大規模日本語文法の開発

野呂　智哉[†]　橋本　泰一[†]
徳永　健伸[†]　田中　穂積[†]

構文解析において，多様な言語現象を扱うためには大規模な文法が必要となるが，一般に人手で文法を開発することは困難である．一方，大規模な構文構造付きコーパスから様々な統計情報を取り出し，自然言語処理に利用する研究が多くの成果をあげてきており，構文構造付きコーパスの整備が進んでいる．このコーパスから大規模な文脈自由文法(CFG，以下，文法と略す)を抽出することが考えられる．ところが，コーパスから抽出した文法をそのまま用いた構文解析では多数の解析結果(曖昧性)を作り出すことが避けられないことが問題であり，それが解析精度の悪化や解析時間，使用メモリ量の増大の要因ともなる．効率的な構文解析を行うためには，曖昧性を増大させる要因を分析し，構文解析の段階では曖昧性を極力抑えるよう文法やコーパスを変更する必要がある．本論文では，構文解析で出力される曖昧性を極力抑えた文法を開発するための具体的な方針を提案し，その有効性を実験により明らかにしている．

キーワード:　大規模日本語文法，構文構造付きコーパス，構文解析

Building a Large-Scale Japanese Grammar

Tomoya Noro [†] , Taiichi Hashimoto [†] , Takenobu Tokunaga [†]
and Hozumi Tanaka [†]

Although large-scale grammars are prerequisite for parsing a great variety of sentences, it is difficult to build such grammars by hand. Yet, it is possible to derive a context-free grammar (CFG) automatically from an existing large-scale, syntactically annotated corpus. While seemingly a simple task, CFGs derived in such fashion have seldom been applied to existing systems. This is probably due to a great number of possible parse results (i.e. high ambiguity). In this paper, we analyze some causes of high ambiguity, and we propose a policy for building a large-scale Japanese CFG for syntactic parsing, capable of decreasing ambiguity. We also provide an experimental evaluation of the obtained CFG showing reduction in the number of parse results (reduced ambiguity) created by the CFG and the improved parsing accuracy.

KeyWords: *Large-Scale Japanese CFG, Syntactically Annotated Corpus, Syntactic Analysis*

1　はじめに

近年，情報化が進むにつれて，大量の電子テキストが流通するようになった．これを有効活用するために，情報検索や情報抽出，機械翻訳など，計算機で自然言語を処理する技術の重要

† 東京工業大学 大学院情報理工学研究科, Graduate School of Information Science and Engineering, Tokyo Institute of Technology

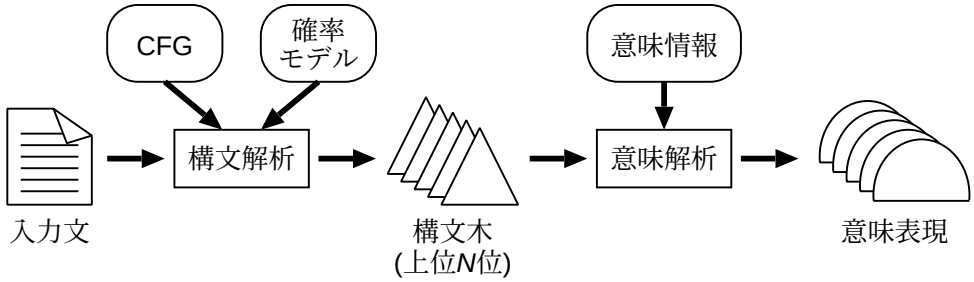


図 1　自然言語解析の流れ

性が増している．この自然言語処理技術は様々な知識を必要とするが，その中で構文解析の際に最もよく用いられるものは文脈自由文法(CFG，以下，文法と略す)である．ところが，人手で作成した大規模な文法は，作成者の想定する言語現象にどうしても“もれ”があるため，網羅性に欠けるという問題がある．一方，最近では，コーパスから抽出した統計情報を用いて自然言語を解析するコーパスベースの研究が成果をあげており，それに伴い，(電子)コーパスの整備が進んでいる．このコーパスから文法を自動的に抽出する研究もあり(Charniak 1996; 白井, 徳永, 田中 1997)，文法作成者に大きな負担をかけることなく，コーパス内に出現する多様な言語現象を扱える大規模な文法を作成することが可能である．しかし，コーパスから抽出した文法には問題がある．それは，コーパスから抽出した文法で構文解析を行うと，一般に，膨大な量の構文解析結果(曖昧性)[1]が出力されることである．その要因については後述するが，これが，解析精度の低下や解析時間，使用メモリ量の増大の要因となる．コーパスから抽出した大規模文法がこれまで実用に供されなかった最大の理由はここにある．

コーパスには意味を考慮した構文構造が付与されていることが普通であり，そのコーパスから抽出した文法で構文解析を行うと，意味解釈に応じた異なる構文解析結果が多数生成される．しかし，意味情報を用いない構文解析の段階では，意味的に妥当な少数の構文構造に絞り込めず，可能な構文構造を全て列挙せざるを得ない．我々は，構文解析結果(構文木)に沿って意味解析を進める構文主導意味解析(Syntax Directed Semantic Analysis, SDSA) (Jurafsky and Martin 2000)を想定し，構文解析の段階で生じる曖昧性を極力抑え，次の意味解析の段階で意味的に妥当な意味構造を抽出するという2段階の解析手法を採用する(図1).

本論文では，構文解析の段階の曖昧性を極力抑え，その後の意味解析の段階にも有効な構文構造を生成する大規模日本語文法について検討する．その結果，検討前の文法と比較して，出力される解析木の数を10^{12}オーダから10^{5}オーダまで大幅に減少させることが可能になった．さらに，この文法から得た解析結果に対して，意味情報をまったく用いず，確率一般化LRモデル(PGLRモデル) (Inui, Sornlertlamvanich, Tanaka, and Tokunaga 1998)によるスコア付け1

1 以降，特に断わらない限り，構文解析結果の曖昧性を単に曖昧性と呼ぶ.

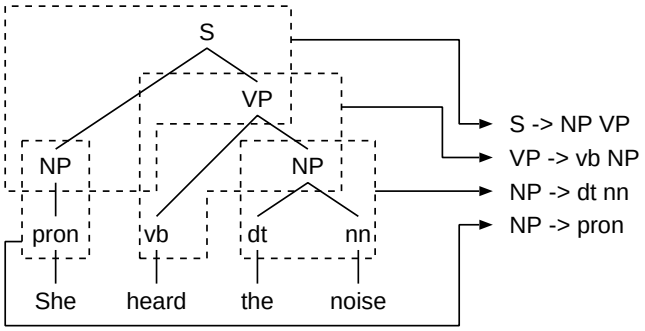


図 2　Penn Treebank コーパスからの文法抽出

位の解析木の文の正解率は約60%であった．一方，スコア付け1位の解析木に対し，機械的な方法で文節の係り受けの精度を測定したところ，意味情報を用いなくても，89.6%という高い係り受け精度が得られた．意味情報を本格的に利用することで，さらなる精度向上が図れるという見通しを得ている．

以下に本論文の構成を述べる．第2節では，コーパスから文法を抽出する主な研究を二つ紹介する．第3節では，我々が大規模日本語文法を作成する際の手順について述べる．第4節では，コーパスから抽出した文法が，構文解析において膨大な量の曖昧性を出力する要因を考察する．第5節では，構文解析結果の曖昧性の削減を考慮した具体的な文法とコーパスの変更方針を述べ，第6節，第7節では，変更したコーパスから抽出した文法の有効性を実験により明らかにする．最後に，第8節で本研究を総括し今後の課題を述べる．

2　関連研究とその問題点

本節では，文法をコーパスから抽出する主な類似研究を紹介する．

英語の大規模な構文構造付きコーパスとしてPenn Treebank コーパスがある (Marcus, Santorini, and Marcinkiewicz 1993)．Charniak はこのコーパスから “tree-bank grammar” と呼ばれるCFG を抽出し，人手で作成した文法との比較を行っている (Charniak 1996)．tree-bank grammar は，各中間ノードについて，そのラベルを左辺に，子ノードのラベルを右辺に持つCFG 規則を獲得することで抽出できる (図2)．これまで，コーパスから抽出した文法では，構文解析はうまくいかないと言われていたが，人手で作成した文法との比較実験の結果，特に単語数の多い長い文では，コーパスから自動抽出した文法の解析精度が良くなることを示し，それまでの一般的な見識が誤りであることを明らかにしている．

一方，日本語では，Penn Treebank コーパスのような大規模な構文構造付きコーパスが存在しない．大規模なコーパスとしてEDR コーパス (日本電子化辞書研究所 2001) と京大コーパス (黒橋 長尾 1997) がある．しかし，EDR コーパスは括弧付きコーパスであり，付与されている

構文木の内部ノードにラベルが付いていない．京大コーパスは，二つの文節間の依存関係が付与されているだけで，文節内の構造は付与されていないので，tree-bank grammar のような CFG は抽出できない．白井らは EDR コーパスからの CFG の自動抽出を試みている (白井他 1997)．構文木の内部ノードにラベルが付与されていないので，各内部ノードに対して適当なラベル (非終端記号) を付与する方法を提案している．

しかし，日本語，英語いずれの場合にも，構文構造付きコーパスから抽出した大規模な CFG で構文解析を行うと，膨大な数の構文解析結果が出力される．この問題に対し，Charniak は，コーパス中の出現頻度の低い文法規則を削除し，確率文脈自由文法 (PCFG) で得られる生成確率に基づく最良優先解析 (best-first parsing) を行い，解析途中で曖昧性を抑えている．これは，出現頻度の低い文法規則は構文解析における曖昧性を増大させるだけで，解析精度にほとんど影響を与えないという仮定に基づいている．しかし，詳細は後述するが，出現頻度の低い文法規則だけが構文解析結果の曖昧性を増大させるわけではない．労力は要するが，構文解析における曖昧性を増大させる要因を人手で分析する必要があると我々は考えている．

白井らは，構文解析結果の曖昧性を増大させる要因を分析し，多数の曖昧性を作り出す文法規則を機械的に変更することで，曖昧性の削減を図っている．しかし，機械的な変更だけで曖昧性を削減することには限界があり，人手による変更も必要になる．人手による変更が必要となる例を以下に挙げる．

機能による助詞の細分化:　白井らは，助詞を形態素ごとに細分化することで曖昧性を抑えている．しかし，格助詞，終助詞，並列助詞など機能による細分化も曖昧性の削減には必要である．EDR コーパス中の助詞に付与されている品詞はすべて “助詞” であり，機能による細分化は人手を要する．

意図しない非終端記号の割り当て:　白井らは，括弧付きコーパスである EDR コーパスから CFG を抽出するために，内部ノードに付与するラベルを機械的に推定している．しかし，機械的な推定では，アルゴリズムで想定していない文法規則を生成することがある．例えば,「変化/し/まし/た/か」という単語列をカバーするノードのラベルを考えると (スラッシュは単語区切りを示す)，白井らのアルゴリズムでは，右端の「か」が助詞であるため，“後置詞句” となり，次の CFG 規則が得られる．

後置詞句 → 動詞 語尾 助動詞 助動詞 助詞

直観的には，後置詞句ではなく動詞句の方が適切であるが，機械的な推定では，意図しない非終端記号の割り当てを細かく除外していくことは困難である．

後者は曖昧性の増減と直接は関係のないことである．しかし，人間が見て妥当な CFG を作成するためには，機械的に内部ノードのラベルを推定するのではなく，(Penn Treebank コーパスのような) 構文構造付きコーパスを用意し，そこから文法を抽出すべきであると考えている．

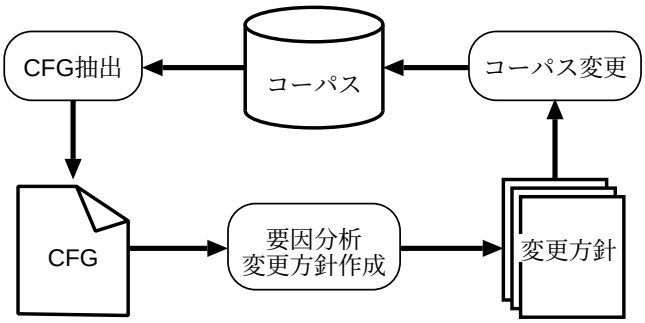


図 3　大規模日本語文法作成手順

3　大規模日本語文法の作成手順

我々は，既存の構文構造付きコーパスを出発点とし，以下の手順で文法を作成することを試みている (図3).

