論文 / 著書情報
Article / Book Information

題目(和文)	大規模構文構造付き日本語コーパスの作成と日本語文法の構築
Title(English)	
著者(和文)	野呂智哉
Author(English)	Tomoya NORO
出典(和文)	学位:博士（工学）, 学位授与機関:東京工業大学, 報告番号:甲第6170号, 授与年月日:2005年3月26日, 学位の種別:課程博士, 審査員:田中穂積
Citation(English)	Degree:Doctor of Engineering, Conferring organization: Tokyo Institute of Technology, Report number:甲第6170号, Conferred date:2005/3/26, Degree Type:Course doctor, Examiner:
学位種別(和文)	博士論文
Type(English)	Doctoral Thesis

大規模構文構造付き日本語コーパスの作成と日本語文法の構築

計算工学専攻

野呂 智哉

目 次

図目次

表目次

第1章　序論

1.1　本研究の背景と目的

近年，情報化が進むにつれ，大量の電子テキストが流通するようになった．これらを有効活用するために，情報検索や情報抽出，機械翻訳，自動要約などの自然言語処理技術の重要性が増している．この自然言語処理技術は，様々な知識資源を必要とする．例えば，以下のようなものが知識資源として挙げられる．

単語辞書: 単語の品詞や活用形，意味を記述したもの．

シソーラス: 単語を意味的に分類し，体系化したもの．

共起辞書: 複数の単語が同時に出現し得るかどうかを記述したもの．

格フレーム辞書: 動詞がとり得る格に関する情報を記述したもの．

文法: 文の構造を記述するための規則を収集したもの．

コーパス: 現実に存在する文章を収集したもの．

これらの知識資源は，一般的には人手で作成されるが，人手による作成は莫大な時間と労力を必要とし，大規模なものを作成することは非常に困難である．小規模な知識資源では多様な言語現象を網羅できず，機械処理できる対象はごく限られたものとなってしまう．大規模化のためには，(半) 自動的にこれらの知識を作成する必要がある．

本研究では，これらの知識資源のうち，コーパスに注目する．先に述べたように，コーパスは，新聞や雑誌記事，小説，ウェブ文書，音声対話データの書き起こしなど，実在する文章を収集したものであるが，単に文章を大量に収集しただけでは，利用価値が低い．より多くの自然言語処理技術でコーパスを利用するためには，様々な情報を付与した注釈付きコーパスが必要となる．主な注釈付きコーパスを以下に挙げる[1]．

[1]付加的情報を持たない，単に文章を収集しただけのコーパスを，平文コーパスと呼ぶ．

品詞タグ付きコーパス: 各文を単語(形態素)ごとに区切り，各単語に対して品詞を割り当てたコーパス

構文構造付きコーパス: 各文に対して構文構造を付与したコーパス

語義タグ付きコーパス: 各単語に対して語義を割り当てたコーパス

パラレルコーパス: 複数言語間の対訳を収集し，文や句，単語間の対応関係を記述したコーパス

これらの注釈付きコーパスのうち，どのコーパスを使用するかは，その使用目的によって異なる．コーパスの主な使用目的には，以下の4つがある．

機械学習: 確率モデル等のパラメータの学習に利用する．形態素解析では，品詞タグ付きコーパスを利用し，コスト最小法における品詞間や単語のコスト，隠れマルコフモデルのパラメータの学習がある．構文解析では，構文構造付きコーパスを利用し，確率文脈自由文法(PCFG)の各CFG規則の適用確率や，確率一般化LRモデル[21]の各アクションの適用確率の学習がある．

直接利用: コーパスを用例集として利用する．用例に基づく格解析では，格関係を記述した構文構造付きコーパスを利用する．例えば,「大学が彼に期待をかける」という文について,「大学が」,「彼に」,「期待を」の格を決定する場合，コーパスから「～が～に～をかける」というパターンを収集し，その中で最も類似している文を参照する．用例に基づく機械翻訳では，パラレルコーパスを利用し，格解析と同様に類似文を検索して，その対訳を参照する．

他の知識資源の自動獲得: 文法，格フレーム辞書，シソーラスなど，コーパス以外の知識資源の自動獲得に利用する．コーパスからの自動獲得により，大規模な知識を容易に作成できる．

システムの評価: 形態素解析システムや構文解析システムなど，構築したシステムの正解率を評価するためのテストセットとして利用する．同じコーパスを利用することにより，異なるシステムを公平に評価し，比較することが可能になる．

本研究では，構文構造付きコーパスから文脈自由文法(CFG)[2]を抽出することに焦点を当てる．

[2]以降，特に断わらない限り，文脈自由文法を単に文法と表記する．

最近では，様々な言語において大規模な構文構造付きコーパス[3]の整備が進んでいる．代表的なものとして，英語ではPenn Treebankコーパス[35]，ドイツ語ではNegraコーパス[50]などが挙げられる．さらに，これらのコーパスから文法を自動獲得することで，文法作成者に大きな負担をかけることなく，コーパス中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となっている[7]．しかし，同様のことを日本語で行う際には問題が発生する．それは，Penn Treebankコーパスや Negra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ことである．日本語においても，EDR日本語コーパス[40]，京大コーパス[31]などが開発されているが，これらに付与されている構文情報は，Penn Treebankコーパスや Negraコーパスとは異なり[4]，同様の手法で文法を自動的に獲得することはできない[5]．同様の手法を採用するためには，Penn Treebankコーパスのように，完全な(全ての中間ノードにラベルが付与されている)構文構造を持つコーパスを作成する必要がある．

しかし，仮にPenn Treebankコーパスのような構文構造付きコーパスが日本語でも開発されたとしても，コーパスから抽出した文法自体に別の問題がある．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)[6]が出力されることである．その最大の要因は，従来のコーパス作成において，そのコーパスから抽出される文法による曖昧性に関する考慮が十分でなかったからである．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．実用的な文法をコーパスから抽出するためには，抽出した文法が出力する曖昧性を極力抑えるように，コーパスに付与する構文構造を決定する必要がある．

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ないため，可能な構文構造を全て列挙することになる．そこで，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)[24]を想定し，次の意味

[3]以降，特に断わらない限り，構文構造付きコーパスを単にコーパスと表記する．

[4]EDRコーパスは，句のまとまりを括弧で表現しているだけで，中間ノードにラベルが付与されていない．京大コーパスは，文節間の依存関係を表現しているコーパスであり，Penn Treebankコーパスなどとは性質が異なる．

[5]EDRコーパス中の構文構造の中間ノードのラベルを自動推定し，文法を抽出する手法はある[48]．詳細は後述する．

[6]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表記する．

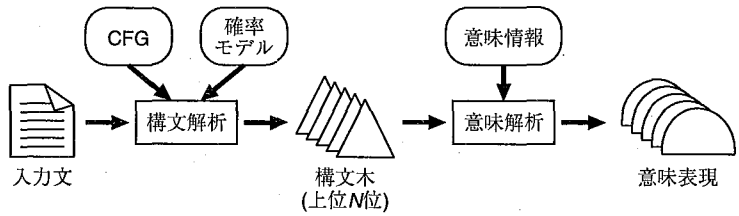


図 1.1: 本研究で想定する自然言語解析の流れ

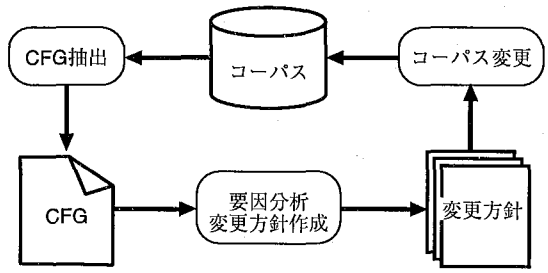


図 1.2: 抽出した文法による曖昧性を考慮した構文構造付きコーパス作成手順

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ことで(図1.1)，構文解析の段階の曖昧性を極力抑えることを考えている．

本研究では，Penn Treebank コーパスや Negra コーパスと同様の構文構造付き日本語コーパスを作成する．その際，以下の2点に留意してコーパス作成方針の検討を行う．

- コーパスから抽出した文法が出力する構文解析の段階の曖昧性を極力抑える．
- コーパスに付与されている構文構造は，その後の意味解析の段階においても有効となる．

本研究では，この検討を，EDR コーパス[7]に対し，以下の手順で行った(図1.2) [8]．

(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．

[7]EDR コーパスは括弧付きコーパスであり，非終端記号は割り当てられていない．実際には，本研究の前に，EDR コーパスに対して非終端記号などの情報を追加し，完全な構文構造を付与したものを用意した．このコーパスの概要は付録 A で述べる．

[8]この手順では既存の構文構造付きコーパスを変更しているが，新たにコーパスを作成する際には，この変更方針を作成方針として利用できる．

(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．

ただし，文法の抽出は，Charniakによる"tree-bank grammar"の抽出方法[7]と同様の方法を採用する．その結果，検討前のコーパスから抽出した文法と比較して，同じコーパス中の文を構文解析することで出力される解析木の数は10^{12}オーダから10^5オーダまで大幅に減少した．さらに，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)[21]によるスコア付け1位の解析木の文正解率は59.0%であった．また，品詞体系の異なるRWCテキストコーパス[19]に対し，同様の方針で構文構造を付与したところ，そのコーパスから抽出した文法でもほぼ同様の結果(解析木の数，文正解率)が得られた．一方，EDRコーパスに付与した構文構造から抽出した文法について，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，89.61%という高い係り受け精度が得られた．さらに，RWCコーパスに付与した構文構造から抽出した文法について，京大コーパス[31]を評価データ，正解データとして同様に文節の係り受けの精度を測定したところ，82.88%であった．今後，意味情報を本格的に利用することで，さらに精度向上が図れるという見通しを得ている．

1.2　本論文の構成

本論文の構成を以下に述べる．

第2章では，構文構造付きコーパスについて述べる．まず，句構造と依存構造の2種類の構文構造について説明し，一般公開されている主な句構造付きコーパス，依存構造付きコーパスを紹介する．

第3章では，関連研究として，コーパスから文法を抽出する主な研究を四つ紹介する．まず，Penn Treebankコーパスから単純に抽出した文法("tree-bank grammar")に関する研究を紹介する．次に，EDR日本語コーパスから，非終端記号を機械的に推定しながら文法を抽出する研究を紹介する．さらに，コーパスに対しどのような情報を追加すると，抽出した文法による解析精度が向上するかに関する考察を，Penn TreebankコーパスとNegraコーパスを対象に行った研究を紹介し，本研究との相違点を述べる．

第4章では，構文構造付きコーパスから抽出した文法が曖昧性を増大させる要因について述べ，それにもとづき，曖昧性の削減を考慮した具体的なコーパス変更方針を述べる．これらの検討はEDR日本語コーパスを対象に行う．

第5章では，実際にEDR日本語コーパスに対して構文構造を付与し，変更前，変更後のコーパスから抽出した文法で構文解析した際の曖昧性の増減と解析精度の変化を実験により示し，コーパス変更方針(作成方針)の有効性を明らかにする．具体的には，まず，この変更により曖昧性が大幅に減少することと文正解率が向上することを示す．さらに，変更後のコーパスから抽出した文法による解析結果と変更前のコーパスに付与されている構造を比較した場合の文節の係り受け精度が，意味情報を利用していないにも関わらず，他の係り受け解析に関する研究の結果と比較しても遜色ないことを示す．また，RWCコーパスに対しても同様の実験を行い，別のコーパスを利用しても同等の結果が得られることを示す．

第6章では，コーパス作成方針とコーパスから抽出した文法について考察し，今後の作成方針の再検討や新たな文のコーパスへの追加の際に留意すべき点を述べる．具体的には，コーパスから抽出できなかった文法規則に関する考察と，コーパス作成における作業者間のラベル付けの不一致に関する考察を行う．

最後に，第7章で本研究を総括し，今後の課題を述べる．

第2章　構文構造付きコーパス

本研究の目的は，日本語の構文構造付きコーパスを作成することである．本章では，まず，構文構造付きコーパスについて述べ，現在公開されている主な構文構造付きコーパスを紹介する．

2.1　句構造と依存構造

構文構造付きコーパスとは，その名の通り，コーパス中の各文に構文構造を付与したものである．構文構造には，大きく分けて2種類ある．

句構造: いくつかの単語が集まって句を構成し，句が集まってさらに大きな句を構成するというように，階層的に表現する構文構造 (木構造).

依存構造: 文の構成素間の支配，従属の関係 (係り受け関係) を表現する構文構造.

"An earthquake struck Northern California, killing more than 50 people." を例に，句構造と依存構造をそれぞれ図2.1と図2.2に示す．

句構造と依存構造の大きな違いは，階層関係に関する情報の有無である．これより，依存構造から句構造への変換は，階層関係に関する情報を必要とするため，句構造から依存構造への変換に比べて困難である．逆に，依存構造は，項構造 (argument structure) や結合価構造 (valency structure) を明確に表現できるのに対し，句構造は，階層関係にない離れた要素間の関係を直接表現できない[1]．

2.2　句構造付きコーパス

表2.1に主な句構造付きコーパスを示す．

[1]多くの句構造ベースの理論は，句構造を GB 理論 [9, 18] の d-structure，HPSG[45] の結合価構造，LFG[25] の f-structure，TAG[23] の解析木 (parse tree) などの依存関係のグラフに対応させている．

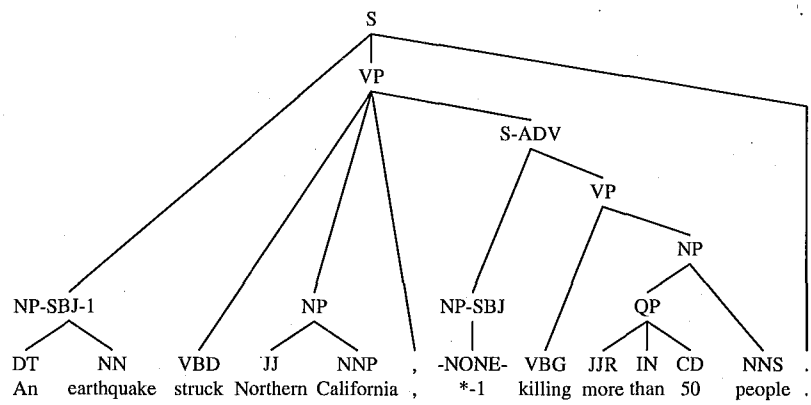


図 2.1: 句構造の例

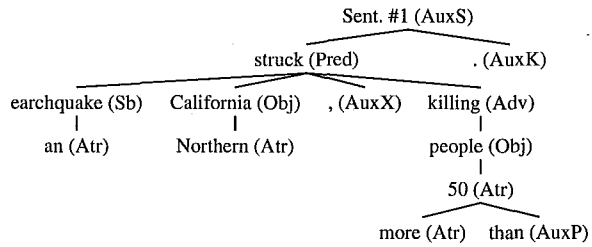


図 2.2: 依存構造の例

表 2.1: 主な句構造付きコーパス

	言語	ソース	文数	語数
Penn Treebank	英語	雑誌	49,000	1,200,000
Negra Corpus	ドイツ語	新聞	21,000	360,000
TIGER Corpus	ドイツ語	新聞	40,000	620,000
Penn Chinese Treebank	中国語	雑誌	15,000	440,000
Korean English Treebank	ハングル語，英語	QA 文	51,000	100,000
FLORESTA Corpus	ポルトガル語	新聞	8,300	200,000
EDR 日本語コーパス	日本語	新聞，雑誌	200,000	4,900,000

