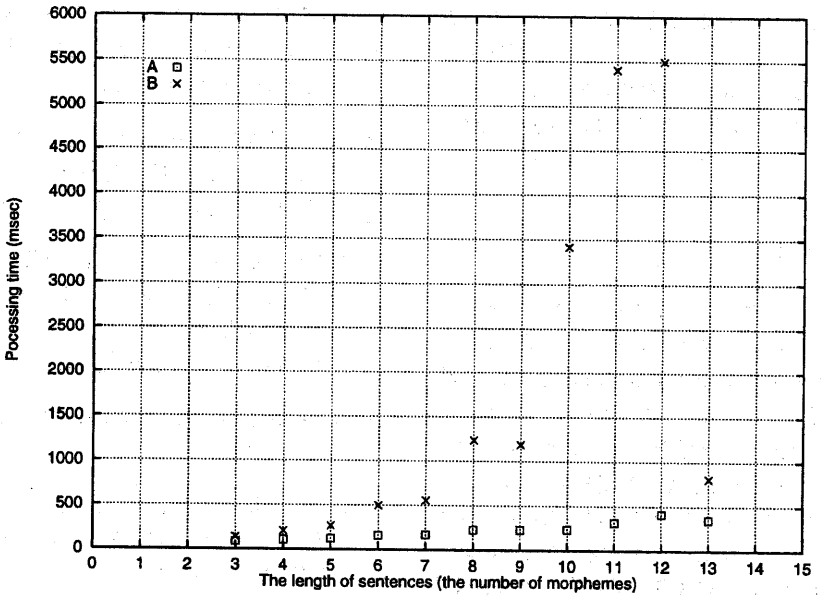

図 5: 一文の長さ (形態素数) と解析時間の関係

報が示唆する係り受け関係を文法規則が認可しない場合，その係り受け関係を放棄して次候補を選択して処理を行なっている．しかし，入力文中に何らかの不適格性が含まれている場合には，文法規則が認可しないからといって，その係り受け関係を放棄してしまうよりは，その係り受け関係の優先度を下げ，後に必要となった時に利用できるようにしておいた方が処理の頑健性の観点からも望ましい．

[今一 97] では，文法的不適格文処理のための統合的枠組みを提案している．そこでは，入力文の解析途中で生成される部分構造にコストと報酬という二つの指標を与えている．コストは，その部分構造がどの程度不適格性を含んでいるかを示すものであり，報酬は，その部分構造が解析失敗の回復にどの程度貢献するかを示すものである．[今一 97] では，文法記述の枠組みとして単一化文法 [Shieber86] を採用している．単一化文法では，文法に課せられた制約条件の充足/違反を単一化演算の成功/失敗と捉えている．したがって，制約違反の度合であるコストを計算するためには，単一化演算の拡張を行なえばよい．拡張した単一化演算をコスト付き単一化と呼ぶ．入力文の解析が失敗した場合に，コストと報酬の二つの指標を考慮することによって失敗の回復に有効な部分構造を選択し，処理を行なっている．チャート法に照らして考えてみると，各弧に付与されたコストと報酬を基にアジェンダ制御を行なっていることになる．

係り受け情報は，この枠組みにおける報酬と捉えることができる．係り受け情報から得られる報酬をうまく利用するためには，報酬とコストを組み合わせた値を考え，報酬が高くてもコストが大きい場合には，その値を下げるようにすればよい．具体的には，係り受け確率 p を以下の式でコスト w_dに変換する．

$$w_d = -log(p)$$

制約違反の度合を考慮に入れたコスト w_cは，コスト付き単一化によって定義される．したがって，部分構造の評価に使われるコスト Wは，

$$W = \alpha w_c + (1-\alpha) w_d \ \ (0 \leq \alpha \leq 1)$$

と定義することができる．αは二つのコストの統合比率である．そうすると，解析の初期状態では，係り受け情報から得られた係り受け確率を変換したものが Wとなっている．統語解析が進むにしたがって，個々の係り受け関係が文法規則によってテストされ，何らかの不適格性が発見された場合にコスト

が計算されて，Wの値が変化していく．Wの値を適切に見積もることができれば，Wの値だけをガイドラインとして統語解析を制御していくことができ，適格文に対する処理と不適格文に対する処理を区別せずに統一的に行なうことができる．

5 おわりに

本稿では，統語的曖昧性によるパーザの解析効率の低下を防ぐために，統計的に得られた係り受け情報を用いてパーザを効率化する方法を述べた．3節の実験では係り受け情報として正解データを利用したが，実際には統計的学習を行なった係り受け解析[藤尾 97]の結果を利用する．統計的に得られた係り受け情報は必ずしも正しいとは限らないが，確率によって順序付けられた係り受け関係を確率の高い順に利用していくことにより，係り受け情報を利用しない場合に比べて，効率的に処理していくことができる．

4節では，係り受け情報をロバスト解析に応用する方法について述べた．そこでも述べたように文法記述として単一化文法の枠組みを用いている．我々は現在，HPSG (Head-Driven Phrase Structure Grammar) [Pollard94] に基づく日本語文法を作成しており，今後は，その日本語文法を用いて，係り受け情報と制約違反の度合のバランスを考慮に入れた効率の良いロバスト解析を実現していく予定である．

参考文献

[Charniak93] Charniak, E.: *Statistical Language Learning*, The MIT Press (1993).

[Kay80] Kay, M.: Algorithm Schemata and Data Structure in Syntactic Processing, Technical Report CSLI-80-12, Xerox PARC (1980).

[Pollard94] Pollard, C. and Sag, I. A.: *Head-Driven Phrase Structure Grammar*, The University of Chicago Press (1994).

[Shieber86] Shieber, S. M.: *An Introduction to Unification-Based Approaches to Grammar*, No. 4 in CSLI Lecture Notes, CSLI (1986).

[今一 97] 今一修, 松本裕治：文法的不適格文処理のための統合的枠組み, 人工知能学会誌, Vol. 12, No. 3, pp. 404–411 (1997).

[松本 94] 松本裕治, 伝康晴, 宇津呂武仁：構文解析システム SAX 使用説明書 version 2.1, 奈良先端科学技術大学院大学 (1994).

[松本 97] 松本裕治, 北内啓, 山下達雄, 今一修, 今村友明：日本語形態素解析システム『茶筌』version 1.0 使用説明書, Technical Report NAIST-IS-TR97007, 奈良先端科学技術大学院大学 (1997).

[清水 79] 清水護, 成田成寿：和英辞典, 講談社学術文庫 (1979).

[藤尾 97] 藤尾正和, 松本裕治：統計的手法を用いた係り受け解析, 情報処理学会研究報告自然言語処理 117-12 (1997).