(1)　既存の構文構造付きコーパスから文法を抽出する
(2)　構文解析結果の曖昧性を増大させる要因を分析する
(3)　分析結果をもとに構文構造付きコーパスの変更方針を作成する
(4)　変更方針に従ってコーパスを変更し，そこから新しい文法を再抽出する
(5)　(2)〜(4) を繰り返す

文法の抽出は，Charniak による tree-bank grammar の抽出方法 (Charniak 1996) と同様の方法をとる．出発点として使用した構文構造付きコーパスの概要については，付録 A節で述べる．

上述の文法作成手順では，変更対象が構文構造付きコーパスであり，文法はコーパスから抽出されるだけであるため,「文法の作成」という表現に違和感を感じるかもしれない．しかし，既存のコーパスから抽出した文法は，コーパス作成者の意図に反し，きわめて多数の構文解析結果を出力する．そのため，コーパスの作成は，そこから抽出した文法による構文解析結果を考慮しながら行うことが望ましい．換言すれば，文法の作成，変更とコーパスの作成，変更は並行して進める必要があると考えている．このようにして作成したコーパスは，PCFG モデル等の確率モデルによる学習の際に，訓練データとしても利用できる．

4　構文解析結果の曖昧性を増大させる要因

繰り返し述べたように，大規模な構文構造付きコーパスから抽出した CFG をそのまま利用して構文解析を行うと，多数の曖昧性を生じる．曖昧性が増大すると，解析に必要な時間，メモリ量が増大するだけでなく，その中から構文的に正しいものを選択することが困難になる．この問題を解決するためには，曖昧性を増大させる要因を分析しなければならない．

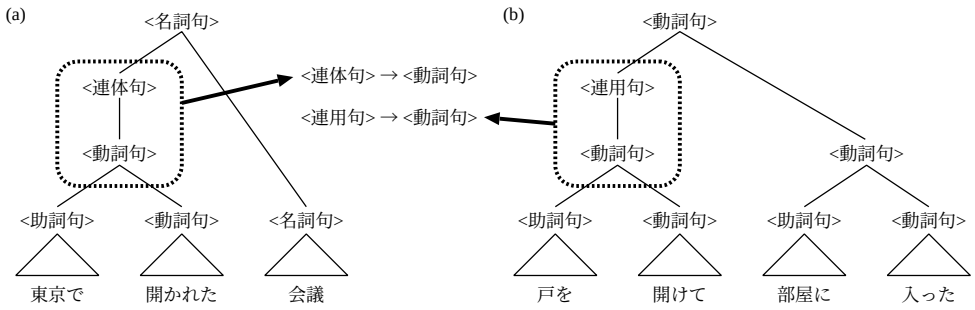


図 4　CFG 抽出時における構文情報の欠落

曖昧性を増大させる要因は，以下の 4 種類に大別できる．

ラベル付けの誤り (要因 1):　構文構造は人手で付与するため，誤りは避けられない．誤った構造が付与されたコーパスから抽出した文法は，誤った構造を生成し，それが無意味な曖昧性の増大につながる．

構文構造の不一致 (要因 2):　大規模なコーパスを作成する際，作業は一人ではなく複数で行うことが一般的である．この時，作業者による構造の付け方の “ゆれ” が問題となる．一貫性のない構文構造付きコーパスから抽出した文法は冗長な文法規則を持ち，それが無意味な曖昧性の増大につながる．

構文情報の欠落 (要因 3):　構文構造を付与する際，コーパス作成者は文全体の構造を考慮しながら，部分的な構造を決定することが多い．ところが，CFG の各規則はノードの親子関係に関する情報しか持たず，それ以外の周辺文脈情報 (各子ノードを根とする部分木の情報や，親ノードを根とする部分木の外側の構文情報) を持たない．構文構造の曖昧性を解消する上で有用な構文情報が欠落することで，構文解析において，構文的に誤った解析木を余分に生成することがある．例えば，図 4に示す 2 つの構文木が存在した場合，構文木 (a) からは “＜連体句＞→＜動詞句＞” という規則が，構文木 (b) からは “＜連用句＞→＜動詞句＞” という規則が抽出される．しかし，これらの規則には動詞の活用形に関する情報が欠落しているため，活用形に関係なく，すべての動詞句が連体句にも連用句にもなれてしまう．その結果，連用形の動詞句が連体句として体言を修飾したり，終止・連体形の動詞句が連用句として用言を修飾したりする解析木が生成でき，これが曖昧性を不必要に増大させる要因となる．

意味情報の必要性 (要因 4):　曖昧性の中には，その解消において，構文情報だけではなく意味情報も必要とするものがある．例えば,「彼の目の色」の「彼の」が「目」と「色」のどちらを修飾するかの曖昧性の解消には，各語の意味を考慮する必要があり，構文情報だけでは解消できない．詳細は後述するが，我々が想定する自然言語解析では，構文解析時は構文情報のみを利用し，意味情報を必要とする曖昧性の解消は，構文解析後の意

味解析で行うこととしている．構文解析時に解消できない曖昧性を列挙することは，構文解析結果を組み合わせ的に増大させることになる[2]．

要因1と2は，コーパスの誤りであるため，訂正すべきもとのして，以下の考察から除外する．一方，要因3と4はコーパスの誤りではない．要因3の解決には，どの構文情報が必要であるかを考察し，その情報を非終端記号に追加し，細分化する．要因4の解決には，意味情報を利用しない限り解決が困難な曖昧性を包含した単一の構文構造をコーパスに付与し，CFGを再抽出する．すなわち，再抽出したCFGによる構文解析結果では，要因4による曖昧性を区別しない．こうすることで，構文解析結果の曖昧性を抑えられるだけでなく，意味解析で解消すべき曖昧性の所在が明らかになる．次節では，我々の具体的な変更方針について述べる．

5　文法，コーパスの変更方針

要因3の曖昧性はすべて除外することが理想である．EisnerやKomagataは，Categorial Combinatory Grammar(CCG)について，解析器側を変更することによってこの曖昧性を完全に除外し，一つの意味解釈に対して一つの解析木を出力する(exactly one syntactic structure per semantic reading)手法を提案している(Eisner 1996; Komagata 1997)．本研究ではCFGを使用し，解析器に変更を加えるのではなく，文法とコーパスそのものを変更しながら，この曖昧性を抑える．

さらに，我々は，要因4の曖昧性を包含した単一の構造を生成する(意味的情報は利用しないことを前提とした)，構文解析のための大規模日本語文法の構築を目的としている．しかし，この方針によって，出力される構文解析木の数を抑えることは，その後の意味解析を困難にすることもあり得る．そのため，構文解析時には包含された曖昧性を意味解析で解消することを念頭に置きながら，要因4の曖昧性のうち，どれを単一の構文構造で表現し，構文解析結果の曖昧性を抑えるかを詳細に検討する必要がある．

我々が使用しているコーパスには，以下のような不備や欠点があった．

(1)　用言の活用形に関する情報の欠落(要因3)
(2)　複合名詞内の構造の曖昧性(要因4)
(3)　連用修飾句，連体修飾句の係り先の曖昧性(要因4)
(4)　並列構造の曖昧性(要因4)

これらについて，具体的に変更方針を述べる[3]．

2 英語においても，PP attachment問題を構文情報だけで解決することはできない．この曖昧性は，前置詞句の数に対するCatalan数のオーダで増大し(Martin, Church, and Patil 1987; Church and Patil 1982)，文全体の構文解析結果の曖昧性の増大の最大の要因の1つとなる．

3 我々は，第1節で述べた文法開発の手順のサイクルを複数回に分けてこれらの検討を行った(野呂, 橋本, 徳永, 田中 2003)．

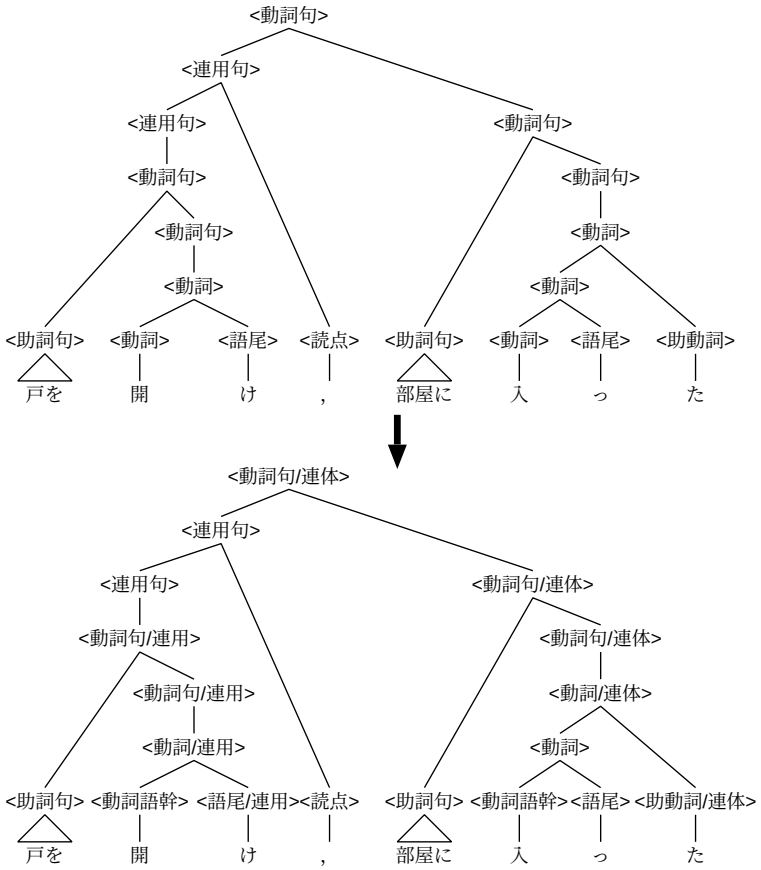


図 5　活用形に関する情報の付与

5.1　用言の活用形に関する情報の欠落

用言の活用形の情報が欠落しているためにそれが連体修飾句になるか連用修飾句になるかで曖昧になることを，第4節で，要因3の曖昧性の例として挙げた．実際，我々が使用しているコーパスで，この問題があった．これを解決するために，用言等の語尾や助動詞の活用形に関する情報を構文構造に引き継ぐように変更する (図5)．ただし，未然形，連用形等すべての活用形を付与するのではなく，その語が末尾に現れることで連用修飾句，または連体修飾句になり得る場合にのみ，それぞれ「連用」,「連体」というラベルを追加する．これは，活用形の情報を付与する目的が，その用言が連用修飾句になり得るものか，連体修飾句になり得るものかを区別するためであり，それ以外の情報は必要ないからである．