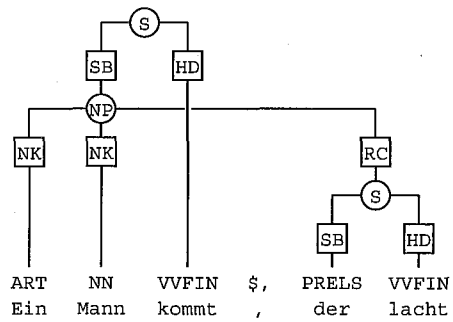


図 2.3: 枝の交差を認めた構造

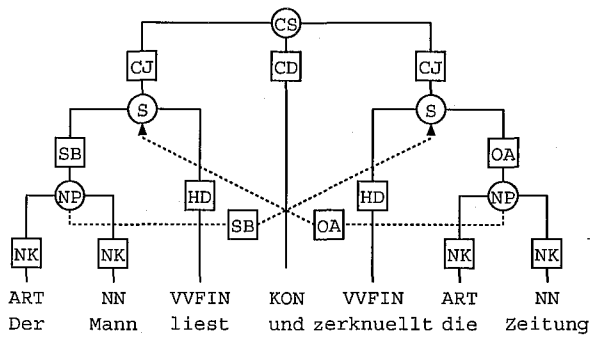


図 2.4: 補助枝を使用した構造

最も有名な句構造付きコーパスは Penn Treebank コーパス [35] である．Penn Treebank コーパスを利用した研究は多く，様々なアルゴリズムが提案されている．ドイツ語では，Negra コーパス [50] や TIGER コーパス [5] がある．これらは，Penn Treebank コーパスとは異なり，構文木の枝の交差を認め (図 2.3)，さらに，木構造を表現する枝の他に補助枝 (secondary edge) を利用することで，木構造では表現の難しい複雑な言語現象を表現している (図 2.4) [2]．主語，目的語などの文法機能に関する情報は，Penn Treebank コーパスは非終端記号の末尾に追加しているが，Negra コーパスと TIGER コーパスは枝に付与している．Penn Treebank コーパスと Negra コーパス，TIGER コーパスは，構文構造の付け方に違いがあるため，オリジナルのフォーマットは異なる．しかし，Negra コーパスと TIGER コーパスは，オリジナルのフォーマットの他に，Penn Treebank コーパスと同じフォーマットでも公開している．

[2]Penn Treebank コーパスでは，図 2.1 に示すように，null 要素 (“*”) やインデックス (null 要素や非終端記号の末尾の数字) を挿入することにより，複雑な言語現象を表現している．

表 2.2: 主な依存構造付きコーパス

	言語	ソース	文数	語数
Prague Dependency Treebank	チェコ語	新聞，雑誌	82,000	1,300,000
Prague Czech-English Dependency Treebank	英語 チェコ語	Penn Treebank	21,600	—
Alpino Dependency Treebank	オランダ語	新聞	7,100	150,000
京大コーパス	日本語	新聞	38,000	1,400,000

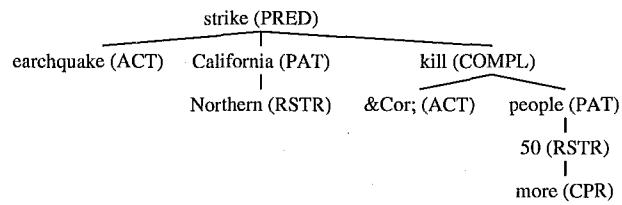


図 2.5: tactogrammatical level の構造

英語，ドイツ語以外の言語の句構造付きコーパスは，中国語では Penn Chinese Treebank コーパス [54]，ハングル語では Korean English Treebank コーパス[3][20]，ポルトガル語では FLORESTA コーパス [1] がある．これらは，Penn Treebank コーパスと同じフォーマットとなっている[4]．

日本語では，EDR 日本語コーパス [40] がある．規模は他のコーパスより大きいが，句のまとまりを括弧で表現しているだけで，各中間ノードに対して非終端記号は割り当てられていない．一方，中間ノードに非終端記号が割り当てられている日本語の句構造付きコーパスは，現在のところ，存在しない．

2.3　依存構造付きコーパス

表 2.2 に主な依存構造付きコーパスを示す．

最大規模の依存構造付きコーパスには，チェコ語の Prague Dependency Treebank コーパス [4] がある．このコーパスは，morphological level，analytical level(表層的構文構造，図 2.2)，tectogrammatical level(言語学的意味構造，図 2.5) の 3 つのレ

[3]ハングル語の各文に句構造が付与されているだけでなく，それらの英訳文に対しても句構造が付与されている．

[4]FLORESTA コーパスは，TIGER コーパスと同じフォーマットでも公開している．

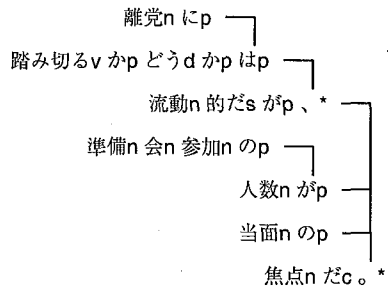


図 2.6: 京大コーパスの文節係り受け構造

ベルに分けて構造を付与している．英語でも，Penn Treebank コーパス中の文に依存構造を付与した Prague Czech-English Dependency Treebank コーパス [11] がある[5]．このコーパスは，Penn Treebank コーパスから抜き出した英文に対し Prague Dependency Treebank コーパスと同様に依存構造を付与するだけでなく，そのチェコ語訳に対しても依存構造を付与している．コーパス作成方針は，基本的に Prague Dependency Treebank コーパスと同じである．英語，チェコ語以外の依存構造付きコーパスは，オランダ語では Alpino Dependency Treebank コーパス [2] がある．

日本語では，京大コーパス [31] がある．EDR コーパスと同様に大規模であるが，京大コーパスには文節間の依存関係しか付与していないため (図 2.6)，文節内部の構造や主辞に関する情報などを持たない．さらに，文節間の依存関係の種類も，並列関係，部分並列内関係，同格関係，その他の係り受け関係の 4 種類しかなく，他のコーパスに比べて情報量が少ない．

[5]単語数については，記載がなかったため，不明である．

第3章　関連研究

本章では，関連研究として，四つの研究を紹介する．まず，Penn Treebank コーパスから単純に抽出した文法 (tree-bank grammar) に関する研究 [7] を紹介する．次に，EDR 日本語コーパスから，非終端記号を機械的に推定しながら文法を抽出する手法に関する研究 [48] を紹介する．さらに，コーパスに対し，どのような情報を追加すると，抽出した文法による構文解析精度が向上するかに関する考察を行った研究 [26, 47] を紹介する．

3.1　Tree-bank grammar

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 [35]．Charniak は，このコーパスから"tree-bank grammar" と呼ばれる文法を抽出し，人手で作成した文法との比較を行っている [7]．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 3.1)．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，特に単語数の多い文では，tree-bank grammar の方が解析精度が良くなることを示している．

ところが，Charniak は，tree-bank grammar が出力する曖昧性について明確に言及していない．tree-bank grammar の文法規則数は約 16,000 に上るが，Charniak は，コーパス中の出現頻度が 1 回のみである規則を削除し，規則数を抑えている．さらに，Krotov らは右辺の長い文法規則に注目し，それが他の文法規則で表せるものであれば削除している [28]．例えば，以下の規則があるとする．

$$\mathrm{NP} \rightarrow \mathrm{DT}\ \ \mathrm{NN}\ \ \mathrm{CC}\ \ \mathrm{DT}\ \ \mathrm{NN} \tag{3.1}$$

$$\mathrm{NP} \rightarrow \mathrm{NP}\ \ \mathrm{CC}\ \ \mathrm{NP} \tag{3.2}$$

$$\mathrm{NP} \rightarrow \mathrm{DT}\ \ \mathrm{NN} \tag{3.3}$$

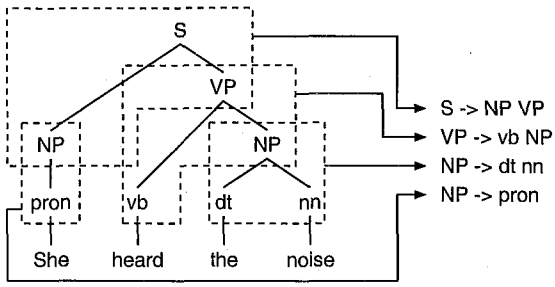


図 3.1: Penn Treebank コーパスからの tree-bank grammar の抽出

この時，規則 (3.1) は，規則 (3.2) と規則 (3.3) の二つで表現できる．これは，正しくは規則 (3.2) と規則 (3.3) を利用した構造を付与するべきであるが，実際には規則 (3.1) を利用した構造が付与された「部分的にラベル付けされた」文が混在していると考えられる．そこで，規則 (3.1) を削除することが可能であると判断できる[1]．

これらが結果として曖昧性を抑える役割を果たしていると考えられるが，第 4.1 節で述べるように，曖昧性を増大させる要因は，コーパス中の出現頻度の低い規則や，右辺の長い規則だけではない．Charniak は，確率付き文脈自由文法 (PCFG) による最良優先解析 (best-first parsing) を採用し，解析途中で曖昧性を絞り込んでいる．しかし，これによって効果的に曖昧性を絞れるのは，割り当てられる確率の低い規則，すなわち，コーパス中の出現頻度の低い規則が曖昧性を増大させている場合に限られ，そうでない場合には，正しい解析結果が途中で棄却されてしまう可能性がある．根本的な解決を図るためには，どの構文構造や文法規則が曖昧性を増大させるのかを分析し，可能ならば曖昧性を抑えられるようにコーパス作成方針を変更する必要がある．

3.2　EDR コーパスからの自動獲得

第 1 章で述べたように，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せず，第 3.1 節で紹介した Charniak と同様の手法で文法を抽出することはできない．これに対し，白井らは，EDR 日本語コーパス [40][2]から文法を自動獲得する手法を提案している [48]．EDR コーパスは括弧付きコー

[1]右辺の長い規則の中には言語学的に意味のある規則もある．そのような規則を削除してしまうことを予防するために，Krotov らは，その規則が使われる確率をもとに削除するかどうかを決定している．

[2]以降，特に断らない限り，EDR 日本語コーパスを単に EDR コーパスと表記する．

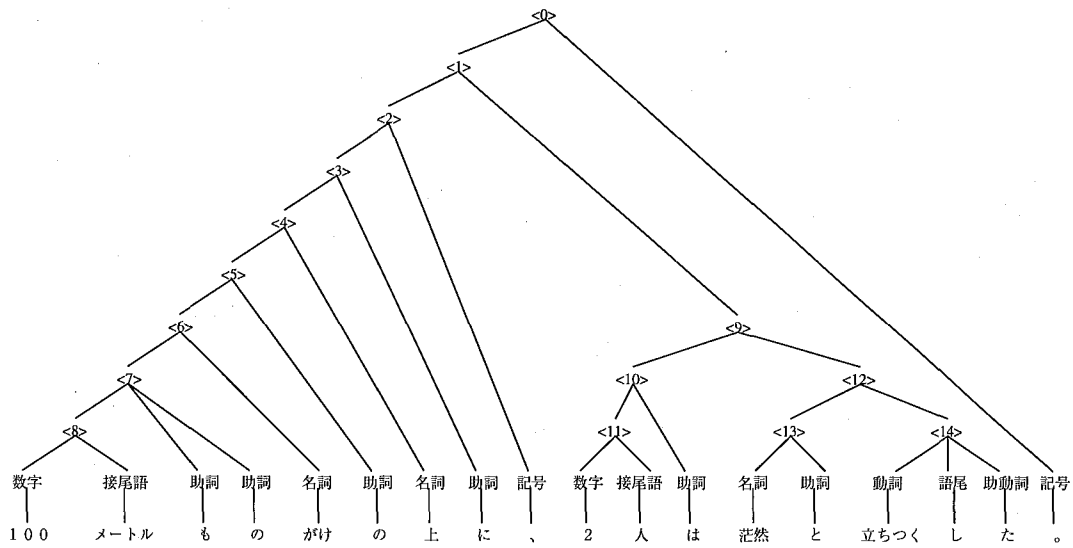


図 3.2: EDR コーパスの構文構造

パスであり，図3.2に示すように，各中間ノードに非終端記号が割り当てられていないため，適切な非終端記号を自動的に決定しなければならない．その手法として Inside-Outside アルゴリズム [34] などがあるが，白井らは，句の主辞の情報を利用し，他の手法より少ない計算量で非終端記号を決定している．

このようにして抽出した文法では，他の手法と同様に以下の問題点がある．

- 文法サイズが大きい
- 曖昧性が大きい

文法サイズの縮小については，Krotov ら [28] と同様の手法を利用している．曖昧性の抑制については，以下の3点についてコーパス中の構造を変更している．

同一品詞列の扱い: 1種類の品詞のみを支配するノード (例えば，名詞のみから成る複合名詞) の下の構造は右下がりの構造に統一．

品詞の細分化: 記号，助詞について，さらに細分化．

助動詞に対する構造の統一: 法，様相を表す助動詞「そう」,「だ」を含む文の構造を，他の助動詞の場合と同じ構造に統一．

これにより曖昧性を抑制しているとは言え，依然として曖昧性は大きい (187,802文から抽出した文法で平均 3.24×10^9 個の解析結果)．曖昧性の巨大さから見て，

他にも曖昧性を増大させる要因があると考える．また，EDRコーパスで使用している品詞が15種類と非常に粗い．白井らは，記号と助詞について自動的に細分化しているが，自動的な細分化には限界がある．例えば，白井らは助詞を形態素ごとに細分化しているが,「太郎と会う」の「と」と「太郎と花子がいる」の「と」は用法が異なるため，区別すべきである．この区別を自動的に行うことは困難であり，人手による細分化が必要である．

さらに，非終端記号の自動推定にも限界がある．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非終端記号を考える(スラッシュは単語区切りを表す)．白井らのアルゴリズムでは，末尾の「か」が助詞であることから“後置詞句”となり，次の文法規則が得られる．

$$\text{後置詞句} \rightarrow \text{動詞}\ \ \text{語尾}\ \ \text{助動詞}\ \ \text{助動詞}\ \ \text{助詞} \tag{3.4}$$

直感的には後置詞句ではなく動詞句の方が適切であるが，自動的な推定では，意図しない割り当てを例外として正確に除外していくことは困難である．これは，曖昧性の増減と直接は関係のないことであるが，人間が見て妥当な非終端記号を割り当てるためには，自動的に非終端記号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人手で作成すべきである．