5.2　複合名詞内の構造の曖昧性

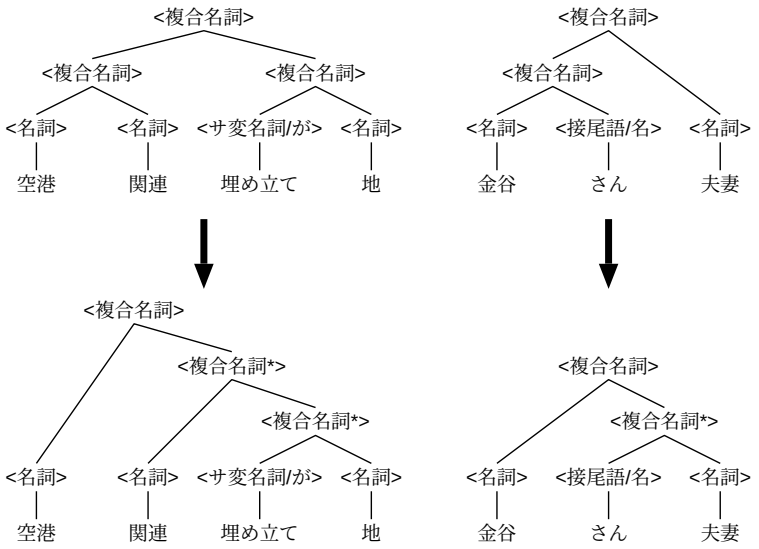


図 6　複合名詞の構造の変更

複合名詞内の構造の曖昧性を構文解析で解消することは困難であり，この曖昧性を構文解析結果の違いとして出力すべきではないと考えている．白井らも，この曖昧性を構文解析結果の違いとして出力しないよう文法を変更している (白井他 1997) [4]．我々もその方針に倣い，複合名詞については，語構成に関係なく右下がりの構造に統一する (図6) [5]．

5.3　連用修飾句，連体修飾句の係り先の曖昧性

次に，連用修飾句，連体修飾句の係り先の曖昧性の扱いを検討する．我々は，連用修飾関係の曖昧性は従来通り別の構造として区別し (すなわち，構造は変更しない) [6]，連体修飾関係を表す構造を，複合名詞の場合と同様，意味に関係なく同一の構造にする (図8) [7]．つまり，連用修飾関係の曖昧性は構文解析結果の曖昧性として残し，連体修飾関係の曖昧性は構文解析の段階では出さず，後の意味解析でこの曖昧性を解消することになる．

上述の方針に決定した理由は二つある．一つは，連用修飾関係を表す構造を意味に関係なく

4 白井らは，複合名詞という言葉ではなく,「同一品詞列を支配するノード」という表現を使用している．我々は，名詞，接頭語，接尾語などで構成され，名詞として働く構成素を対象とし，複合名詞と呼んでいる．

5 構造を右下がりにする際，複合名詞の根ノードと内部のノードのラベルを図6のように区別している．もし，これらを同一のラベルにすると,「金谷さん夫妻」の例において,「さん夫妻」という接尾語が先頭に出現する複合名詞を認める文法規則になってしまう．

6 元のコーパスでは用言のとる表層格の情報を利用していたが，格の区別は意味情報を必要とし，構文解析時の曖昧性解消が困難な曖昧性を増大させる要因となる．そこで，図7に示すように，用言のとる表層格の情報は無視する (野呂他 2003)．

7 構造を変更する際，図8に示すように，右側の名詞句と左側の連体句の下の名詞句を区別している．もし，これらを同一のラベルにすると，抽出した文法規則は変更前後の両方の構造を生成することが可能になり，構造を制限することができなくなる．

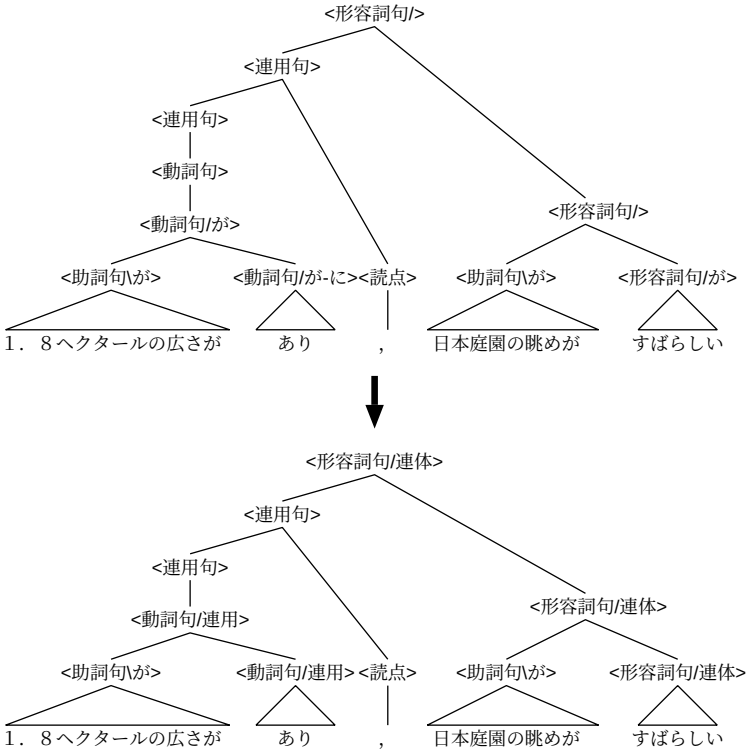


図 7　連用修飾句の係り先に関する変更

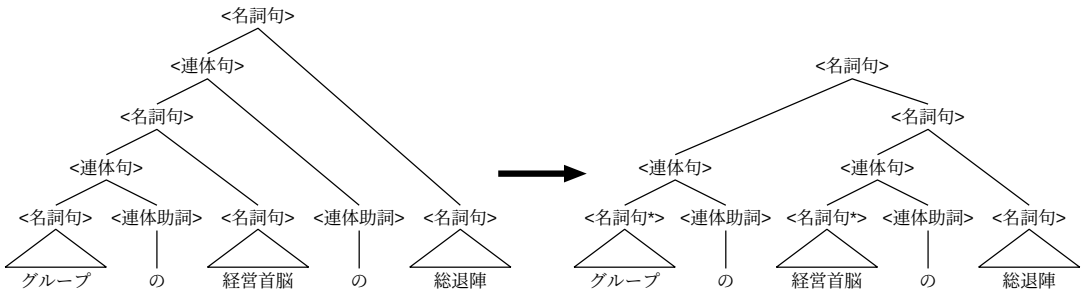


図 8　連体修飾句の係り先に関する変更

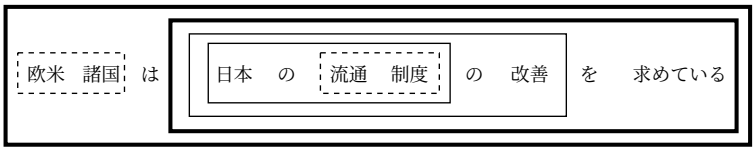


図 9　単文「欧米諸国は日本の流通制度の改善を求めている」の構造

同一の構造にすることは，構文解析後の意味解析を困難にすることになるからである．例えば，「欧米/諸国/は/日本/の/流通/制度/の/改善/を/求めている」という単文を考える．ただし，スラッシュは単語区切りを表す(「求めている」は動詞語幹，助動詞語幹，語尾に分割されるが，簡単のため，ここでは1語として表記する)．この文に対してボトムアップに(意味的に正しい)構文構造を付与すると，次の手順になる．

(1)　「欧米諸国」,「流通制度」のそれぞれを一つの複合名詞にまとめる(図9の破線で囲まれた部分)．

(2)　「日本の」と「流通制度」，そして「(日本の)流通制度の」と「改善」のそれぞれを一つの連体修飾関係にまとめる(図9の細い実線で囲まれた部分)．

(3)　「(日本の流通制度の)改善を」と「求めている」,「欧米諸国は」と「(日本の流通制度の改善を)求めている」の二つの連用修飾関係をまとめる(図9の太い実線で囲まれた部分)．

このように考えると，単文では，連用修飾関係を表すレベルが連体修飾を表すレベルより上にある．複文や重文は，この単文を組み合わせることで構成される．上位レベルである連用修飾関係を表す構造を意味に関係なく同一構造にすることは，複文や重文を構成する単文のまとまりを破壊することになり，文全体の構造がとらえにくくなる．その結果，構文解析後の意味解析が困難になる．下位レベルである連体修飾関係を表す構造を，意味に関係なく同一構造にし，連用修飾関係を表す構造は従来通り別の構造として区別することで，その後の意味解析を困難にせずに，構文解析の段階の曖昧性を抑えられると考えている．

もう一つの理由は，連用修飾句の係り先の曖昧性の解消は，連体修飾句の係り先の曖昧性の解消に比べて，構文解析での解決が容易であるからである．連用修飾句の係り先は，助詞と動詞の関係，副詞と助動詞の関係等を利用することで，決定できる可能性があるのに対し，連体修飾句の係り先は，連用修飾句の場合に比べて，品詞レベルでの解決が難しい．そこで，品詞レベルでの解決が比較的容易な連用修飾関係を表す構造は従来通りとし，連体修飾関係を表す構造は，意味に関係なく同一構造にすべきであると，我々は考えている．

ただし，連体修飾句の係り先の曖昧性が，大別して2種類あることに注意したい．

(1)　連用修飾句の範囲を変えないもの

(2)　連用修飾句の範囲を変えるもの

図10にそれぞれの例を示す．太い実線で囲まれた句は連用修飾句を，細い実線で囲まれた句は連体修飾句を，破線で囲まれた句は動詞を，網掛けの長方形で囲まれた句は連体修飾を受ける名詞を，矢印の始点は修飾関係の係り元を，終点は係り先を表す．

「新しい環境への適応能力を調べる」の場合，連体修飾句「新しい」が「環境」に係る場合でも「適応能力」に係る場合でも，動詞「調べる」に係る連用修飾句は「新しい環境への適応能力を」であることには変わりはない(図10(a)，(b))．ところが,「百年の歴史を持つ祭り」で

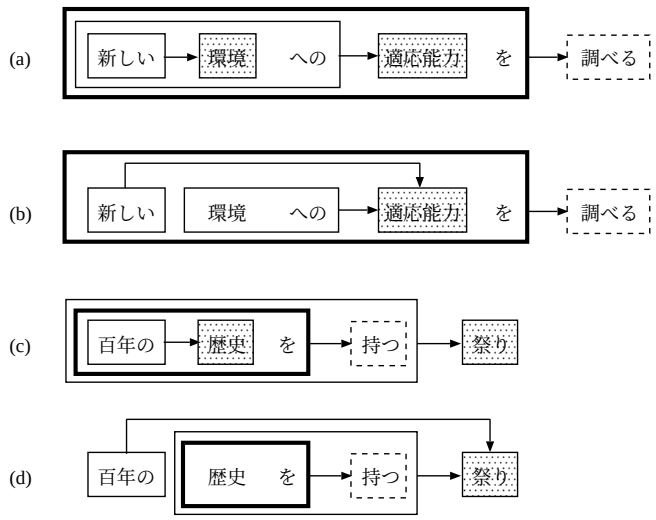


図 10　連体修飾句の係り先に関する2種類の曖昧性

は，連体修飾句「百年の」が「歴史」に係る場合の動詞「持つ」に係る連用修飾句は「百年の歴史を」であるのに対し,「百年の」が「祭り」に係る場合も考えられないこともない．後者の場合は,「歴史を」のみが「持つ」に係る連用修飾句となる (図10(c)，(d)).

我々は，連用修飾句の範囲と係り先は従来のまま変更せず，そこから抽出した文法は，その曖昧性を構文解析の段階に出力することにしている．その方針に合わせ，連用修飾句の範囲を変えない場合に限り，連体修飾関係を表す構造を同一の構造に統一する (すなわち,「新しい環境への適応能力を調べる」の場合は図10(b) の構造に変更し,「百年の歴史を持つ祭り」の場合は図10(c) の構造のままにしておく).

5.4　並列構造の曖昧性

並列構造の曖昧性の解消には意味的情報が必要であり，係り受け解析において並列構造を含む文の正解率は，含まない文に比べて低くなる．予備実験によると，並列構造を含む文の正解率は，含まない文の正解率の半分程度しかない (野呂他 2003) [8]．文の正解率を全体的に上げるためには，並列構造の曖昧性について検討する必要がある．KNP(黒橋 1998) では，先に並列関係にある部分を決定し，次にその内部の構造を分析するアプローチを採用している (黒橋 長尾 1992)．しかし，我々は，並列関係にあるかどうかの判定は構文解析に先立って行わず，その後の意味解析の段階で行うこととする．言い換えると，注目している二つの部分が並列関係にあるかどうかの曖昧性は，構文解析の段階では区別しない．

日本語には，並列名詞句，並列述語句，並列助詞句の3種類の並列構造がある[9]．我々は，こ

8 「文の正解率」の定義は第6節で述べる．
9 黒橋らは，それぞれを名詞並列，述語並列，部分並列と呼んでいる (黒橋 1998).

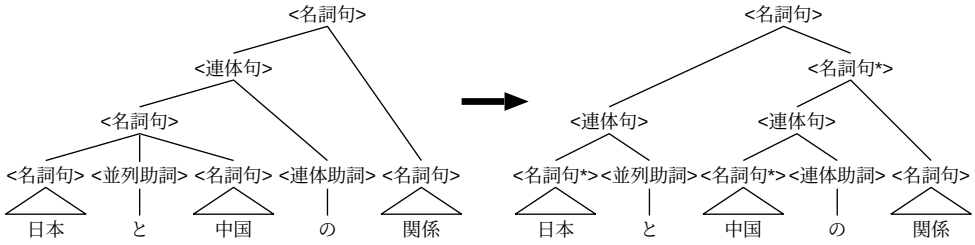


図 11　並列名詞句の構造に関する変更

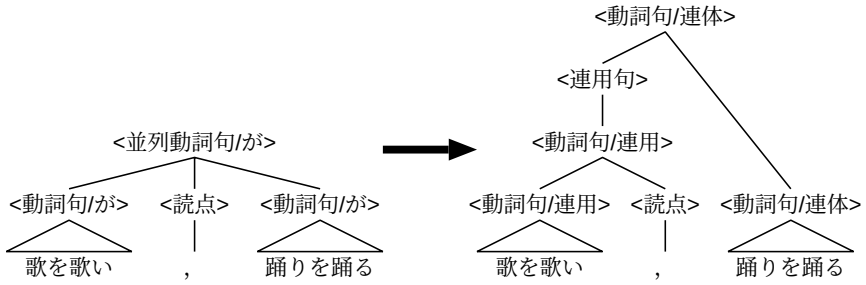


図 12　並列述語句の構造に関する変更

れらの構造を以下の方針で変更する.

並列名詞句:　名詞句「日本と中国の関係」において,「日本」と「中国」が並列関係にあるのか，それとも「日本」と「関係」が並列関係にあるのかという曖昧性の解消には，各語の意味的情報が必要となる.「AのBのC」,「AとBのC」の二つの名詞句を考えると，どちらの場合も名詞「A」,「B」,「C」の間の関係を分析することになる．このことから，並列名詞句の分析は連体修飾句の係り受けの解析に似ている．構文解析の段階では「Aと」を連体修飾句と同様に扱い，並列構造の曖昧性の解消は，次の意味解析の段階で，連体修飾関係の曖昧性の解消と同時に行うこととする (図11).

並列述語句:　予備実験 (野呂他 2003) によると，並列述語句を含む文の正解率は，それ以外の並列構造を含む文の正解率と比べて大幅に低くなる．これは，二つの述語句が並列関係にあるか否かの判断が，構文解析の段階では難しいためである．例えば,「歌を歌い，踊りを踊る」という文が並列構造を持つか否かは，並列関係の定義を明確にしなければ，コーパス作成者によっても判断が分かれるところである．構文解析の段階では並列述語句は連用修飾関係と同様に扱い，二つの述語句が並列関係にあるか否かの判断は，後の意味解析の段階で行うこととする (図12).

並列助詞句:　並列助詞句は,「国政段階でも個別産業レベルでも影響力は小さい」のように，並列関係にある二つの助詞句に含まれる助詞が同じであることが多いので，並列助詞句を含む文の正解率はそれほど低くならないと思われるかもしれない．ところが，予備実

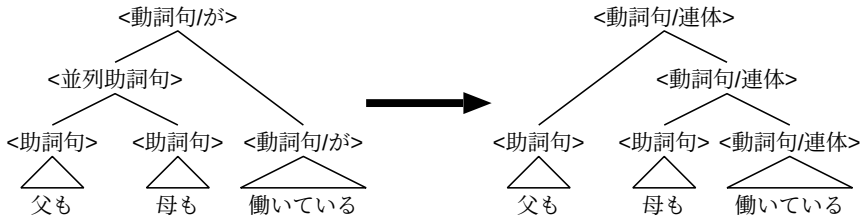


図 13　並列助詞句の構造に関する変更

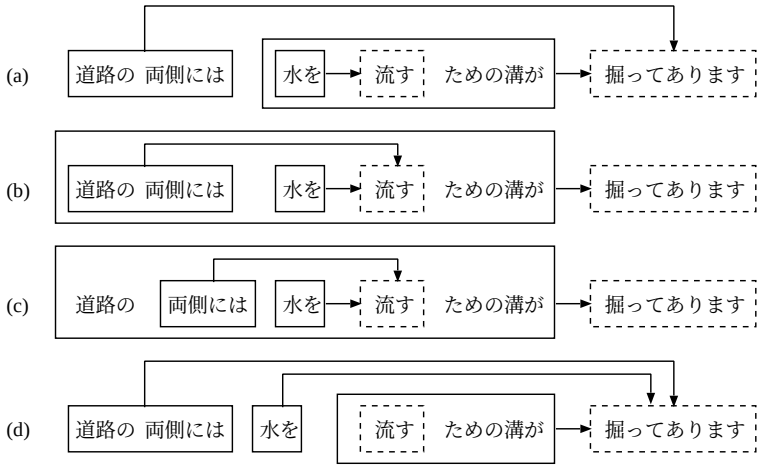


図 14　構文解析で生成される構造

験 (野呂他 2003) によると，並列助詞句を含む文の正解率は並列述語句を含む場合よりは高いが，並列名詞句を含む場合とほぼ同じであった．二つの助詞句が並列関係にあるか否かの判定には意味的情報が必要であり，構文解析の段階で解決することは困難である．構文解析の段階では，とりあえず並列関係にある助詞句は別個に動詞に係る構造を作ることとし，二つの助詞句が並列関係にあるか否かの判定は意味解析の段階で行うこととする (図13)．

以上をまとめると，我々の文法とコーパスの変更方針は以下のようになる．

(1)　複合名詞内の構造，連用修飾句の範囲を変えない連体修飾句の係り受け関係の構造は，語構成や意味に関係なく同一の構造にする．

(2)　連用修飾句の係り受け関係の構造，連用修飾句の範囲を変える連体修飾句の係り受け関係の構造は，従来通りの構造にする．ただし，用言のとる表層格の情報は無視する．

(3)　二つの句が並列関係にあるか否かの判定は構文解析の段階では行わず，並列関係にあるか否かで構造の区別はしない．

以上の方針に従って構築した文法を使用し,「道路の両側には水を流すための溝が掘ってあります」という文を構文解析すると，図14に示す4個の構文構造が生成される．ただし，実線で

囲まれた句は連用修飾句を，破線で囲まれた句は動詞を，矢印の始点は連用修飾関係の係り元を，終点は係り先を表す．これら4個の構文構造は，連用修飾句の範囲とその係り先の違いを表し，この中から一つの構文構造を選択することは，連用修飾句の範囲とその係り先を決定することを意味する．一方，連体修飾句の係り先は，各構文構造が持つ意味的曖昧性の中から一つの意味解釈を生成することによって決定する．例えば，構文構造(b)では連体修飾句「道路の両側には水を流す」が「ため」に係るか「溝」に係るかを決定し，構文構造(c)では連体修飾句「道路の」と「両側には水を流す」が，それぞれ「ため」に係るか「溝」に係るかを決定する．一方，構文構造(b)では,「道路の」が「ため」や「溝」に係る可能性は，動詞「流す」を連用修飾する「道路の両側には」という句の範囲を変えることになるので考慮する必要はない．

6　評価実験

前節で述べた方針によるコーパスへの構文構造の付与の有用性を確認するため，コーパスから抽出した文法を用いて，以下の2点について評価実験を行った．

(1)　構文解析結果の曖昧性がどの程度抑えられているか

(2)　どの程度の構文解析精度が得られるか

(1)の評価実験は本研究の目的そのものであるが，曖昧性が抑えられていても，解析精度が低ければ問題であるので，(2)の評価実験も必要である．

6.1　文法，コーパスの構文構造の変更

まず，付録A節で述べたコーパス8911文(平均20形態素)に対し，我々の方針に従って構文構造付きコーパス作成支援ツール(岡崎, 白井, 徳永, 田中 2001)で構文構造を変更した[10]．具体的には，以下の手順で変更を行っている．

(1)　我々の変更方針に従って文法を人手で変更.