3.3　解析精度向上に有効な構文情報の検討

本節では，コーパスから抽出した文法について，コーパスに追加した情報が，解析精度の向上にどの程度貢献しているかを検討している論文を二つ紹介する．一つは英語(Penn Treebankコーパス)での検討[26]，もう一つはドイツ語(Negraコーパス)での検討[47]である．

3.3.1　Penn Treebankコーパスでの検討

これまで，PCFGによる解析精度の向上には，各非終端記号に主辞を追加する語彙化(lexicalization)が必要であるとされてきた．しかし，語彙化しなくても，各ノードについて親ノードの非終端記号を追加するだけで(図3.3)解析精度が向上するという結果も示されるなど[22]，語彙化による解析精度の向上がどれほど有効であるか疑問を生じさせる結果がいくつか報告されている．Kleinらは，語彙化しない段階でどの程度解析精度を向上させられるかを検討している．

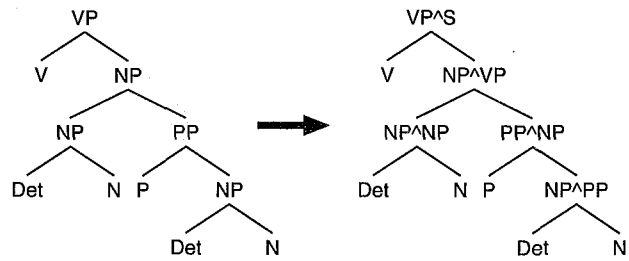


図 3.3: 親ノードの非終端記号の追加

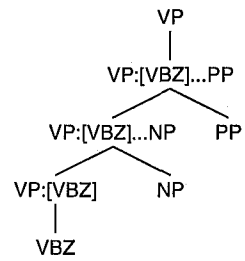


図 3.4: VP → VBZ NP PP の水平方向のマルコフ化 ($h = 1$)

Klein らは，まず，垂直方向および水平方向のマルコフ化を検討している．垂直方向のマルコフ化とは，各ノードの上位 $v-1$ 個のノードの非終端記号を追加することであり，図3.3の例は $v = 2$ の場合を表す ($v = 1$ の場合は，何の情報も追加しない)．一方，水平方向のマルコフ化とは，多数の子ノードを持つ中間ノードについて，その構造を二分木の組み合わせで表現することである．具体的には，主辞となる子ノードから開始し，その右隣または左隣の子ノードと結合しながら，そのノードの非終端記号を追加していく．ただし，追加する非終端記号は，最近の h 個に制限する (水平方向のマルコフ化を行わない場合を $h = \infty$ と表記する)．また，新たに作られる中間ノードの名前は，親ノードと主辞となる子ノードの組み合わせで表される．図3.4に，文法規則 VP → VBZ NP PP の $h = 1$ の水平方向のマルコフ化の例を示す．マルコフ化を Penn Treebank コーパスで行い，抽出した文法による解析精度 (F 値) を調べたところ，$v = 3$，$h \leq 2$ の時に最大となった．ただし，$h \leq 2$ とは，新しく作られた非終端記号の出現頻度が 10 回未満となる場合は縮退することを意味する．例えば，非終端記号 $<$ VP : [VBZ] ... PP PP $>$ の出現頻度が 10 回未満であった場合，$<$ VP : [VBZ] ... PP $>$ とする．

次に，$v \leq 2$，$h \leq 2$ の場合をベースラインとして，以下の変更を行っている．

- 一分木に対する情報の追加
  - 子ノードを一つだけ持つノードをそれ以外のノードと区別 (UNARY-INTERNAL)
  - 兄弟ノードを持たない限定詞または副詞をそれ以外の限定詞や副詞と区別 (UNARY-DT，UNARY-RB)
- 品詞タグの細分化
  - 品詞タグを親ノードの非終端記号ごとに細分化 (TAG-PA)
  - 品詞 IN(接続詞，補文標識，前置詞) を 6 種類に細分化 (SPLIT-IN)
  - 助動詞"be" と"have" を区別 (SPLIT-AUX)
  - 接続詞"but" と"&" を区別 (SPLIT-CC)
  - 記号"%" を区別 (SPLIT-%)
- Penn Treebank コーパスに付与されている機能タグの利用[3]
  - 時間を表す名詞句のためのタグ TMP を付与 (TMP-NP)
  - 空の主語 (empty subject) を持つ文 (S) を区別 (GAPPED-S)
- 主辞の追加
  - 所有格の名詞句を区別 (POSS-NP)
  - 動詞句 (VP) を主辞ごとに区別 (SPLIT-VP) [4]
- 距離による区別
  - 品詞のみを子ノードに持つ名詞句を区別 (BASE-NP)
  - 動詞を含む名詞句を区別 (DOMINATES-V)
  - 右再帰する名詞句を区別 (RIGHT-REC-NP)

これらの変更の結果，解析精度 (F 値) は 86.32%となり，最新の語彙化モデルとの差は小さいことを示している．

[3]ベースラインとなる文法を抽出する際，機能タグはすべて除去されている．
[4]主辞が定動詞の場合は区別しない．

Klein らは，解析精度を向上させるために必要な情報について検討しているが，その文法が出力する曖昧性に関する議論はしていない．結果的に曖昧性の抑制につながっている変更点もあるが，曖昧性の観点からの変更の有効性の検討も必要であると考えている．例えば，英語には PP attachment 問題がある．前置詞句がどの語を修飾するかは，構文情報だけでは決定できず，意味情報を必要とする．このような場合の曖昧性について検討することは，その後の意味解析においても有用であると考えている．

3.3.2　Negra コーパスでの検討

これまで，構文構造付きコーパスを利用した統計的構文解析は良い成果をあげてきた．しかし，そのほとんどは英語を対象とした場合，特に Penn Treebank コーパスを対象とした場合であり，他言語では，英語の場合ほど大きな向上は見られないという報告もある．例えば，英語では語彙化により解析精度が向上するとされているが [6, 8, 12]，チェコ語や中国語では，精度は向上するものの英語と比較するとその差は小さく [3, 13]，また，ドイツ語では語彙化しない場合より精度が悪化する [16]．英語に適した設定がそのまま他言語にも適しているとは限らず，対象とする言語に適した設定を検討する必要がある．Schiehlen は，ドイツ語を対象とし，Negra コーパス [50] を利用して第 3.3.1 節で紹介した Klein らと同様の検討を行っている [47]．

まず，親ノードの非終端記号の追加 (垂直方向のマルコフ化) と，(水平方向の) マルコフ化を行っているが，どちらの場合も，精度 (依存関係の F 値) の向上に寄与しない．親ノードの非終端記号の追加が精度の向上に貢献しない理由は，コーパスから抽出した文法が十分ではないからである．ドイツ語は英語に比べて語順に関する制約が弱く，あらゆる語順を網羅するためには，より多くの文法規則が必要となる．親ノードの非終端記号を追加することは，文法規則をさらに細分化することになり，文法規則数の増大により解析精度が向上しない．一方，水平方向のマルコフ化は文法規則の一般化を行うものであるが，それにより解析において重要となる文法機能に関する情報が失われてしまうことが，精度が向上しない要因として挙げられている．

さらに，垂直方向および水平方向のマルコフ化を行わない場合をベースラインとして，以下の変更を検討している．

- トレース情報を追加 (Traces)

- マルコフ化以外の方法による文法規則の一般化
  - 並列関係 (等位関係) にある句を他の句と区別しない (Coordinated Categories)
  - grammatical rule [5]の右辺の終端記号 (品詞タグ) を，対応する非終端記号に置換 (Hiding POS Tags)
  - 特殊な品詞タグ (基数，最上級，固有名詞など) を一般的な品詞タグに統一 (Multi-Word Lexemes)
- 構文的役割のトップダウンによる伝播
  - 関係節を区別 (Relative Clause)
  - 修飾句になる名詞句を区別 (Adjunct NP)．Klein らの TMP-NP と類似．
  - 数字と単位 (助数詞) から成る句を区別 (Measure Phrase)
  - "von (of)" で始まる前置詞句を区別 (Pseudo-Genitive PP)
  - 副詞を細分化 (Adverbial Classification)
  - 前置詞"bis (to)" など接続詞的役割を果たす (等位接続詞以外の) 語を区別 (Coordinating Item)
  - "than Peter" のような比較を表す句を他の名詞句と区別 (Comparative Phrase)
  - 格情報を追加 (Case)
- 語彙情報のボトムアップによる伝播
  - 動詞を活用形により細分化 (Verb Form)．Klein らの SPLIT-VP と類似．
  - 助動詞 "sein (be)"，"haben (have)"，"werden (will)" を細分化 (Auxiliary Split)．Klein らの SPLIT-AUX と類似．
  - 中性，単数，主格または対格の人称代名詞 "es (it)" を区別 (Neuter Pronoun)
  - 補文標識，従属接続詞，疑問副詞で始まる節を他の節と区別 (Subordinating Conjunctions)

[5]右辺が終端記号だけで表されている規則を lexical insersion rule，それ以外の規則を grammartical rule と呼んでいる．

- 動詞，名詞，形容詞の下位範疇化フレームを利用 (Subcategorization)
- コーパス処理
  - 1文につき1つの構文木が割り当てられるようにする (Sentence Boundary) [6]
  - 地名辞典を利用してNamed Entityの認識を行う (Named Entity Recognition)

これらのうち，トレース情報の追加 (Traces) と文法規則の一般化 (Coordinated Categories，Hiding POS Tags，Multi-Word Lexeme) は精度向上に貢献しなかった．残りの変更を行った場合，精度 (依存関係のF値) は81.69% (Named Entity Recognitionを除くと81.03%) となり，他の手法 [16, 46] より良いことを示している[7]．

Schiehlenが行った変更は，一部はKleinらの変更と類似しているが，大部分において異なる．これは，解析精度向上に寄与する変更点が言語によって異なることを意味し，日本語でも同様の検討を行う必要がある．しかし，第3.3.1節で紹介したKleinらと同様，Schiehlenも，文法が出力する曖昧性に関する議論はしていない．解析精度の向上やその後の意味解析のためには，曖昧性の抑制も考慮すべきであると考えている．

[6]Negraコーパスでは，ラベル付け前に文境界を自動的に決定する．この決定が誤っている場合，作業者は文境界を訂正せずに複数の構文木に分割して割り当てている．

[7]Dubeyら [16] は構成素のF値を求めている．その値は71.12%であり，Schiehlen[47]の68.36%を上回っている．しかし，マルコフ化において水平方向を$h \leq 2$，垂直方向を$v = 2$とした場合，構成素のF値は71.82%となり，Dubeyらの結果を上回る．

第4章　コーパス作成方針の検討

大規模な構文構造付きコーパスから抽出した文法をそのまま利用して構文解析を行うと，多数の曖昧性を出力する．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，本章では，構文構造付きコーパスから抽出した文法が曖昧性を増大させる要因を分析し，分析結果に基いてコーパスの変更方針を提案する．

4.1　構文解析結果の曖昧性を増大させる要因

曖昧性を増大させる要因は，以下の4種類に大別できる．

作成者の誤り (要因1): 構文構造は人手で付与するため，作成者による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，誤った構造を出力し，それが無意味な曖昧性の増大につながる可能性がある．

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，コーパス作成は一人ではなく複数人で，長時間をかけて行うことが一般的である．この時，作成者または作成日時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，冗長な文法規則を含み，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，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．しかし，コーパスから抽出した文法 (tree-bank grammar) の各規則は，ノード間の親子関係に関する情報しか持たず，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文法抽出の段階で欠落することで，構文解析において，構文的に誤った解析木を余分に出力する可能性がある．例えば，図

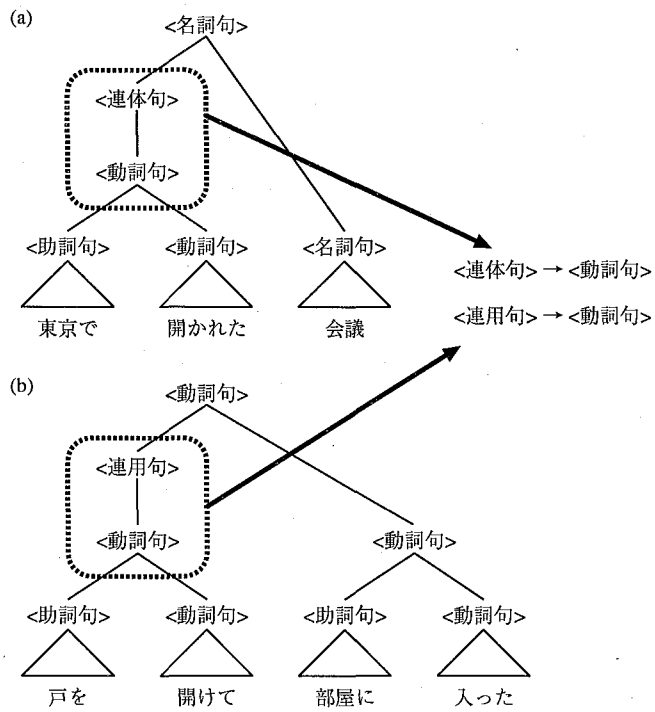


図 4.1: 文法抽出時における構文情報の欠落

4.1 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り得る．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，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も生成し，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，構文情報だけでなく意味情報も必要とするものがある．例えば，名詞句「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，構文情報だけではそのいずれが正しいかを決定できない．第 1 章で述べたように，本研究で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

味性をむやみに列挙することは，構文解析結果を組み合わせ的に増大させることになる[1].

要因1と2はコーパスの誤りであるため，訂正すべきものとして以下の考察から除外する[2]．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消が困難な曖昧性の場合には，その曖昧性を包含した単一の構文構造をコーパスに付与し，文法を再抽出する．すなわち，再抽出した文法による構文解析結果では，要因4による曖昧性は区別されないことになる．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，具体的な変更方針について述べる．

4.2　コーパス，文法の変更方針

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種の曖昧性を完全に除外し，一つの意味に対して一つの解析木を出力する (one syntactic structure per semantic reading) 手法を提案している [17, 27]. 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，コーパスと文法そのものを変更しながらこの曖昧性を抑える．

さらに，本研究では，要因4の曖昧性は単一の構文構造で表現することとしている．しかし，この方針によって曖昧性を抑えることは，その後の意味解析を困難にすることもあり得る．構文解析時は包含されている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，曖昧性を抑えるかを検討する必要がある．

本研究で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如 (要因3)

(2) 複合名詞内の構造の曖昧性 (要因4)

---

[1]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 [10, 36]，文全体の曖昧性の増大の最大の要因の一つとなる

[2]コーパス中の誤りを検出 (訂正) する手法は，既にいくつか提案されている [14, 15].