(2)　MSLRパーザ(白井, 植木, 橋本, 徳永, 田中 2000)でコーパス中の文を構文解析し，構文解析結果の集合(統語圧縮共有森，packed-shared forest(Tomita 1986))を獲得.

(3)　コーパス作成支援ツールで，構文解析結果の集合を絞り込み，最終的に1つの正しい構文構造を選択.

手順(3)で使用するコーパス支援ツールは，解析結果を1つずつ表示させながら正しい構文構造を選択するためのものではなく，非終端記号名や特定の句の係り先を，正しい構文構造が満たすべき制約として，作業者が順々に与え，それを満たさない候補を排除しながら正しい構文構造を残すためのものである．制約は，構文構造が曖昧な箇所(制約の教示を必要とする非終端記号や係り受け)をマウスで選択し，表示される選択肢から正しい候補を選択することで与える．

10 変更前のコーパスは約2万文あるが，8911文しか変更していないので，それに対応する文だけを変更前のコーパスから抜き出し，実験に使用する．

表 1　変更前，変更後の文法による構文解析結果の数

	文法規則数	非終端記号数	終端記号数	構文解析結果数
変更前	1,694	249	600	1.868×10^{12}
変更後	1,949	279	600	9.355×10^{5}

作業は，100 文をラベル付けするのに約 3 時間かかり，4 人でこの約 9000 文をラベル付けするのに約 1 ヶ月かかった[11]．

6.2　構文解析結果の曖昧性の変化

変更前，変更後のコーパス全 8911 文からそれぞれ文法を抽出し[12](以降，変更前，変更後のコーパスから抽出した文法を，それぞれ「変更前の文法」,「変更後の文法」と呼ぶ)，MSLR パーザで構文解析を行った[13]．変更前，変更後の文法による構文解析結果の数を表 1 に示す．我々のコーパスの変更方針により，文法規則数は約 250 個増加しているが，構文解析結果の数は 10^{12} オーダから 10^{5} オーダに減少した[14]．

白井らの手法では，EDR コーパス約 188,000 文から抽出した文法で 1 文あたり 10^{9} オーダの解析木が出力される (白井他 1997)．文法抽出に使用した文の数に大きな差があるため公平な比較にはならないが，白井らの文法に比べて曖昧性が減少している主な要因として，以下の 3 つが考えられる．

連体修飾句と連用修飾句の区別:　白井らの文法では，連体修飾句か連用修飾句かを区別するためのラベルが付与されていない．これは，第 4 節で挙げた曖昧性を増大させる要因の 3 番目にあたる．この問題は第 4 節で挙げた用言の活用形の問題だけでなく，後置詞句でも同様に起こり得る．EDR コーパスでは,「が」,「を」等と「の」を区別せず，すべて「助詞」としているので，白井らの文法ではこれらの助詞が末尾に現れる句はすべて「後置詞句」となり，連体修飾句か連用修飾句かの区別が付かなくなる．我々の文法では,「東京へ行く」のように連用修飾句になる場合は「助詞句」,「東京の人口」のように連体修飾句になる場合は「連体句」となるので，このような曖昧性は出ない[15]．

品詞の細分化:　我々が使っているコーパスの品詞体系は EDR 日本語単語辞書に基づいて細分化されている．例えば，白井らは名詞を細分化していないが,「今日，東京へ行く」の「今日」のように助詞を伴わずに連用修飾可能な名詞を他の名詞と区別しておかなけれ

11 4 人のうち 3 人は変更方針の検討に直接は関わっておらず，ラベル付け前に我々(方針作成者) がその説明を行ったが，作業者が方針を理解し，本格的にラベル付けを開始できるまでに (作業期間の 1 ヶ月の他に)1，2 週間かかった．

12 (Charniak 1996) のようにコーパス出現頻度の低い文法規則を削除することはせず，全文法規則を利用する．

13 MSLR パーザは形態素解析と構文解析を同時に行うものであるが，品詞列を入力とすることで構文解析のみを行うことができる．今回は，品詞列を入力とし，形態素解析は終了しているものとしている．

14 今回の我々の方針では品詞レベルの変更は考慮しておらず，非終端記号の数の変化はない．

15 「鼻の長い象」のように,「の」が末尾に出現する句が連用修飾句になる場合もあるので，曖昧性は残る．しかし，逆に，現代語において，他の助詞が末尾に出現する句が連体修飾句になることはほとんどない.「駅を中心に発展する」のような例外もあるが，これは「中心に」の後に動詞「して」が省略されていると考えることができる．現段階では省略を CFG で扱うことを考えていないので，このような文は対象外とし，曖昧性を除外している．

表 2　文法 G_{train} の被覆率と再現率

	変更前	変更後
被覆率	98.51%	97.32%
再現率	96.63%	95.88%

ば，すべての名詞が助詞を伴わずに連用修飾することを認める文法規則となり，曖昧性を増大させる要因となる[16].

連体修飾関係と並列関係:　我々の文法では，連体修飾句の係り先の曖昧性と2つの句が並列関係にあるか否かの曖昧性を出さないようにしている．1文に含まれる連体修飾句や並列句の数はそれほど多くなく，先に挙げた2つの要因ほど，大きく曖昧性の削減に貢献していないが，構文解析での解決が困難な曖昧性を抑えることは，その後の意味解析においても重要なことである.

6.3　構文解析精度の変化

構文解析結果を確率一般化LR(PGLR)モデル(Inui et al. 1998)でランク付けし，解析精度を調べた[17]．ただし，8911文を10分割し，一つを評価用，残りをPGLRモデルの学習用とし，10分割交差検定で評価を行った．文法は全8911文から抽出したもの(G_{all})と，学習用データのみから抽出したもの(G_{train})の2通りを用意した．図15，図16に，上位1位から100位以内の解析結果についての文の正解率を示す．ただし，文の正解率は以下のように定義される.

$$\text{文の正解率} = \frac{\text{出力した解析木の集合の中に正しい木が含まれる文の数}}{\text{解析した文の総数}}$$

ここで,「正しい木」とは，コーパスの構文構造と完全に一致する解析木を指す．また，文法 G_{train} の被覆率，再現率を表2に示す．ただし，被覆率，再現率は以下のように定義する.

$$\text{被覆率} = \frac{\text{解析した文の総数} - \text{解析に失敗した文の総数}}{\text{解析した文の総数}}$$

$$\text{再現率} = \frac{\text{出力されたすべての解析木の中に正しい木が存在する文数}}{\text{解析した文の総数}}$$

従来の研究では，評価尺度として括弧付けの再現率や適合率など部分的な構造の正しさを示すものを使用することが多い．しかし，我々は，構文解析結果の集合から尤もらしい解析結果をいくつか選択し，それらに対して意味解析を行うことを前提としているので，構文解析の段階では，意図した構文構造と完全に一致していることが望ましい．構文構造の部分的な正しさを示す括弧付けの再現率や適合率よりも，上述の文の正解率の方が条件が厳しいが，重要な尺度であると考えている.

16 日本語では助詞が頻繁に省略され，一般的な名詞であっても助詞を伴わずに連用修飾する例もあるが，助詞が省略されていると判断できるものは対象外としている.

17 (Charniak 1996; 白井他 1997) はPCFGを用いているが，我々はPGLRモデルを使用する．我々が行った予備実験によると，PGLRモデルの方がPCFGよりも解析精度が高くなる.