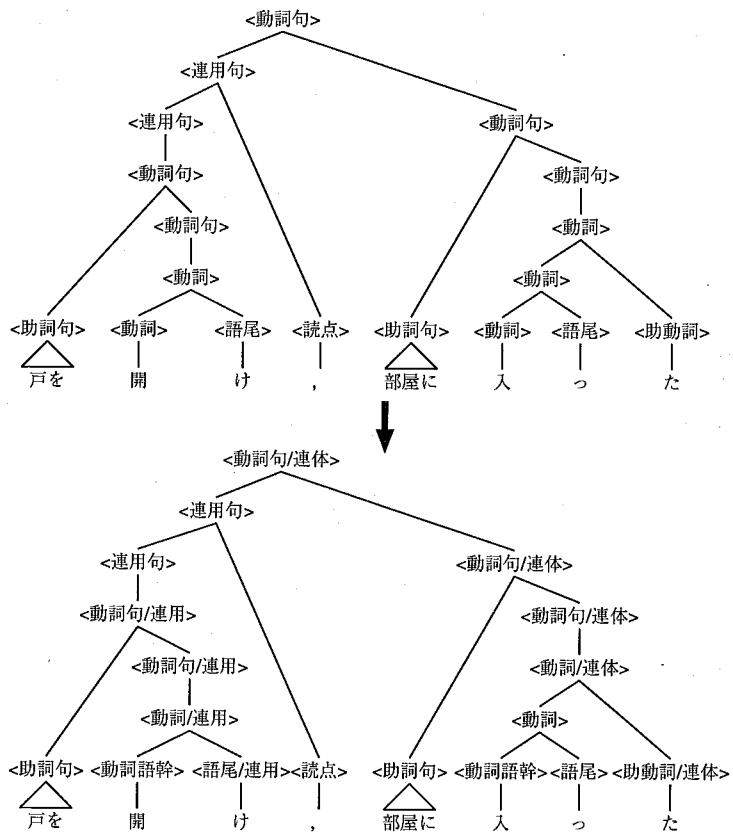


図 4.2: 活用形に関する情報の付与

(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，以下に変更方針を述べる．

4.2.1　用言の活用形に関する情報の欠如

用言の活用形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という曖昧性が発生することを，第 4.1 節で，要因 3 の曖昧性の例として述べた．実際，本研究で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，用言などの語尾や助動詞の活用形に関する情報を上位ノードに引き継ぐように変更する (図 4.2)．この変更は，Klein ら [26] の “SPLIT-VP” や Schiehlen[47] の “Verb-Form” と類似している．ただし，未然形，連用形など活用形ごとに細か

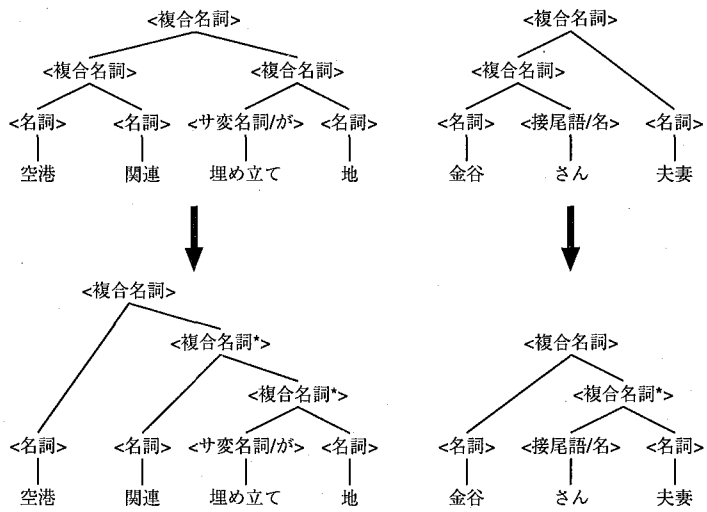


図 4.3: 複合名詞の構造の変更

く分類するのではなく，その語が末尾に出現することで連用修飾または連体修飾になり得る場合にのみ“連用”，“連体”というラベルを追加する．これは，活用形に関する情報を追加する目的が，その用言が連用修飾になり得るものか，連体修飾になり得るものかを区別するためだけであり，それ以外の情報は必要ないからである．

4.2.2　複合名詞内の構造の曖昧性

一般に，複合名詞内の構造の曖昧性を構文解析の段階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曖昧性として出力すべきではないと考えている．白井らも，名詞連続などの同一品詞列を支配するノードの下の構造について，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 [48]．本研究でもその方針に倣い，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 4.3) [3]．ただし，本研究では，同一品詞列ではなく，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，これを複合名詞と呼んでいる．

[3]構造を右下がりにする際，複合名詞の根ノードと内部ノードの非終端記号を図 4.3 のように区別している．もし，これらを同一の非終端記号にすると，抽出した文法は,「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ことになってしまう．

4.2.3　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．本研究では，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)[4]，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(右下がりの構造)にする(図4.4，図4.5)[5]．つまり，コーパスから抽出した文法による構文解析では，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力せず，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同一の構造にすることは，構文解析後の意味解析が困難になると考えるからである．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」それぞれを一つの複合名詞にまとめる(図4.6の破線で囲まれた部分)

(2) 「日本の」と「流通制度」，「(日本の)流通制度の」と「改善」の二つの連体修飾関係をまとめる(図4.6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4.6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関係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

[4]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(付録A)，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4.4に示すように，用言のとる表層格の情報は無視する[43]．

[5]構造を変更する際，図4.5に示すように，右側の名詞句(係り先)と左側の連体句(係り元)の下の名詞句を区別している．もし，これらを同一の非終端記号にすると，抽出した文法は右下がり以外の構造を出力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．

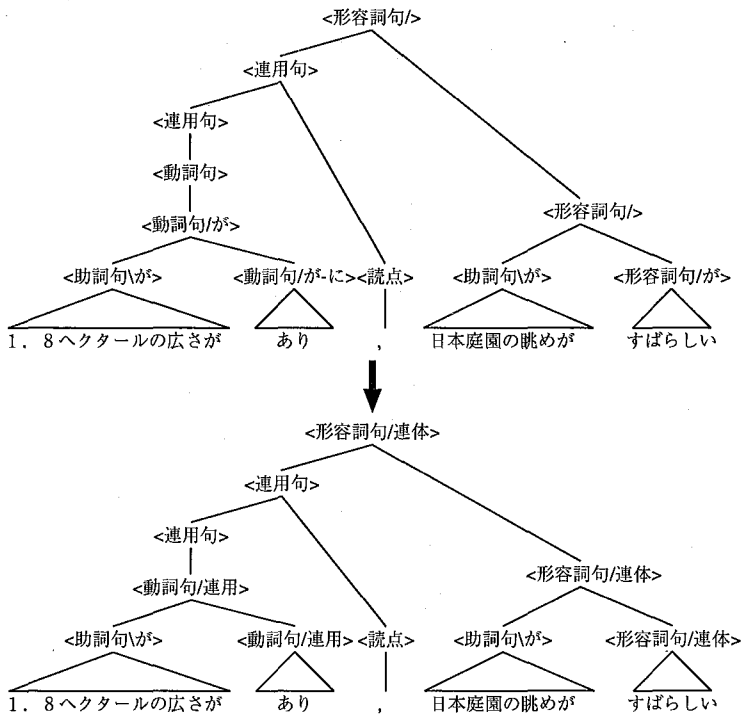


図 4.4: 連用修飾句の係り先に関する変更

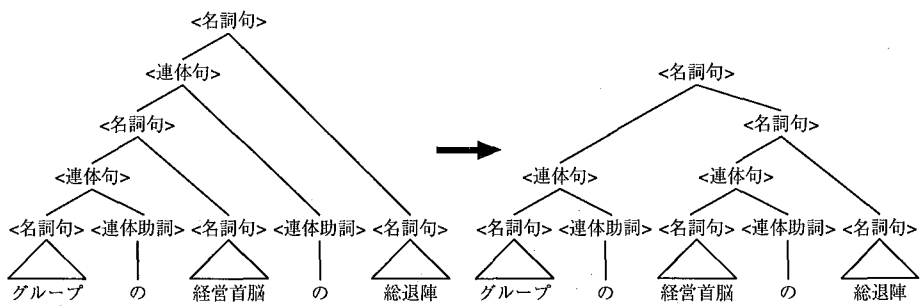


図 4.5: 連体修飾句の係り先に関する変更

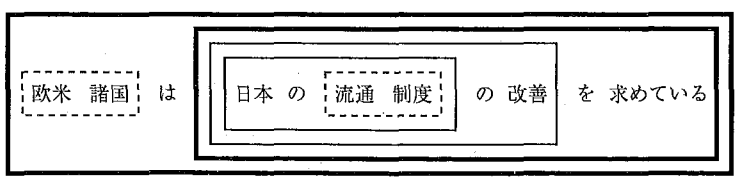


図 4.6: 単文「欧米諸国は日本の流通制度の改善を求めている」の構造

構造がとらえにくくなる．その結果，抽出した文法による構文解析後の意味解析を困難にする要因となる．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容易にするとともに，構文解析の段階の曖昧性を抑えることが可能であると考えている．

別の理由は，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と考えているからである．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品詞レベルでの解決は難しい．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4.7にそれぞれの例を示す．太い実線に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，破線で囲まれた語は動詞を，網掛けの長方形で囲まれた語は連体修飾を受ける名詞を，矢印は修飾関係を表す．

「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 (太い実線で囲まれた) 連用修飾句は「新しい環境への適応能力を」であることに変わりはない (図4.7(a), (b))．ところが,「百年の歴史を持つ祭り」では，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」に係る場合は「歴史を」のみが動詞「持つ」に係る連用修飾句となる (図4.7(c), (d))．

本研究では，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で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，コーパス中では，連体修飾関係を表す構造を同一の構造で表現する．すなわち,「新しい環境への適応能力を調べる」の場合は意味に関係なく図4.7(b) の構造とし,「百年の歴史を持つ祭り」の場合は意味を考慮して図4.7(c) の構造とする[6]．

[6]このコーパスから抽出した文法で構文解析を行うと,「新しい環境への適応能力を調べる」の場

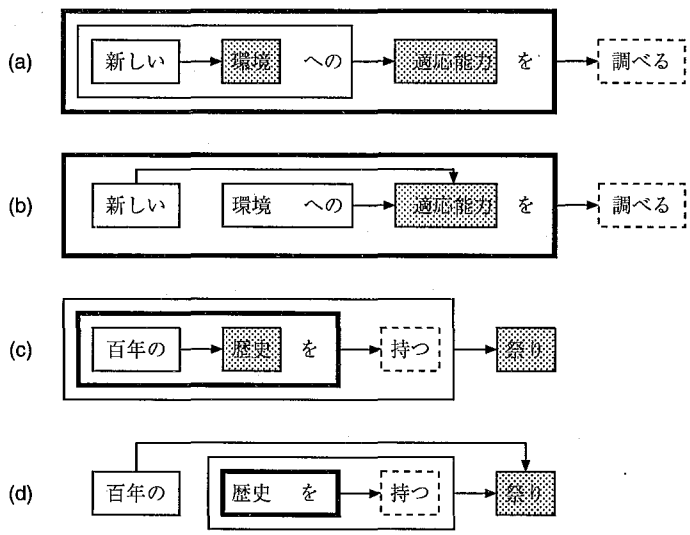


図 4.7: 連体修飾句の係り先に関する2種類の曖昧性

4.2.4　並列構造の曖昧性

並列構造の曖昧性 (二つの句が並列関係にあるか否か，あるいは対象としている句と並列関係にある句はどれか) の解消には意味情報が必要であり，構文解析の段階で解消することは困難である．予備実験によると，並列構造を含む文の解析精度は，含まない文の解析精度の半分程度しかない [43]．解析精度を全体的に上げるためには，並列構造の曖昧性の扱いについて検討する必要がある．構文解析器 KNP[32] 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解析するアプローチを採用している [30]．しかし，本研究では，並列関係にあるか否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，二つの句が並列関係にある場合とない場合 (つまり，係り受け関係にある場合) でコーパスに付与する構文構造は同一にし，抽出した文法は，二つの句が並列関係にあるか否かの曖昧性を区別しない．この変更は，Schiehlen[47] の "Coordinated Categories" と類似している．

日本語には，並列名詞句，並列述語句，並列助詞句の3種類がある[7]．これらの構造を以下の方針で変更する．

合は，図 4.7(b) だけが出力される．一方,「百年の歴史を持つ祭り」の場合は，図 4.7(c) と図 4.7(d) の両方が出力される．

[7]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 [30]．

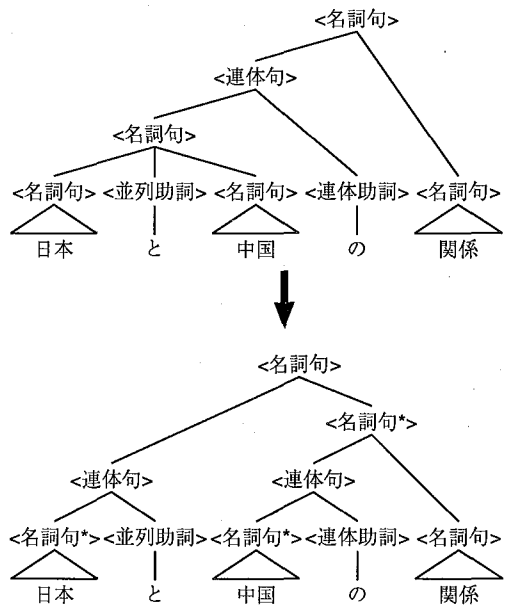


図 4.8: 並列名詞句の構造に関する変更

並列名詞句

例えば，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，各語の意味情報が必要となる．一般に,「AのBのC」と「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と類似している．そこで，構文解析の段階では「Aと」を連体修飾句と同様に扱い，並列関係にあるか否かの判定は，次の意味解析の段階で，連体修飾句の係り先の判定と同時に行うこととする (図 4.8).

並列述語句

予備実験によると，並列述語句を含む文の解析精度は，それ以外の並列構造を含む文の解析精度と比べて大幅に低くなる [43]．これは，二つの述語句が並列関係にあるか否かの判断が，他の並列構造に比べて難しいからである．例えば,「歌を歌い，踊りを踊る」という文において，二つの動詞句が並列関係にあるか係り受け関係にあるかは，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

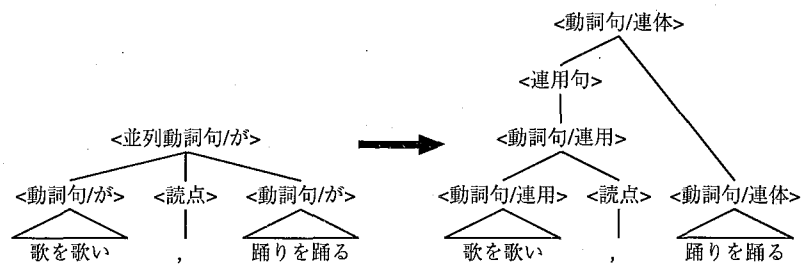


図 4.9: 並列述語句の構造に関する変更

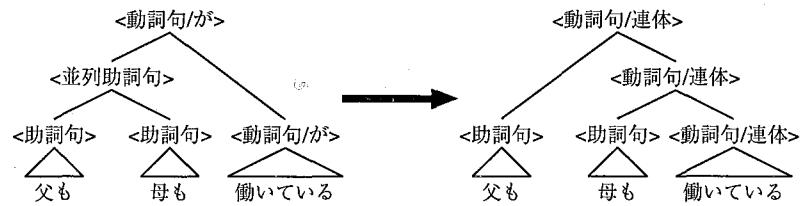


図 4.10: 並列助詞句の構造に関する変更

て判断が分かれる．そこで，並列名詞句の場合と同様に，述語句の並列構造と係り受け構造は同一にする．つまり，二つの述語句が並列関係にある場合でも，前の述語句が連用修飾句として後の述語句を修飾する構造で表現する (図 4.9)．

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く，並列助詞句を含む文の解析精度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，予備実験によると，並列助詞句を含む文の解析精度は並列述語句を含む文よりは高いが，並列名詞句を含む文とほぼ同じであった [43]．この要因として二つ挙げられる．一つは，並列助詞句を含む文が少なく，十分な学習ができていないからである．もう一つは,「1時 に 東京 に 到着する」のように，同じ助詞を含んでいても並列関係にない例も多く，助詞の情報だけでは判断できないからである．二つの助詞句が並列関係にあるか否かの判定には意味情報が必要であり，構文解析の段階での解決は困難である．そこで，コーパスには，並列関係にある助詞句は別個に動詞を修飾する構造を付与し (図 4.10)，抽出した文法は二つの助詞句が並列関係にあるか否かの区別をしないこととする．

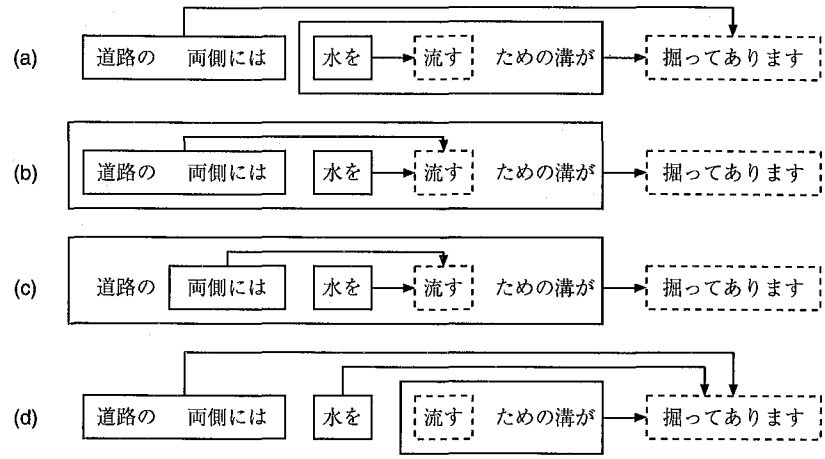


図 4.11: 構文解析の段階で生成される構造

4.3　変更方針のまとめ

以上をまとめると，コーパスの変更方針は以下のようになる．

(1) 用言の活用形に関する情報を上位ノードに追加する．

(2) 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で表現する．

(3) 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，意味によって構造を区別する．ただし，用言のとる表層格の情報は無視する．

(4) 二つの句が並列関係にあるか否かで構造上の区別はしない．

以上の方針に従ってコーパスを変更し，抽出した文法を利用して「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 4.11 に示す 4 個の構文構造が出力される (この方針が想定する正しい構造は (a) である)．ただし，実線で囲まれた句は連用修飾句を，破線で囲まれた部分は用言を，矢印は連用修飾関係を表す．これら 4 個の構文構造は，連用修飾句の範囲と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，各構文構造が包含する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．

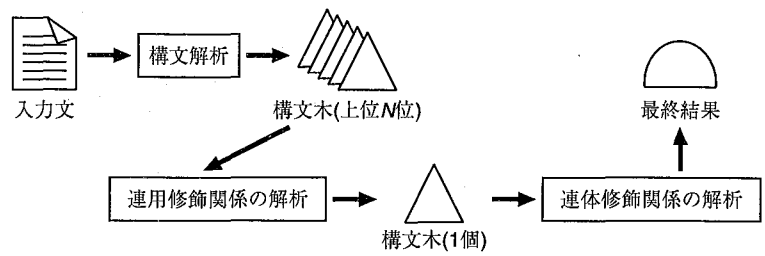


図 4.12: 構文解析後の意味解析の流れ

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」に係るかを判定し，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」に係るかを判定する．一方，構文構造 (b) では，連体修飾句「道路の」が「ため」や「溝」に係る可能性は，動詞「流す」を修飾する連用修飾句「道路の両側には」の範囲を変えることになるので，考慮する必要はない．

このコーパス作成方針では，曖昧性を連用修飾関係と連体修飾関係に大きく分けて扱っている．これは，日本語の処理において，連用修飾関係の解析で使用する手法と連体修飾関係の解析で使用する手法は異なると考えているからである．連用修飾関係の解析は，用言やそれに付属する助動詞を中心に，格や呼応関係などを考慮しながら進めていく．一方，連体修飾関係の解析は，体言を中心に進めていくことになる．本研究で提案する方針で変更した (あるいは新しく作成した) コーパスから抽出した文法を利用して構文解析を行った場合，次の意味解析では，図 4.12 に示すように，まず，用言を中心に連用修飾関係を決定し (つまり，出力された解析木の中から一つを選択し)，その後，体言を中心に連体修飾関係を決定することになる (つまり，選択した解析木が包含する意味的曖昧性を解消する)．

第5章　評価実験

第4章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利用して，以下の2点について評価を行った．

(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) どの程度の解析精度が得られるか

評価実験(1)は本研究の目的そのものであるが，曖昧性だけでなく，得られた構文解析結果の精度も重要な要素であるので，評価実験(2)も行う．

第5.1節では，変更前および変更後コーパスそれぞれから抽出した文法について，上述の2点を評価した結果を示す．ただし，評価実験(2)では，文法抽出に使用したコーパスに付与されている構造を正解データとして，解析精度の評価を行っている．第5.2節では，変更後コーパスから抽出した文法について，文節の係り受け精度を評価する．ただし，ここでは，変更前コーパスに付与されている構造を正解データとして評価を行っている．

5.1　曖昧性と文正解率に関する評価

5.1.1　EDRコーパスによる評価

まず，付録Aで述べたコーパス8,911文(1文あたり平均20.01形態素，最短5形態素，最長63形態素)に対し，第4.2節で述べた方針に従って，構文構造付きコーパス作成支援ツール[44]で構文構造を変更した[1]．図5.1に，使用するコーパス8,911文の1文あたり形態素数の分布を示す(縦軸はコーパス全体に占める文数の割合を表す)．構文構造の変更は，以下の手順で行っている．

(1) 変更方針に従い，変更前のコーパスから抽出した文法を変更

[1]変更前のコーパスは約2万文あるが，8,911文しか変更していないため，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．

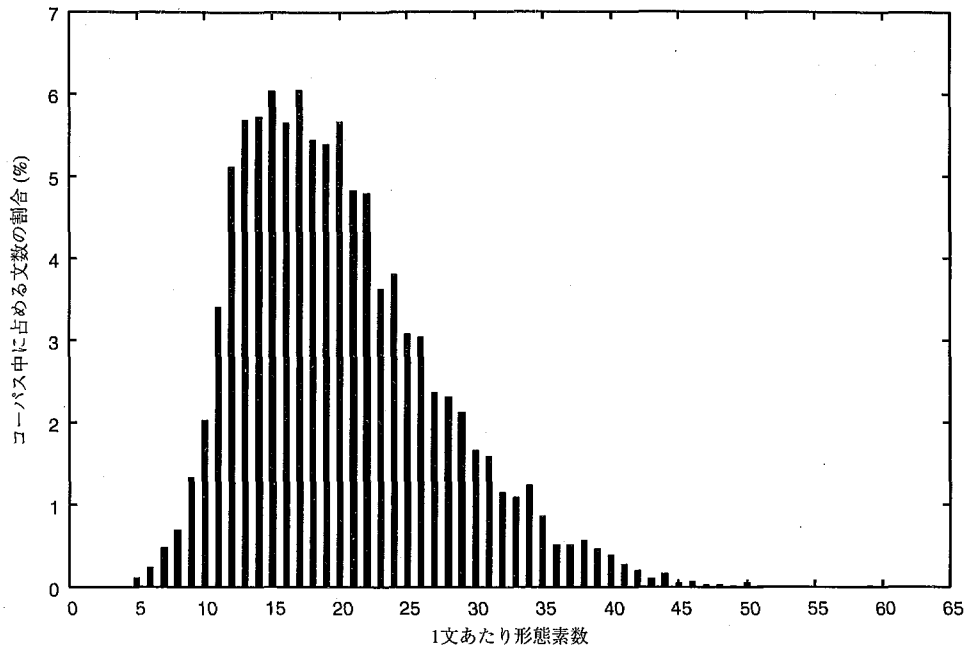


図 5.1: 使用するコーパスの1文あたり形態素数の分布

(2) MSLRパーザ[49]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[51])を獲得.

(3) コーパス作成支援ツールで，統語圧縮共有森から正しい解析木を決定.

手順(3)で使用するコーパス作成支援ツールは，解析結果を一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絞り込むためのものである．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．

こうして作成した変更前，変更後のコーパス全8,911文からそれぞれ文法を抽出し[2](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$g_{\mathrm{all}}^{\mathrm{edr}}$，$G_{\mathrm{all}}^{\mathrm{edr}}$と表記する)，MSLRパーザで同じ8,911文を構文解析した[3]．結果を表5.1に示す．本研究で提案した変更方針により，文法規則数は約250個増加しているが，構文

[2]Charniak[7]のように出現頻度の低い規則を削除せず，全規則を利用する.

[3]MSLRパーザは形態素解析と構文解析を同時に行えるが，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.

表 5.1: 文法 $g_{\mathrm{all}}^{\mathrm{edr}}$, $G_{\mathrm{all}}^{\mathrm{edr}}$ 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
$g_{\mathrm{all}}^{\mathrm{edr}}$	1,694	249	60	1.868×10^{12}
$G_{\mathrm{all}}^{\mathrm{edr}}$	1,949	279	600	9.355×10^{5}

解析結果の数は 10^{12} オーダから 10^{5} オーダに減少した[4].

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 [48]．文法抽出に使用した文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大きく減少している主な要因として，以下の 3 点が挙げられる．

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4.1節で挙げた曖昧性を増大させる要因の3番目にあたるが，用言の活用形だけでなく，後置詞句 (助詞句) でも同様の問題が起こり得る．EDR コーパスでは,「が」,「を」などと「の」を区別せず，すべて「助詞」としているので，白井らの手法では，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を区別できない．本研究で使用しているコーパスでは,「東京へ行く」のように連用修飾句になる場合は「助詞句」とし,「東京の 人口」のように連体修飾句になる場合は「連体句」として区別しているので，このような曖昧性は出ない[5].

品詞の細分化: 本研究で使用し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している．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が可能な名詞を他の名詞と区別しておかなければ，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[6].

[4]今回の方針では品詞レベルの変更を考慮しておらず，非終端記号の数に変化はない．

[5]「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 中心に発達する」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 で扱うことを考えず，このような文はコーパス作成の対象外とし，曖昧性を除外している．

[6]日本語では助詞が頻繁に省略され,「太郎, 東京へ行く」のように，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，このような助詞の省略はコーパス作成の対象外としている．

表 5.2: 文法 $g^{\mathrm{edr}}_{\mathrm{train}}$, $G^{\mathrm{edr}}_{\mathrm{train}}$ の被覆率と再現率

	被覆率	再現率
$g^{\mathrm{edr}}_{\mathrm{train}}$	98.51%	96.63%
$G^{\mathrm{edr}}_{\mathrm{train}}$	97.32%	95.88%

連体修飾関係と並列関係: 本研究で提案する変更方針では，連体修飾句の係り先の曖昧性と二つの句が並列関係にあるか否かの曖昧性は，同一の構造で表現するようにしている．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，先に挙げた二つの要因ほど，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．

次に，構文解析結果を確率一般化LR(PGLR)モデル [21] でランク付けし，解析精度を調べた[7]．ただし，8,911文を10分割し，一つを評価用，残りをPGLRモデルの学習用とする10分割交差検定により評価を行った．文法は，全8,911文から抽出したもの ($g^{\mathrm{edr}}_{\mathrm{all}}$, $G^{\mathrm{edr}}_{\mathrm{all}}$) と，PGLRモデルの学習用データのみから抽出したもの ($g^{\mathrm{edr}}_{\mathrm{train}}$, $G^{\mathrm{edr}}_{\mathrm{train}}$) の2通りを用意した．この二つの文法それぞれによる解析結果の上位1位から100位以内についての文正解率を図5.2と図5.3に，文法 $g^{\mathrm{edr}}_{\mathrm{train}}$, $G^{\mathrm{edr}}_{\mathrm{train}}$ の被覆率と再現率を表5.2に示す．ただし，文正解率，被覆率，再現率は以下のように定義する．

$$\text{文正解率} = \frac{\text{上位}\,n\,\text{個の解析木の集合の中に正しい木が含まれる文の数}}{\text{解析した文の総数}} \tag{5.1}$$

$$\text{被覆率} = 1 - \frac{\text{解析に失敗した文の総数}}{\text{解析した文の総数}} \tag{5.2}$$

$$\text{再現率} = \frac{\text{全解析木の集合の中に正しい木が含まれる文の数}}{\text{解析した文の総数}} \tag{5.3}$$

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，本研究では，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かどうかが重要となる．したがって，構文構造の部分的な正しさを示

[7] Charniakや白井らはPCFGを用いているが [7, 48]，本研究ではPGLRモデルを用いる．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 [41].