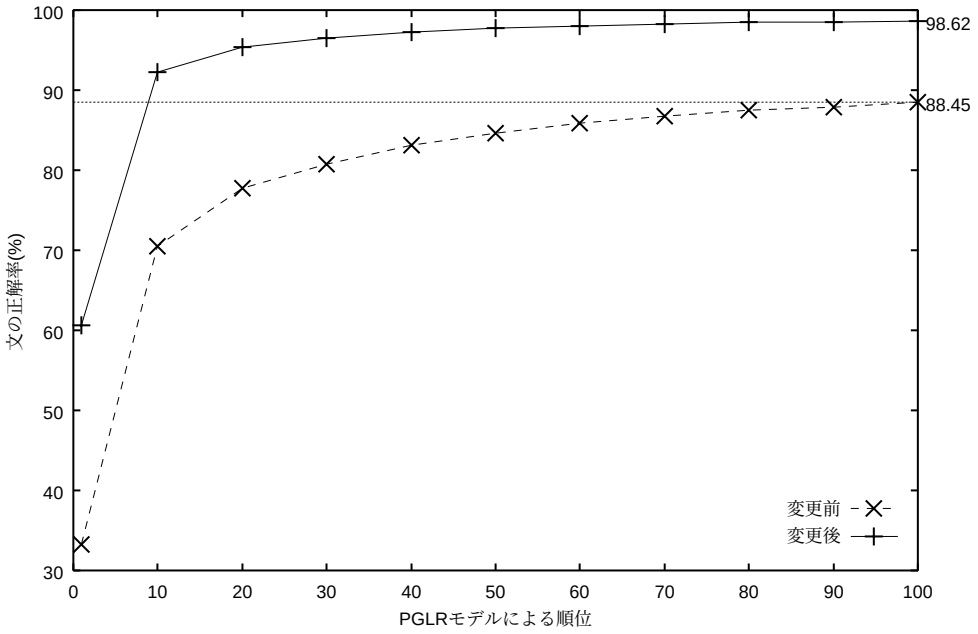


図 15　変更前と変更後の文法G_{all}による構文解析結果の文の正解率

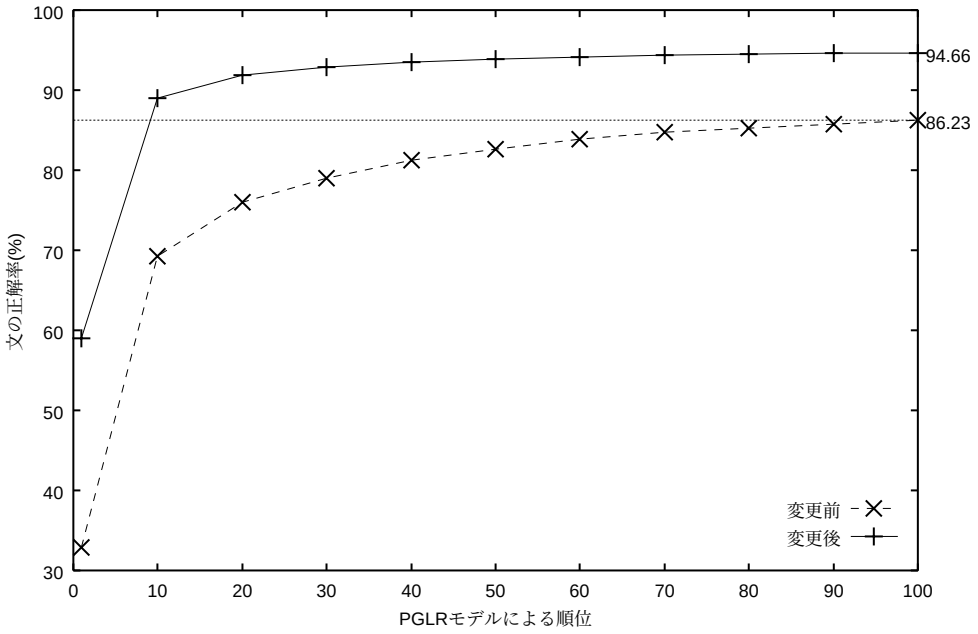


図 16　変更前と変更後の文法G_{train}による構文解析結果の文の正解率

PGLR モデルによる生成確率の上位100位以内の解析結果について見てみると，変更前，変更後の文法による文の正解率は，文法G_{all}ではそれぞれ88.45%，98.62%となり，文法G_{train}ではそれぞれ86.23%，94.66%となり，変更後の文法の方が8〜10%高くなっている．またG_{all}，G_{train}どちらの場合でも，変更後の文法で，変更前の文法による上位100位以内の文の正解率に達するには，上位10位以内の解析結果を考慮するだけで十分であり，我々のコーパスの変更方針が有効であることが分かる．

表2より，我々の文法G_{train}の被覆率は97%以上であり，広範囲の文の解析が可能であることが分かる．一方，被覆率，再現率ともに，我々の方針による変更によって1%程度低くなり，解析不能なものが変更前に比べて1%程度多く生じる．これは，構文解析結果の曖昧性を抑えるために非終端記号を細分化したことによるものである．文法G_{train}による上位100位の文の正解率の差が文法G_{all}によるものの差より小さくなる要因は，この再現率の差にある．しかし，文の正解率が我々の変更によって10%近く上がることから，被覆率や再現率がこの程度低下することは許容できると考えている[18]．

7　PGLR モデルによる解析結果を利用した係り受け解析

前節で，我々の方針により作成したコーパスから抽出した大規模日本語文法が，構文解析結果の曖昧性を抑え，文の正解率が約10%向上することを示した．しかし，構文解析結果の曖昧性を抑えるために，一部の曖昧性を同一の構造で表現することとし，その内部構造を厳密に決定していないため，文の正解率が高くなるのは当然であるという疑問が残る．そこで，PGLR モデルによる解析結果を利用した文節係り受け解析を行い，係り受け精度を調べた．

7.1　構文木からの文節係り受け関係の抽出

文節の係り受け関係は，構文木から取り出す．その手順を以下に示す．

(1)　文節区切りを決定する

(2)　構文構造をもとに，各文節について，係り先となる文節を決定する

例えば，図17の構文木の場合，文節区切りと各文節の係り先となる文節は表3のようになる．我々が使用しているコーパスに付与されている構造は句構造であり，文節中のどの語に係るかをさらに厳密に決定することも可能である．しかし，今回の実験では，どの文節に係るかのみを決定する．

変更後の文法では，連体修飾句の係り受け関係の構造は連用修飾句の範囲を変えない場合に限り，同一の構造(右下がりの構造)に制限している．そのため，連体修飾句の係り受け解析を行う際は，連体修飾関係の曖昧性をすべて考慮しなければならない．しかし，今回は，PGLR

18 コーパスをさらに増やせば，被覆率，再現率の差は小さくなる．

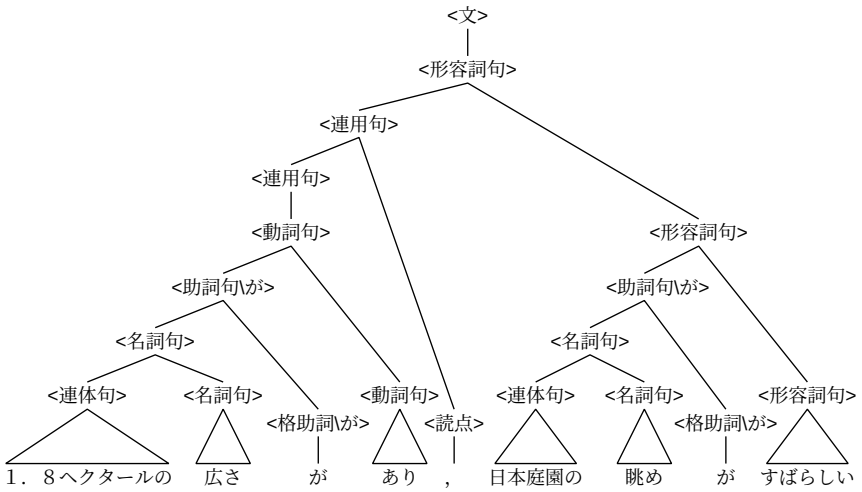


図 17　構文構造の例

表 3　図17の木構造から抽出される文節の係り受け関係

文節番号	文字列	係り先文節番号
1	１．８ヘクタールの	2
2	広さが	3
3	あり,	6
4	日本庭園の	5
5	眺めが	6
6	すばらしい	—(文末)

モデルによる生成確率1位の構文木中の連体修飾句は (意味的情報を用いず) 係り得る名詞の中で最も近いものを含む文節に係ることとする[19].

例えば,「青い目のアメリカから来た男性に会う」という文の「青い目のアメリカから来た男性に」という助詞句を考える．変更後の文法でこの文を構文解析すると，この助詞句について，図18(a)，(c)，(e) の3通りの解析木が出力される (中間ノードのラベルは省略する) [20]．図18(a) のような構文木が生成された場合，その係り受けは，図18(b) に示すように，文節「青い」が文節「目の」に，文節「目の」が文節「男性に」に直接係るとして係り受け精度を計算する (文節「アメリカから」は文節「来た」に，文節「来た」は文節「男性に」に係る)．図18(c)，(e) の場合は，それぞれ図18(d)，(f) に示すような係り受け構造となる．

連用修飾句の係り先の曖昧性は構文解析結果の曖昧性として残しているので，構文解析結果として出力された構造をそのまま利用する．

19 係り得る名詞の中で最も近いものに係るとすると，連体修飾句の係り先の正解率は70%弱であった．
20 実際には助詞句「青い目のアメリカから」が「会う」に係る解析木も出力されるが，ここの議論では省略する．

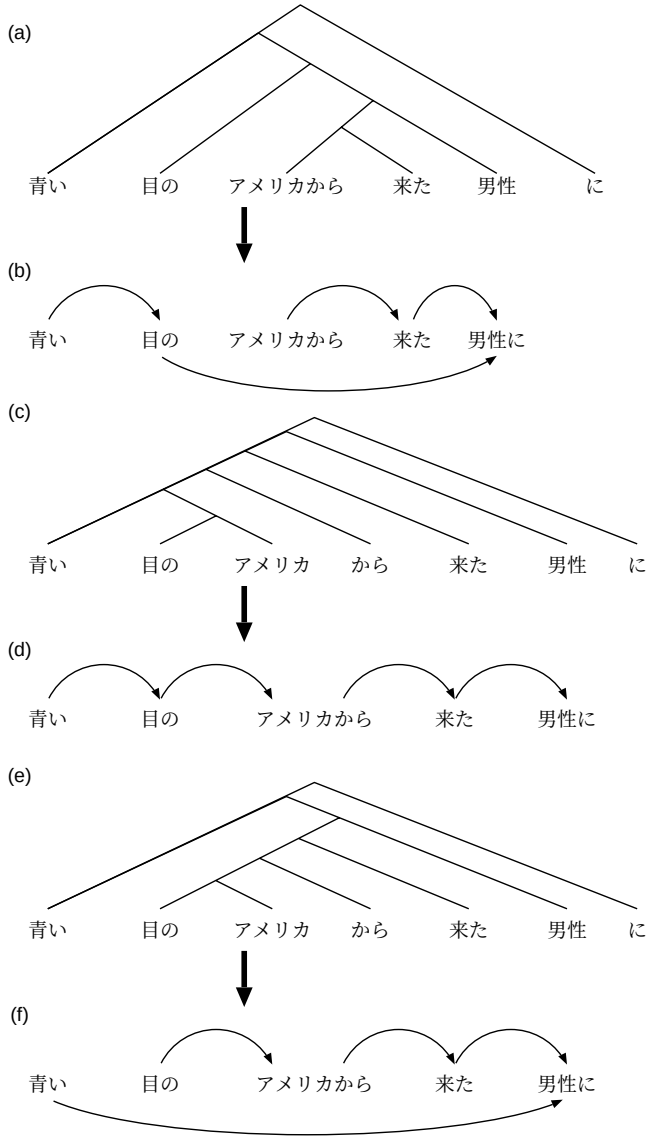


図 18　連体句の係り受け関係

複合名詞の内部の構造は語構成に関係なく同一の構造としているが，今回の実験では文節の係り受け構造を抽出するだけであるので，複合名詞の内部の構造までは考慮されない．

2つの文節が並列関係にあるか否かの曖昧性は，今回の評価実験では無視し，並列名詞句は連体修飾関係として，並列述語句と並列助詞句は連用修飾関係として扱う．

表 4　係り受け解析の実験結果

係り受けA型	係り受けB型	文正解率	文節不一致
91.32%	89.61%	61.54%	9

7.2　評価実験

評価は，変更後のコーパス8912文で行った[21]．このうち，評価用として100文をランダムに選択し，残りをPGLRモデルの学習用とした．この評価用データの100文は，1文あたり平均19.84形態素，7.16文節であり，並列構造を持つ文は17文含まれている．PGLRモデルにより評価用データ100文を構文解析し，生成確率が1位の構文木から，第7.1節で述べた方法により，半自動的に係り受け構造を抽出し[22]，その精度を以下の3つの尺度で評価した．

$$\text{係り受けA型} = \frac{\text{正しい係り受け関係の数}}{\text{総文節数} - (\text{テスト文の総数} \times 1)}$$

$$\text{係り受けB型} = \frac{\text{正しい係り受け関係の数 (文末2文節の関係以外)}}{\text{総文節数} - (\text{テスト文の総数} \times 2)}$$

$$\text{文正解率} = \frac{\text{正しい係り受け関係をすべて決定できた文数}}{\text{テスト文の総数}}$$

「係り受けA型」とは，全ての係り受け関係の正解率であり,「係り受けB型」とは，文末2文節の係り受け関係以外の係り受け正解率である.「文正解率」とは，文全体の文節の係り受け関係の正解率である．テスト文の正しい係り受け関係は，変更前のコーパスに付与されている構文木から，先に述べた方法で取り出した．その結果を表4に示す．ただし,「文節不一致」は文節区切りが正解と一致しなかった文の数を表す.