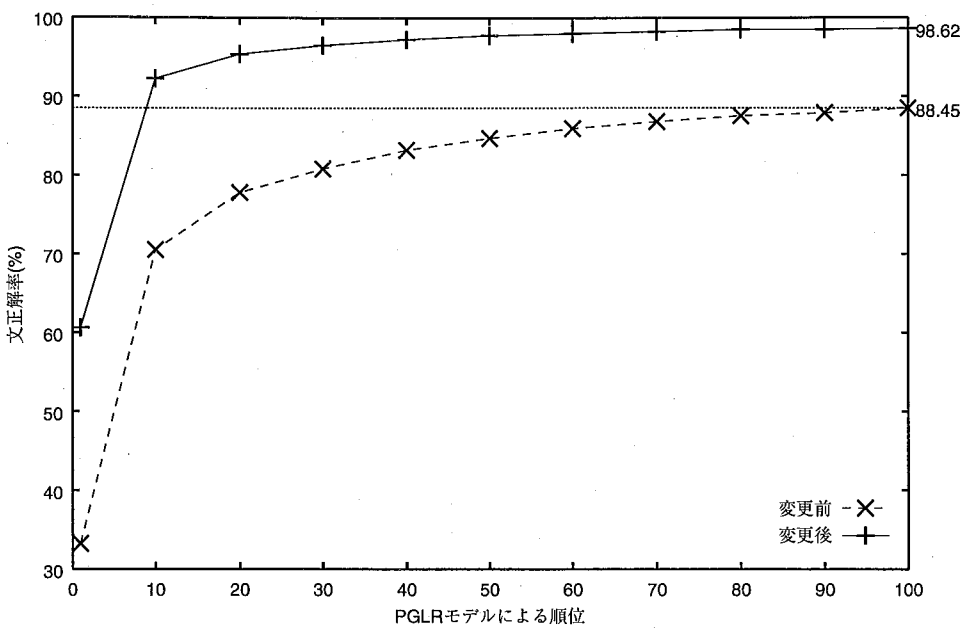


図 5.2: 文法 $g^{\mathrm{edr}}_{\mathrm{all}}$, $G^{\mathrm{edr}}_{\mathrm{all}}$ による構文解析結果の文正解率

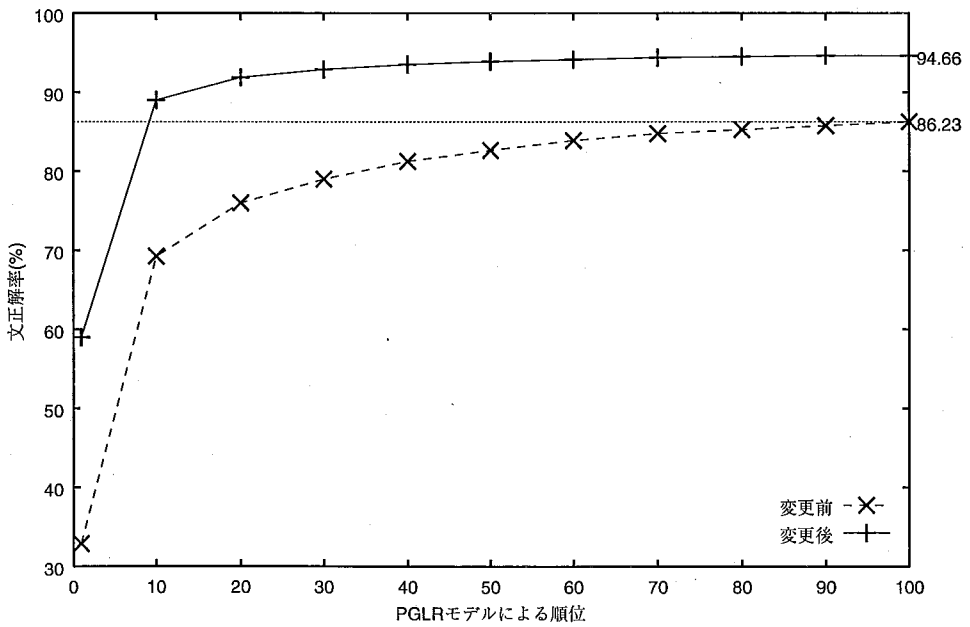


図 5.3: 文法 $g^{\mathrm{edr}}_{\mathrm{train}}$, $G^{\mathrm{edr}}_{\mathrm{train}}$ による構文解析結果の文正解率

す括弧付けの再現率や適合率よりも，上述の文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．

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，文法 $g^{\mathrm{edr}}_{\mathrm{all}}$，$G^{\mathrm{edr}}_{\mathrm{all}}$ の文正解率はそれぞれ88.45%，98.62%，文法 $g^{\mathrm{edr}}_{\mathrm{train}}$，$G^{\mathrm{edr}}_{\mathrm{train}}$ の文正解率はそれぞれ86.23%，94.66%となり，変更後のコーパスから抽出した文法の方が8〜10%高くなっている．また，変更後のコーパスから抽出した文法で，変更前のコーパスから抽出した文法による上位100位以内の文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，本研究で提案するコーパス変更方針が有効であることが分かる．

表5.2より，文法 $g^{\mathrm{edr}}_{\mathrm{train}}$，$G^{\mathrm{edr}}_{\mathrm{train}}$ 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，コーパスの変更によって1%程度低下し，解析不能な文が変更前に比べて1%程度増加する．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 $g^{\mathrm{edr}}_{\mathrm{train}}$，$G^{\mathrm{edr}}_{\mathrm{train}}$ による上位100位以内の文正解率の差が，文法 $g^{\mathrm{edr}}_{\mathrm{all}}$，$G^{\mathrm{edr}}_{\mathrm{all}}$ 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，文正解率が10%近く上昇することから，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[8]．

5.1.2　RWCコーパスによる評価

本研究では，コーパス変更方針を検討するためにEDRコーパスを使用してきた．しかし，このコーパスから抽出した文法を使用して構文解析を行うことには，問題がある．それは，このコーパスで使用している品詞体系に基づく形態素解析器が公開されていないことである．MSLRパーザは形態素解析と構文解析を同時に行うことができるが，形態素解析部では単に辞書引きを行っているだけであり，品詞間の連接制約(二つの品詞が連接するか否かの制約)しか扱えない．一方，一般に広く使われている日本語の形態素解析器に茶筌[38]があり，この品詞体系(IPA品詞体系)に基づいて作成された品詞タグ付きコーパスとしてRWCコーパス[19]がある．そこで，RWCコーパス中の16,421文(1文あたり平均21.71形態素，最短5形態素，最長49形態素[9])で，前節と同様の実験を行った．図5.4に，1文あた

[8]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．

[9]構文構造を付与する際，1文あたりの形態素数が5未満の文と50以上の文を除外した．短い文は，「6 3歳.」，「小沢征爾指揮.」，「パパパパパーン.」など，構文構造を付与する対象として相応しくないものが多いからであり，長い文は，作業者の負担が大きくなり，誤った構造を付けてし

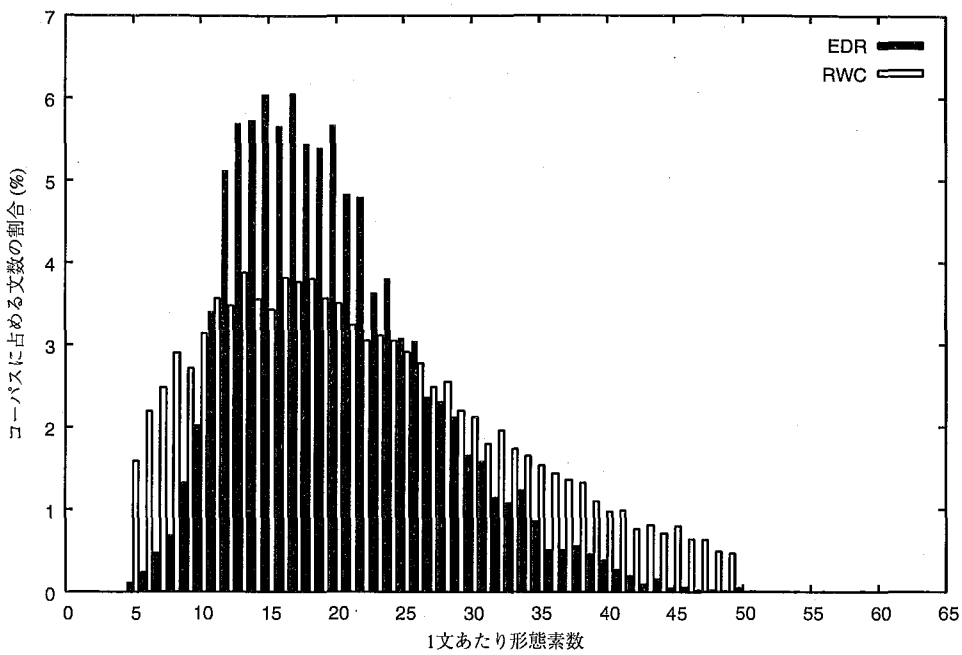


図 5.4: 使用するコーパスの1文あたり形態素数の分布

り形態素数の分布を示す (縦軸はコーパス全体に占める文数の割合を表す). 参考のため, EDR コーパス 8,911 文の 1 文あたり形態素数の分布を併記して示す. 平均形態素数は EDR コーパス 8,911 文と大差ないが, EDR コーパスでは, 1 文あたり 15～20 形態素の文の割合が, それ以外の文の割合に比べて高いのに対し, RWC コーパス 16,421 文では, その差が小さく, 偏りが小さいことが分かる.

先に述べたように, RWC コーパスは品詞タグ付きコーパスであり, 構文構造を付与されていないため, 第 1 章で述べたコーパス作成手順 (作成方針の検討手順) のように, 構文構造付きコーパスを出発点とすることはできない. そこで, 今回は, EDR コーパスを対象に検討した方針 (本研究で提案した方針) をそのまま RWC コーパスに適用することで, コーパス作成手順 (4) から開始し, 直接「変更後コーパス」を作成した (構文構造の付与に関する詳細は付録 B で述べる). この評価実験により, 本研究で提案したコーパス作成方針が, 方針の検討の際に利用したコーパスとは異なるコーパスに対しても適用可能であるかどうかも確認できる.

評価は EDR コーパスの場合と同様の手順で行った. まず, 全 16,421 文から文

まう可能性が高くなるためである. 長い文は数多くの重要な情報を持つので, 将来的には構文構造を付与する予定である.

表 5.3: RWC コーパスから抽出した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
$G_{\mathrm{all}}^{\mathrm{edr}}$	1,949	279	600	9.355×10^5
$G_{\mathrm{all}}^{\mathrm{rwc}}$	2,565	290	391	9.599×10^4

表 5.4: 文法 $G_{\mathrm{train}}^{\mathrm{edr}}$, $G_{\mathrm{train}}^{\mathrm{rwc}}$ の被覆率と再現率

	被覆率	再現率
$G_{\mathrm{train}}^{\mathrm{edr}}$	97.32%	95.88%
$G_{\mathrm{train}}^{\mathrm{rwc}}$	98.38%	97.18%

法 $G_{\mathrm{all}}^{\mathrm{rwc}}$ を抽出し(「変更前コーパス」を作成していないので，文法 $g_{\mathrm{all}}^{\mathrm{rwc}}$ は存在しない)，同じ16,421文の品詞列を構文解析した結果を表5.3に示す．参考のため，文法 $G_{\mathrm{all}}^{\mathrm{edr}}$ による結果も併記しておく．データが異なるため厳密な比較はできないが，EDR コーパスの場合と同様，文法 $G_{\mathrm{all}}^{\mathrm{rwc}}$ でも曖昧性は十分抑えられていることが分かる．

EDR コーパスに対する構文構造の付与と RWC コーパスに対する構文構造の付与における大きな差として以下の2点がある．

品詞体系: EDR コーパスでは，EDR 日本語単語辞書中の品詞名，左右連接属性，動詞が取る表層格情報などを組み合わせて細分化しているが，RWC コーパスは茶筌の品詞体系を基本としている．茶筌の品詞体系の方が分類は若干粗いが，曖昧性は十分抑えられている．

括弧の扱い: EDR コーパスに構文構造を付与する際，括弧を含む文は除外していた．しかし，RWC コーパスに構文構造を付与する際には，括弧を含む文も扱うようにしている．文法規則数，非終端記号数が増加している要因は，括弧を扱うための規則，非終端記号が新たに追加されたからである．

次に，10分割交差検定による文法 $G_{\mathrm{all}}^{\mathrm{rwc}}$ と $G_{\mathrm{train}}^{\mathrm{rwc}}$ の文正解率，被覆率，再現率を図5.5，図5.6，表5.4に示す．参考のため，文法 $G_{\mathrm{all}}^{\mathrm{edr}}$, $G_{\mathrm{train}}^{\mathrm{edr}}$ による結果も併記しておく．結果より，文正解率，被覆率，再現率についても，EDR コーパスの場合と大差のない結果が得られていることが分かる．

本実験により，本研究で提案したコーパス作成(変更)方針が，他のコーパスに対しても適用可能であることが確認できた．これは，コーパスごとに一から作成

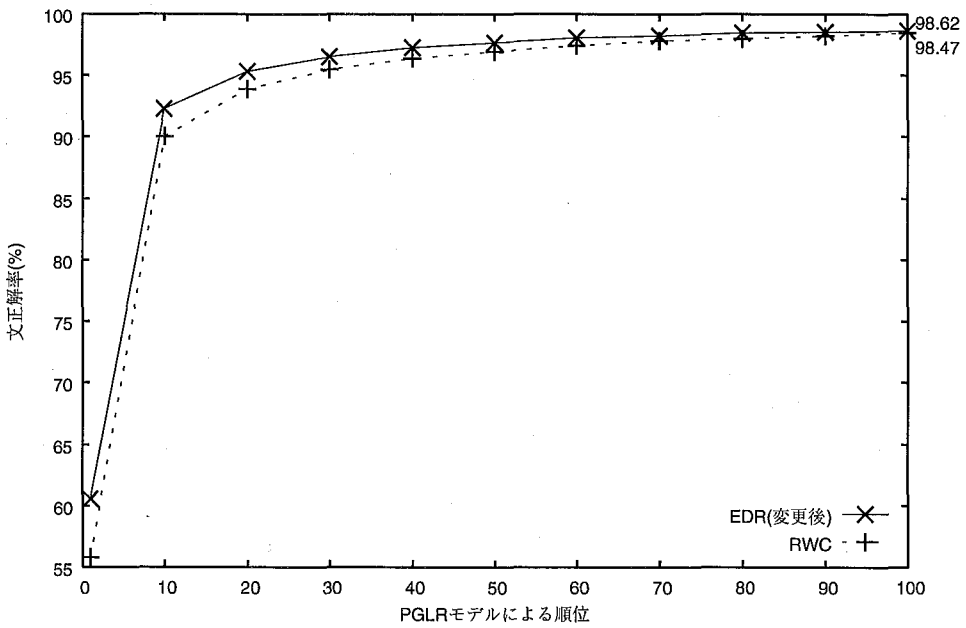


図 5.5: 文法 $G^{\mathrm{edr}}_{\mathrm{all}}$, $G^{\mathrm{rwc}}_{\mathrm{all}}$ による構文解析結果の文正解率