表4より，意味情報を全く利用しなくとも，PGLRモデルのみによる係り受け正解率が90%前後と非常に高いことが分かる．これは，Support Vector Machineや最大エントロピー法を用いた文節係り受け解析の手法の正解率と同程度である (内元, 関根, 井佐原 1999; 工藤 松本 2002)[23]．我々は，構文解析結果に対して意味解析を行うことを想定している．現在，本格的な意味解析の代わりに，SDSAの枠組みのみを利用して，単語の共起に関する統計データを用いた小規模な係り受け解析の実験を行っているが，非常に単純なスコア付けであるにも関わらず，93.0%の係り受け正解率 (係り受けB型)，68.8%の文正解率が得られることを確認している (八木, 野呂, 橋本, 徳永, 田中 2003)．今回は評価用データを100文として実験を行ったが，この規模は非常に小規模であるため，SDSAの有効性を示すには至っていない．しかし，我々は，実験結果か

21 第6節の評価実験では8911文を利用していた．実際のコーパスは8912文であるが，1文は変更前のコーパスから抽出した文法で構文解析を行うと，構文解析結果の数が膨大になり過ぎ，メモリ不足で解析できなかったため，この1文を除いて実験を行った．今回の実験は変更後のコーパスから抽出した文法のみを利用するので，8912文全てを利用している．

22 構文解析結果から文節の係り受け構造を機械的に抽出することは，一見すると簡単に見える．しかし，コーパスに付与されている構造が複雑であるため，完全に自動化することはできなかった．機械的に正しい係り受け構造を抽出できない部分は人手で修正している．今後，文節係り受け構造の抽出を容易にするために，構造の変更を検討していく予定である．

23 使用しているコーパスや実験の条件が異なるため，公平な比較ではない．

らSDSAによるアプローチが有効である可能性があると考えている．今後，コーパスサイズを大きくし，SDSAベースの本格的な意味解析への移行を検討する予定である．

8　まとめ

多様な言語現象を扱える大規模な文法は，構文構造付きコーパスから抽出することで構築可能であるが，そのようにして抽出した文法を用いた構文解析は，構文解析結果の曖昧性を極端に増大させることが多く，実用に供されていないのが現状である．本論文では，困難ではあっても曖昧性を増大させる要因を十分分析し，文法やコーパスの変更を繰り返すことによって，構文解析のための実用的な大規模文法を構築できることを示した．これらの文法，コーパスの変更点は，我々が本論文で扱ったコーパス特有の問題ではなく，一般性を持つものである．また，本論文では既存のコーパスに付与されている構文構造を変更しながら，抽出した文法による構文解析結果の曖昧性の削減を図っているが，新たに構文構造付きコーパスを作成する際には，この方針がコーパス作成基準となる．従来の構文構造付きコーパス作成基準は抽出したCFGによる構文解析結果の曖昧性を十分に考慮していないが，本論文で述べた方針に留意してコーパスを作成することで，CFG抽出に適したコーパスを新たに作成可能であると我々は考えている．

本論文で述べた変更を施したコーパスから抽出したCFGで構文解析を行うと，変更前のCFGを使用した場合に比べて，構文解析における曖昧性を大幅に抑えることが可能であることを実験的に示した．また，PGLRモデルによるスコア付けにより，上位100位以内の構文解析結果に対する文の正解率が10%向上することを確認した．さらに，生成確率が1位となる構文解析結果の文節の係り受けの精度は90%前後であり，既存の係り受け解析の手法と比較しても同程度の精度を有していることを確認した．我々は，SDSAの枠組みを利用し，共起情報を用いて係り受け解析を行うことにより，小規模実験の段階ではあるが，93%の係り受け精度が得られている (八木他 2003)．

今後の課題を以下に示す．

- 本論文で述べた変更方針で構文解析結果の曖昧性を大幅に抑制できたが，まだ十分ではない．例えば，日本語では助詞落ちが頻繁に出現するが，これを扱うことは曖昧性の増大につながる．省略されている助詞を前処理で補うべきか，意味解析で補うべきかを現在検討中である．
- 我々の文法は，構文解析における曖昧性を抑えるために一部の構文構造を制限している．後処理として想定されている本格的な意味解析の手法の検討が必要である．
- 本論文で述べたコーパスの変更方針では，形態素レベル (品詞レベル) の曖昧性を考慮していない．しかし，構文解析結果の曖昧性をさらに抑えるためには，形態素区切りの基準や品詞体系の見直しが必要である．現在，茶筌 (松本, 北内, 山下, 平野, 松田, 高岡, 浅原 2003) の品詞体系を基に検討しているところである．

- コーパスの作成や，変更方針の検討には複数の作業者が必要であり，さらに，その作業は長期間に及ぶため，バージョン管理が重要となる．そのために，コーパスをデータベース化し，検索システムやコーパス作成ツールを組み込むことで，コーパス作成のための大規模な支援システムを構築することが必要である．

謝辞

この研究は，東京工業大学21世紀COEプログラム「大規模知識資源の体系化と活用基盤構築」で行っているものである．コーパス作成，修正において協力を頂いた小林正博氏，大久保佳子氏をはじめとする(株)日本システムアプリケーションの皆様に感謝する．また，コーパス修正作業を担当して頂いた田中・徳永研究室の皆様にも感謝する．

参考文献

Charniak, E. (1996). “Tree-bank Grammars.” In *the 13th National Conference on Artificial Intelligence*, pp. 1031–1036.

Church, K. and Patil, R. (1982). “Coping with Syntactic Ambiguity or How to Put the Block in the Box on the Table.” *American Journal of Computational Linguistics*, **8** (3–4), 139–149.

Eisner, J. (1996). “Efficient normal-form parsing for combinatory categorial grammar.” In *the 34th Annual Meeting of the ACL*, pp. 79–86.

Inui, K., Sornlertlamvanich, V., Tanaka, H., and Tokunaga, T. (1998). “Probabilistic GLR parsing: A new formalization and its impact on parsing performance.” 自然言語処理, **5** (3), 33–52.

Jurafsky, D. and Martin, J. H. (2000). *Speech and Language Processing*. Prentice-Hall.

Komagata, N. (1997). “Efficient Parsing for CCGs with Generalized Type-Raised Categories.” In *IWPT 97*, pp. 135–146.

工藤拓 松本裕治 (2002). “チャンキングの段階適用による日本語係り受け解析.” 情報処理学会論文誌, **43** (6), 1834–1842.

黒橋禎夫 長尾眞 (1992). “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, **33** (8), 1022–1031.

黒橋禎夫 長尾眞 (1997). “京都大学テキストコーパス・プロジェクト.” 言語処理学会 第3回年次大会, pp. 115–118.

黒橋禎夫 (1998). 日本語構文解析システム KNP version 2.0 b6.

Marcus, M. P., Santorini, B., and Marcinkiewicz, M. A. (1993). “Building a Large Annotated Corpus of English: The Penn Treebank.” *Computational Linguistics*, **19** (2), 313–330.

Martin, W. A., Church, K. W., and Patil, R. S. (1987). “Preliminary analysis of a breadth-

first parsing algorithm: Theoretical and experimental results." In Bolc, L. (Ed.), *Natural Language Parsing Systems*, pp. 267–328. Springer-Verlag.

松本裕治, 北内啓, 山下達雄, 平野善隆, 松田寛, 高岡一馬, 浅原正幸 (2003). 形態素解析システム『茶筌』 version 2.3.0.

日本電子化辞書研究所 (2001). EDR 電子化辞書 2.0 版仕様説明書.

野呂智哉, 橋本泰一, 徳永健伸, 田中穂積 (2003). "大規模日本語文法の開発." テクニカル・レポート TR03-0006, 東京工業大学大学院情報理工学研究科計算工学専攻.

岡崎篤, 白井清昭, 徳永健伸, 田中穂積 (2001). "正しい構文木の選択を支援する構文木付きコーパス作成ツール." 人工知能学会 第 15 回全国大会.

白井清昭, 徳永健伸, 田中穂積 (1997). "括弧付きコーパスからの日本語確率文脈自由文法の自動抽出." 自然言語処理, **4** (1), 125–146.

白井清昭, 植木正裕, 橋本泰一, 徳永健伸, 田中穂積 (2000). "自然言語解析のための MSLR パーザ・ツールキット." 自然言語処理, **7** (5), 93–112.

Tomita, M. (1986). *Efficient Parsing for Natural Language*. Kluwer Academic Publishers.

内元清貴, 関根聡, 井佐原均 (1999). "最大エントロピー法に基づくモデルを用いた日本語係り受け解析." 情報処理学会論文誌, **40** (9), 3397–3407.

八木豊, 野呂智哉, 橋本泰一, 徳永健伸, 田中穂積 (2003). "単語の共起情報を利用した文法主導の係り受け解析." 情報処理学会自然言語処理研究会 2003-NL-157, pp. 17–24.

付録

A　文法作成の出発点として使用したコーパス

本節では，我々が文法作成の出発点として使用した構文構造付きコーパスについて述べる．

我々が使用したコーパスは，EDR コーパス中の文 (約 2 万文) に対し，人手で構文構造を付与したものである．基本的な構造は EDR コーパスに付与されている括弧付き構造に準拠しているが，単語区切り，品詞体系，構文構造それぞれについて，元となる EDR コーパスと異なる点がある．

A.1　基本構造

我々が使用したコーパスに付与されている構造は，以下の 3 つの層に分かれている (図 19).

第 1 層:　形態素と終端記号 (品詞) を対応付ける層.

第 2 層:　終端記号 (品詞) をやや粗い品詞分類に変換する層.

第 3 層:　実際の構文構造を示す層.