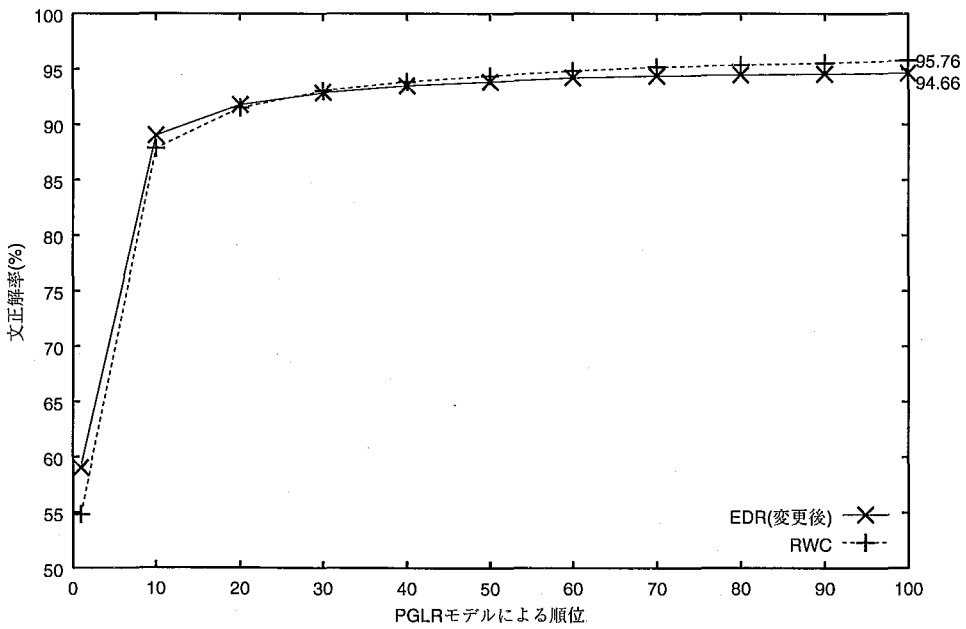


図 5.6: 文法 $G^{\mathrm{edr}}_{\mathrm{train}}$, $G^{\mathrm{rwc}}_{\mathrm{train}}$ による構文解析結果の文正解率

方針を検討する必要はなく，まず，同じ方針を適用することで，コーパス作成を容易にできることを示している．

5.2　文節係り受け精度に関する評価

第5.1節では，本研究で提案した方針により作成したコーパスから抽出した文法が，構文解析結果の曖昧性を抑え，文正解率が約10%上昇することを示した．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て，文節係り受け精度を調べた．

5.2.1　句構造からの文節係り受け関係の抽出

コーパスに付与されている構文構造は句構造であり，文節の係り受け精度を調べるためには，句構造から文節の係り受け関係を抽出しなければならない．その手順を以下に示す．

(1) 文節区切りを決定する．

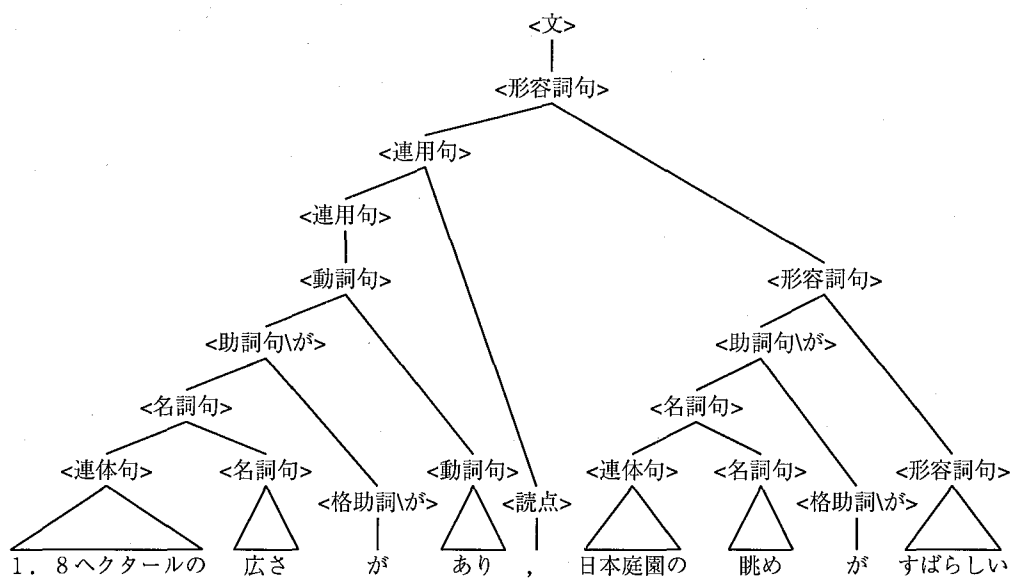
(2) 構文構造(句構造)をもとに，各文節について，係り先となる文節を決定する．

例えば，図5.7に示すように，上記の句構造からは，下記の文節の係り受け関係が抽出される．句構造は，係り先の文節の中のどの語に係るかを厳密に決定することも可能であるが，今回の実験では，どの文節に係るかのみを調べる．

変更後のコーパスでは，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，同一の構造で表現している．そのため，PGLRモデルによる生成確率1位の構文木中に連体修飾句が存在する場合は，その係り先を決定しなければならない．今回の実験では，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と仮定する[10]．

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．変更後のコーパスから抽出した文法

[10]「N_1 の N_2 の N_3」という名詞句について，連体修飾句「N_1 の」が名詞「N_2」に係るとすると，その正解率は72.5%であった[39]．



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)

図 5.7: 句構造からの文節係り受け関係の抽出の例

でこの文を解析すると，この助詞句について，図5.8(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) [11]．図5.8(a) のような解析木が出力された場合，その係り受けは，図5.8(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5.8(c)，(e) の場合は，それぞれ図5.8(d)，(f) に示すような係り受け関係となる．

[11]実際には，助詞句「青い目のアメリカから」が動詞「会う」に係る構造も出力されるが，ここの議論では省略する．

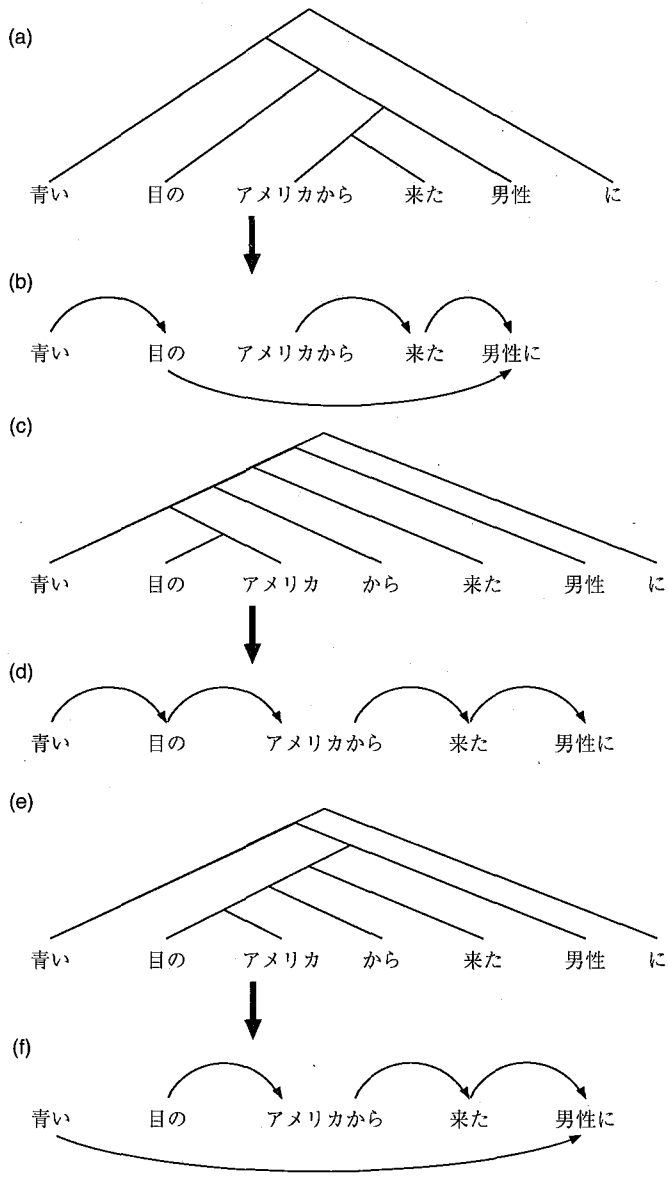


図 5.8: 連体修飾句の係り先の決定

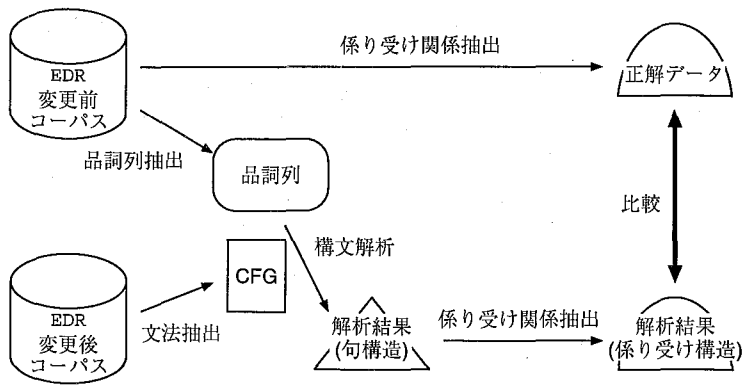


図 5.9: EDR 変更前コーパスを正解データとした場合の係り受け構造の比較

連用修飾句関係の構造は意味によって区別しているので，PGLR モデルによる生成確率が1位の構文解析結果の構造をそのまま利用する．

複合名詞の内部の構造は語構成に関係なく同一としているが，今回の実験では文節の係り受け関係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しない．

二つの文節が並列関係にあるか否かの曖昧性は，今回の実験では無視し，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．

5.2.2　EDR 変更前コーパスを正解データとした場合

まず，EDR 変更後コーパスから抽出した文法で構文解析し，PGLR モデルによる生成確率が1位の解析木から抽出される文節の係り受け関係と，EDR 変更前コーパスに付与されている構文構造から抽出される文節の係り受け関係を比較し，係り受け精度を計算する (図5.9)．評価はランダムに選択した100文 (1文あたり平均19.84形態素，7.16文節) [12]で行い，残りを PGLR モデルの学習に利用する．精度は以下の3つの尺度で評価した．

$$\text{係り受け A 型} = \frac{\text{正しい係り受け関係の数}}{\text{総文節数} - \text{評価用データの文数} \times 1} \tag{5.4}$$

$$\text{係り受け B 型} = \frac{\text{正しい係り受け関係の数} - \text{評価用データの文数} \times 1}{\text{総文節数} - \text{評価用データの文数} \times 2} \tag{5.5}$$

[12]並列構造を含む文は17文存在する．

表 5.5: EDR 変更前コーパスを正解データとした場合の係り受け精度

	係り受けA型	係り受けB型	文正解率	文節不一致
本実験	91.32%	89.61%	61.54%	9
KNP	89.97%	—	—	—
SVM	89.29%	—	47.53%	—
ME	87.93%	—	43.58%	—

$$\text{文正解率} = \frac{\text{全ての係り受け関係を正しく決定できた文の数}}{\text{評価用データの文数}} \tag{5.6}$$

「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文節間の係り受け関係以外の係り受け関係の正解率である．「文正解率」とは，全ての文節の係り受け関係が正しい文の割合である．表5.5にその結果を示す．ただし，「文節不一致」は文節区切りが正解データと一致しなかった文の数を表し，精度の計算は，文節区切りが正解データと一致した文のみを対象としている．参考として，KNP[32]，Support Vector Machine[29]，最大エントロピー法[53]を用いた係り受け解析の結果を併記しておく[13]．

結果より，意味的情報を利用しなくとも，PGLRモデルのみによる係り受け精度が90%前後と非常に高いことが分かる．使用しているコーパスや実験の条件が異なるため，公平な比較にはならないが，この結果は，KNPやSupport Vector Machine，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．本研究では，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．予備実験として，本格的な意味解析の代わりに，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(SDSA)[24]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を行ったところ，非常に単純なスコア付けの手法であるにも関わらず，93.0%の係り受け精度(係り受けB型)，68.8%の文正解率が得られている[55]．

本実験で使用した評価用データは100文であるが，この規模は非常に小さく，SDSAの有効性を示すには至っていない．しかし，実験結果から，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．

[13]KNPによる結果は，内元らの実験結果[52]を引用した．

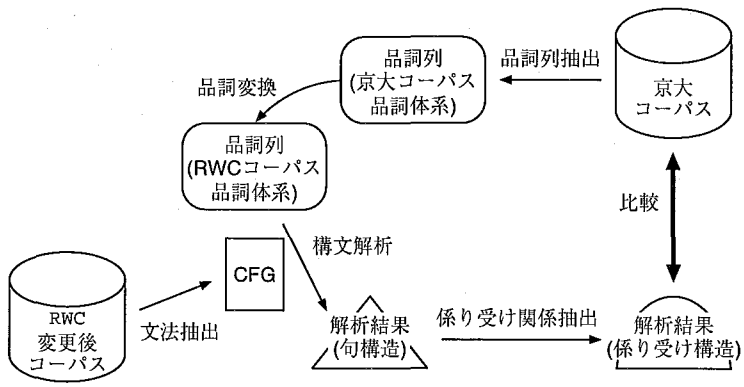


図 5.10: 京大コーパスを正解データとした場合の係り受け構造の比較

5.2.3　京大コーパスを正解データとした場合

本研究では，RWC コーパスに対して「変更後」の構文構造を直接付与した．そのため，EDR コーパスの場合と同様に文節係り受け精度を求めることができない．そこで，京大コーパス中の文を構文解析し，京大コーパス中の係り受け構造を正解データとして評価を行った．その手順は，以下のようになる (図 5.10)．

(1) 京大コーパスの品詞体系を RWC コーパスの品詞体系に自動変換

(2) RWC コーパスから抽出した文法で構文解析

(3) PGLR モデルによる生成確率 1 位の解析結果から文節係り受け関係を抽出

(4) 京大コーパス中の文節係り受け関係と比較し，精度を計算

京大コーパスの品詞体系は形態素解析器 JUMAN[33] に基づいており，RWC コーパスの品詞体系とは異なるため，品詞体系の変換が必要となる．本研究では，約 3,000 個の規則を利用し，京大コーパスの品詞体系を RWC コーパスの品詞体系に自動的に変換している．品詞体系を変換した後の処理は，EDR コーパスを利用した場合の評価実験と同様である．文法は，RWC 変更後コーパス全 16,421 文から抽出し，PGLR モデルの学習も同じ 16,421 文で行った．評価用データとして，京大コーパス中の毎日新聞記事 1995 年 1 月 1 日分から 9 日分までの 8,835 文を使用した．

RWC 変更後コーパスから抽出した文法で京大コーパスの文 (品詞列) を解析する際，品詞体系の変換によって単語区切りが変わる箇所があるという問題が発生

表 5.6: 評価用データの文数

毎日新聞 1995 年 1 月 1～9 日記事	8,835 文
文節区切りと単語区切りの不一致を除外	7,601 文
RWC 変更後コーパスに出現しない品詞を含む文を除外	5,837 文
1 個以上の解析結果が出力された文	3,764 文

表 5.7: 京大コーパスを正解データとした場合の係り受け精度

	係り受け A 型	係り受け B 型	文正解率	文節不一致
直近	85.76%	82.88%	52.38%	1,280
最適	88.35%	86.00%	58.82%	1,280
KNP	89.97%	—	—	—
SVM	89.29%	—	47.53%	—
ME	87.93%	—	43.58%	—