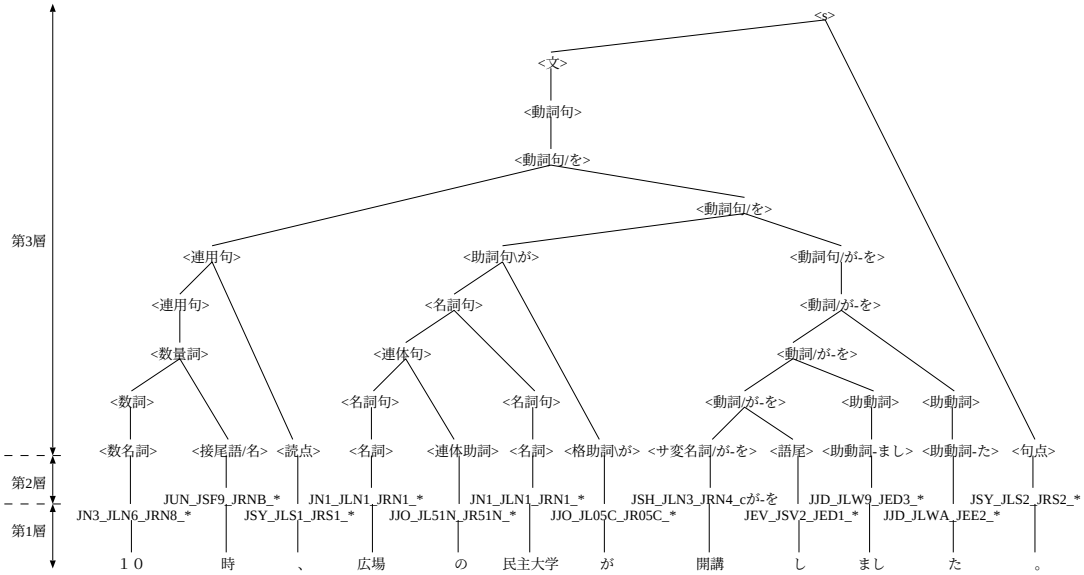


図 19　構文木を構成する3つの層

表 5　品詞の細分化の例(1)

単語	EDR コーパス	使用したコーパス
れんが	名詞	JN1_JLN1_JRN1_＊
埋め込(む)	動詞	JVE_JLV1_JRVM_cが-に-を
(埋め込)む	語尾	JEV_JSVM_JEE1_＊
と(格助詞)	助詞	JJO_JL30C_JR30C_＊
と(接続助詞)	助詞	JJO_JL30S_JR30S_＊
と(並列助詞)	助詞	JJO_JL30H_JR30H_＊
開講	動詞	JSH_JLN3_JRN4_cが-を
(開講)する	語尾	JEV_JSV2_JEE1_＊
(のぼり)はじめ(る)	動詞	JAX_JLV9_JRV1_＊

A.2　単語区切りと品詞体系

EDR コーパスで使用されている品詞は15種類しかなく，比較的粗い品詞体系となっている．しかし，これは構文解析を行うのに十分であるとは言えない．白井らは，助詞と記号を，表層情報(形態素)を利用して細分化しているが(白井他 1997)，それでも，まだ十分ではないと考えている．そこで，EDR 日本語単語辞書に記載されている品詞名，左右連接属性(連接属性対)，用言のとる表層格情報を組み合わせることにより，さらに細分化したものを第1層の品詞として使用している(表5)．ただし,「開講」等「する」を伴って動詞を形成するものは，EDR 日本語単語辞書では “JN1;JVE” という品詞が割り当てられているが，我々が使用したコーパスでは “JSH” で置き換えている．また,「(のぼり)はじめる」のように動詞に続く動詞や形容詞は，補助動詞(JAX)としている．

表 6　品詞の細分化の例 (2)

単語	EDR コーパス	使用したコーパス
(強め) てい (る)	助詞/動詞	JJP_JL26S_JRV1_*
によって	助詞/動詞/語尾/助詞	JJ1_JL48C_JR26S_h によって
において	助詞/動詞/語尾/助詞	JJ1_JL48C_JR26S_h において

EDR コーパスでは,「不安感を強めている」の「てい (る)」は「て (助詞)」,「い (動詞)」の 2 単語に分かれているが，我々が使用したコーパスでは，助動詞相当句として 1 単語としている (表 6).「によって」等の助詞相当句も同様である[24].

EDR 日本語単語辞書をもとにした品詞体系は非常に細かく，実際にコーパスに出現した品詞だけでも 600 種類を数える (存在し得る品詞を含めると優に 1000 種類を超える)．この品詞の上に直接構文構造を付けると，そのコーパスから抽出した文法規則が複雑になる．そこで，品詞分類を粗くする層として第 2 層を設けている．これにより，品詞分類が 100 種類程度に減少する．本論文では，構文構造を図示する際，必要でない限り，第 1 層の品詞を省略し，第 2 層の粗い品詞を終端記号とする．

A.3　構文構造

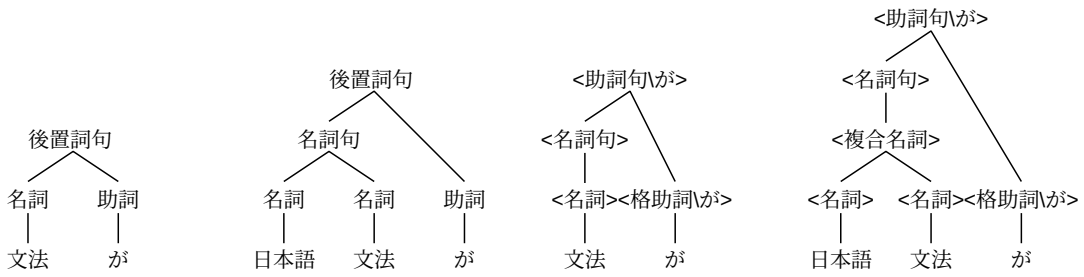
第 3 層の構造は基本的に EDR コーパスの括弧付けに従い，各中間ノードに非終端記号を付与する．ただし，我々が使用したコーパスでは 1 つの中間ノードに複数の非終端記号を縦に続けて割り当てることもあり，これにより，コーパスから抽出した文法が非終端記号の置き換え規則を含むようになる．例えば,「文法が」と「日本語文法が」という 2 つの句に対して，(白井他 1997) の場合は図 20(a) のような構造になり，我々が使用したコーパスでは図 20(b) のような構造になる．(a) から抽出される後置詞句に関する文法規則は，名詞句に助詞が結合する規則と名詞に助詞が結合する規則の 2 つになるが，(b) から抽出される助詞句に関する文法規則は，名詞句に助詞が結合する規則のみである[25]．その代わり，名詞句を構成するまでの部分が深くなるが，名詞や複合名詞から名詞句への置き換え規則を設け，類似の規則をまとめることで，句より上のレベルと下のレベルを明確に分けることができ，構造や抽出した文法規則が分かりやすくなる．

構造は基本的に EDR コーパスの括弧付けに従うが，次の場合には括弧付けとは異なる構造を付与する．

法，様相を表す助動詞:　「そうだ」など法や様相を表す助動詞は，EDR コーパスでは文全体に付加する構造になっている．しかし，白井らは，曖昧性を抑えるため，文末の最後の要素に結合する構造にしている．我々が使用したコーパスも，同様の構造になっている．

24 助詞相当句の左右連接属性は，左端の語の左連接属性と右端の語の右連接属性で決まるので，格助詞「に」ではじまり接続助詞「て」で終わる助詞相当句 (「によって」,「にとって」等) はすべて同じ品詞になってしまう．そこで，これらを区別するために，形態素ごとに分類している．

25 白井らが「後置詞句」と呼んでいる句は，我々が使用したコーパスでは「助詞句」と呼んでいる．



(a) 白井らによるラベル付け　(b) 我々が使用するコーパスのラベル付け

図 20　(白井他 1997) との構文構造の違い

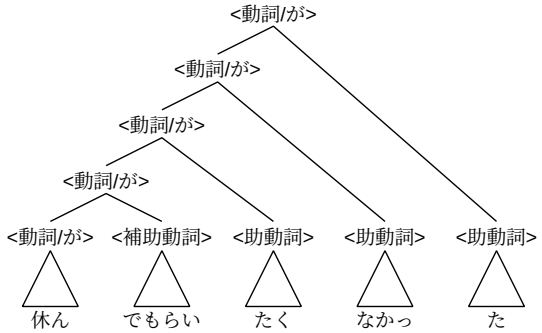


図 21　動詞に複数の助動詞が結合する場合の構造

フラットな構造:　白井らも指摘しているように，EDR コーパスの括弧付けの中には，細かい括弧付けがなく，多くの要素を 1 つの括弧でまとめてしまうものがある．その場合には，さらに細かい構造を付与する．

用言に結合する語尾，助動詞:　用言に複数の語尾や助動詞が結合する場合，EDR コーパスでは 1 つの括弧でまとめられているが，この部分は，左下がりの構造にしている (図 21)．この部分を EDR コーパスに従ってフラットな構造にすると，結合する助詞や助動詞の列のパターンだけ文法規則が必要となるが，こうすることで，少ない文法規則でより多くのパターンをカバーできるようになる．

我々が使用したコーパス中の用言を表す品詞には，それらがとる表層格の情報が付与されている．その表層格の情報は，第 3 層の構文構造にも引き継がれ，該当する助詞句によって打ち消される (図 22)．これにより，二重ヲ格等の制約を取り入れることが可能になる[26]．

略歴

野呂 智哉:　1977 年生．2000 年東京工業大学工学部情報工学科卒業．2002 年同

26 本論文では，簡略化のため，説明において必要でない場合には，表層格の情報を省略して図示することがある．

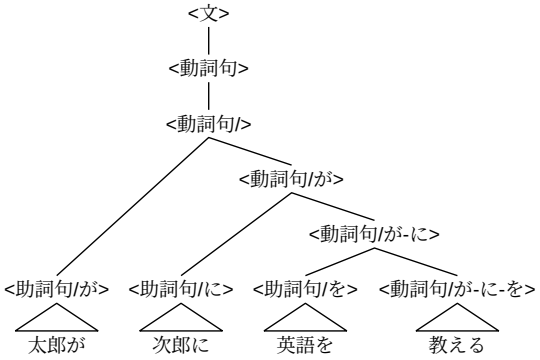


図 22　用言のとる表層格を考慮した構造

大学院情報理工学研究科計算工学専攻修士課程修了．同年同大学院情報理工学研究科計算工学専攻博士後期課程進学，在学中．日本語構文構造付きコーパスと日本語文法の構築に関する研究に従事．

橋本 泰一:　1999 年東京工業大学大学院情報理工学研究科修士課程修了．2001 年同大学院情報理工学研究科博士課程修了．現在，同大学大学院情報理工学研究科助手．博士 (工学)．情報処理学会，言語処理学会，各会員．

徳永 健伸:　1961 年生．1983 年東京工業大学工学部情報工学科卒業．1985 年同大学院理工学研究科修士課程修了．同年 (株) 三菱総合研究所入社．1986 年東京工業大学大学院博士課程入学．現在，同大学大学院情報理工学研究科助教授．博士 (工学)．自然言語処理，計算言語学，情報検索などの研究に従事．情報処理学会，認知科学会，人工知能学会，計量国語学会，Association for Computational Linguistics，ACM SIGIR，各会員．

田中 穂積:　1941 年生．1964 年東京工業大学工学部情報工学科卒業．1966 年同大学院理工学研究科修士課程修了．同年電気試験所 (現産業技術総合研究所) 入所．1980 年東京工業大学助教授．1983 年東京工業大学教授．現在，同大学大学院情報理工学研究科計算工学専攻教授．博士 (工学)．人工知能，自然言語処理に関する研究に従事．情報処理学会，電子情報通信学会，認知科学会，人工知能学会，計量国語学会，Association for Computational Linguistics，各会員．

(2004 年 2 月 16 日 受付)
(2004 年 6 月 14 日 再受付)
(2004 年 7 月 27 日 再々受付)
(2004 年 8 月 2 日 採録)