する．例えば,「今秋には議長国としてアジア・太平洋経済協力会議に臨まなければならない」という文で,「として」は，京大コーパスは助詞「と」と動詞「して」に分かれるが，RWC コーパスでは 1 語の連語の助詞になる．京大コーパスに付与されている文節係り受け構造では,「と」と「して」の間に文節区切りが入り，文節「議長国と」が文節「して」に係るとしているが，RWC 変更後コーパスでは「と」と「して」の間に単語区切りがないため，文節区切りを入れることはできない．今回の実験では，京大コーパスにおける文節区切りの位置に，RWC 変更後コーパスにおける単語区切りがない文は，評価対象から除外する．その結果，評価用データの文数は 7,601 文となった (表 5.6)．

さらに，RWC 変更後コーパスに出現しない品詞を含む文は，構文解析しても解析結果が得られないため，評価対象から除外する．その結果，評価用データの文数は 5,837 文となった (表 5.6)．この 5,837 文を構文解析したところ，1 個以上の解析結果を出力した文は 3,764 文あり (1 文あたり平均 19.33 形態素，7.50 文節)，この 3,764 文の解析結果の PGLR モデルによる生成確率が 1 位であった解析木の文節係り受け精度を評価した．評価用データの文数が 8,835 文から 3,764 文へ大幅に減少しているが，その要因については付録 C.1 で述べる．

結果を表 5.7 の「直近」の行に示す．EDR コーパスを正解データとした場合の結果と比べて精度が下がっているが，特に文節区切りが一致しなかった文が多い．その要因については付録 C.2 で述べるが，それでも，意味情報や語彙情報などを

表 5.8: 係り受けの種類別の精度

		正解	不正解	合計	精度
連体	並列名詞句	406	181	587	69.17%
	その他	4,169	288	4,457	93.54%
	計	4,575	469	5,044	90.70%
連用	助詞句	5,996	1,120	7,116	84.26%
	述語句	1,032	277	1,309	78.84%
	副詞句	759	221	980	77.45%
	引用句	308	17	325	94.77%
	計	8,095	1,635	9,730	83.20%
合計		12,670	2,104	14,774	85.76%

一切使用しない段階での結果としては十分高い係り受け精度が得られていると考えられる．

表5.8に，係り受けの種類別の係り受け精度(A型)を示す．係り受けの種類は，解析結果から係り受け関係を抽出する際に，非終端記号の情報をもとにして決定している．これより，全体的に連体修飾関係の方が精度が高いが，並列名詞句，副詞句(副詞，接続詞)，述語句(従属節，並列述語句)の精度が極端に低いことが分かる．意味解析では，これらの精度の向上が全体の精度の向上につながると考えられる．

今回の実験では，連体修飾句の係り先を，係り得る名詞の中で最も近いものを含む文節としている．意味的情報を利用して連体修飾関係の曖昧性の解消を行い，最適な結果が得られたと仮定した場合の結果を表5.7の「最適」の行に示す．ここでは，解析結果の構造が包含する連体修飾関係の曖昧性の中に，正解と一致するものがあれば正解とし，連用修飾関係の誤りが原因で，解析結果が包含する曖昧性の中に，正解と一致するものが存在しなければ，不正解としている．この結果は，PGLRモデルによる生成確率1位の解析木について連体修飾関係の曖昧性解消を行った場合の係り受け精度の上限を表す．結果より，連体修飾関係の曖昧性解消によって，係り受け精度は最大3%前後，文正解率は最大約6.4%向上する．しかし，最適な連体修飾先を決定できたとしても，係り受け精度は，他の日本語係り受け解析手法[32, 29, 53]に及ばない．さらに精度を向上させるためには，連用修飾関係の精度の向上が必要である．

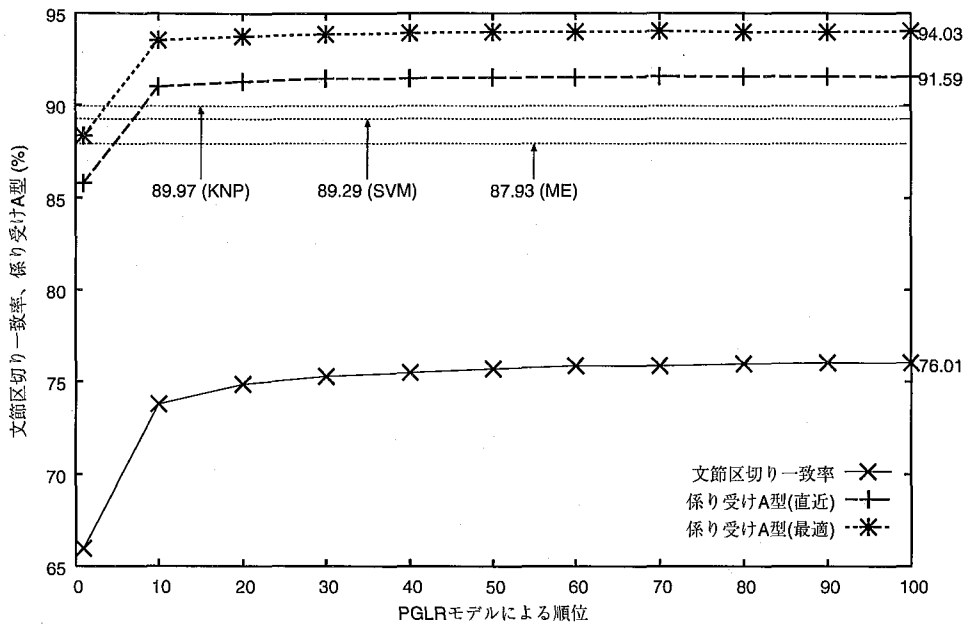


図 5.11: 連用修飾関係の解析が最適に行われた場合

第 4.3 節で述べたように，連用修飾関係の曖昧性の解消は，PGLR モデルによる生成確率の上位 N 個の解析木から一つを選択することで行うことを想定している．そこで，PGLR モデルによる生成確率の上位 N 個の解析木について連用修飾関係の解析を行い，最適な解析木を選択できた場合の係り受け精度 (A 型) と文節区切り一致率 (1 文中の全ての文節区切りが正解データと一致する文の割合) を調べた．結果を図 5.11 に示す．これより，PGLR モデルによる生成確率の上位 100 個の解析木について連用修飾関係と連体修飾関係の曖昧性解消を行った場合，最大で 94.03%まで係り受け精度が向上することが期待できる．

第6章　考察

第5章では，本研究で提案した方針に従って変更(作成)したコーパスから抽出した文法の評価を行い，構文解析結果の曖昧性が十分抑えられ，解析精度も高いことを示した．本章では，RWC変更後コーパスから抽出した文法やコーパス変更(作成)方針について考察し，コーパスへの新しい文の追加やコーパス変更方針の再検討など，今後のコーパスと文法の開発の際に留意すべき点を述べる．

6.1　コーパス中に出現しなかった言語現象に関する考察

現在，構文構造付きコーパスの文数は，RWC変更後コーパスで16,421文であるが，この文数で全ての言語現象を網羅できているとは言えない．コーパスから抽出した文法の被覆率をさらに向上させるためには，コーパス中に出現しなかった言語現象を含む文を追加する必要がある．本節では，コーパス中に出現しなかった言語現象について考察する．この結果は，コーパスに追加する文の選定の参考になる[1]．

付録Bで述べるように，コーパスの作成や変更は，コーパス作成支援ツールを使用して行っている．このツールは，MSLRパーザによる解析結果(統語圧縮共有森)の中から正しい構造を持つ解析木を決定するためのものである．このツールを使用するためには，MSLRパーザで構文解析を行うための文法(コーパス作成用文法)を人手で用意しなければならない．このコーパス作成用文法は，存在すると感じた言語現象を表現する文法規則をすべて列挙しただけのものであり，実際には存在しない言語現象を表現する規則も混在している可能性がある．そのような規則が存在すると，コーパス作成時に誤ってその規則を使用する構文構造を正しいと判断してしまう可能性もあり，できる限り排除すべきである．一方，存在するにも関わらず，コーパス中にその言語現象を含む文が存在しなかったために，抽

[1]コーパスから統計情報を獲得する場合には無作為に文を選定する必要があるが，ここでは，多様な言語現象を網羅するために追加すべき文について考察する．

表 6.1: 文法規則数の分布

	A類	B類	C類	合計
規則数	2,565	1,008	4,161	7,734
割合	33.17%	13.03%	53.80%	100.00%

出した文法中にその言語現象を表現する規則が含まれない可能性もある．さらにコーパスに新しい文を追加すればそのような規則を網羅することができるが，無計画に文を追加することは効率が悪い．効率良く追加するためには，コーパスに出現しなかった言語現象を表現する文法規則のうち，どの規則が必要であるかを検討し，必要な規則であるならば，その規則に対応する言語現象を含む文を，他のコーパスから検索するか，新しく生成するかして，追加すべきである．以下では，コーパス作成用文法とRWC変更後コーパスから抽出した文法の差分を調査し，コーパス中に出現しなかった言語現象を考察する．

6.1.1　文法規則の分類とその分布

コーパス作成用文法を以下のように分類する．

- コーパス中に存在する言語現象を表現する規則 → A類
- コーパス中に存在しなかった言語現象を表現する規則
  - コーパス作成支援ツールへの入力データとなる統語圧縮共有森の中に含まれていた規則 → B類
  - まったく使用されなかった規則 → C類

考察すべき規則はB類とC類である．特に，B類の中に不必要な規則が存在すると，コーパス作成時の選択肢を無意味に増加させ，人手による作業効率を悪化させる要因となる．

表6.1に各種類の文法規則数の分布を示す．ただし，B類の規則の中には，コーパス作成方針に合わないためにラベル付け対象外とした文を解析した際に出力された統語圧縮共有森の中で使用されている規則も含む．表6.1より，実際にコーパス中に出現した文法規則は3分の1程度である一方，統語圧縮共有森にも含まれない規則が過半数に達することが分かる．

6.1.2　コーパス中に出現しなかった言語現象の特徴

コーパス中に出現しなかった言語現象を表現する規則を見てみると，以下の特徴を持つ規則がコーパス中に存在していないことが分かった．

- 右辺が1個の終端記号である規則
- 補足節に関する規則
- 係り受け関係を表す規則
- 用言以外が文末に現れる文に関する規則
- 特殊規則
- 右辺に読点を含む規則

これらについて，以下で詳細を述べる．

右辺が1個の終端記号である規則の調査

ラベル付けに使用する文法規則は以下の2種類に分類できる[2]．

(1) 右辺が1個の終端記号 (品詞) である規則

(2) 右辺が1個以上の非終端記号列である規則

右辺が1個の終端記号である規則は，コーパス作成用の文法中に637規則あり，そのうち，A類は392規則，B類は17規則，C類は228規則あった[3]．B類，C類の規則を見ると，動詞，形容詞，助動詞など活用形を持つ終端記号 (品詞) が右辺に現れるものが目立った．そこで，全637規則を活用形別に分類してみた．活用形別の文法規則数の分布を表6.2に示す．ただし，出現率はコーパス中に出現した規則数の割合であり，以下の式で求められる．

[2]コーパスに付与した構文構造の性質上，この2種類以外の文法規則は存在しない (付録A.1)．

[3]コーパスから抽出した文法中の終端記号数は391個であるが (第5.1.2節)，右辺は同じ終端記号でありながら左辺が異なる非終端記号である規則が存在するため，A類の規則数は392規則となっている．また，B類の規則が存在するのは，ラベル付け対象外とした文の統語圧縮共有森で使用された規則も集計しているからである．

表 6.2: 品詞の活用形別の分布

活用形	A類	B類	C類	合計	出現率
基本	100	6	30	136	73.53%
未然	34	0	25	59	57.63%
連用	76	4	36	116	65.52%
仮定	47	3	61	111	42.34%
命令	20	2	41	63	31.75%
意志	25	0	28	53	47.17%
体言接続	6	2	3	11	54.55%
ガル接続	5	0	2	7	71.43%
活用無し	79	0	2	81	97.53%
合計	392	17	228	637	61.54%

$$\text{出現率} = \frac{\text{A類の規則数}}{\text{A類の規則数} + \text{B類の規則数} + \text{C類の規則数}} \tag{6.1}$$

活用形を持たない終端記号を右辺に持つ規則の中でコーパス中に存在しなかったものは「(手伝って) ちょうだい」などの“名詞-動詞非自立的”と「超 (大きい)」,「真っ(白い)」などの“接頭辞-形容詞接続”の2品詞であった．一方，活用形を持つ終端記号を右辺に持つ規則の中でコーパス中に存在しなかったものは多い．特に，仮定形，命令形，意志形は過半数が出現していない．今後，コーパスに新しい文を追加する際には，これらの活用形を含む文を中心に追加する必要がある．

補足節に関する規則

右辺が1個以上の非終端記号列である規則を左辺の非終端記号で分類すると，最も多いのは補足節 (助詞句) を左辺に持つ規則であり，その数は2,204規則である (ただし，後述の特殊規則を除く)．補足節は，末尾に出現する助詞 (格助詞10種類と「など」,「まで」) によって13種類に細分化される．それぞれについてのA類，B類，C類の規則数を表6.3に示す．ただし,「*」は，12種類のいずれの助詞も出現せず，その他の助詞 (係助詞など) だけで構成される補足節を表す．表6.3より，全般的に補足節に関する規則がコーパス中に存在していないことが分かる．特に，

表 6.3: 補足節を左辺に持つ規則の分布

助詞	A類	B類	C類	合計	出現率
*	60	11	33	104	57.69%
が	39	14	155	208	18.75%
を	39	16	153	208	18.75%
に	86	20	98	204	42.16%
で	61	34	113	208	29.33%
と	38	19	143	200	19.00%
の	8	5	179	192	4.17%
から	49	15	136	200	24.50%
へ	14	5	173	192	7.29%
より	28	26	146	200	14.00%
にて	1	0	191	192	0.52%
など	16	6	26	48	33.33%
まで	17	2	29	48	35.42%
合計	456	173	1,575	2,204	20.69%

「へ」,「の」,「にて」を持つ補足節に関する規則が存在していない[4].

コーパス中に出現しなかった文法規則の特徴を詳細に調べるために，補足節をさらに細かく分類する．補足節は名詞句や動詞句などの句と助詞で構成される．句の種類で補足節を分類し，名詞句以外の句を持つ補足節の規則のみを集計した結果を表6.4に示す[5]．これより，名詞句以外の句を持つ補足節に関する規則がコーパス中に出現していないことが分かる．名詞句以外の句を持つ補足節を含む文の例を挙げる．

- 南アフリカの人種融和政権へ参加するかどうかが注目されていた
- 組み立て役のブロリンにどれだけいい形でボールが渡るかがカギを握る

[4]補足節における助詞「の」は連体助詞ではなく格助詞である．例えば,「鼻の長い象」の「鼻の」は補足節であり,「象の長い鼻」の「象の」は連体節である．

[5]用言に助詞が結合する場合，その助詞が格助詞である場合は補足節に，それ以外である場合は連用節になるように方針を定めている．この方針より，格助詞ではない「など」や「まで」を含む補足節，12種類のいずれの助詞も含まない補足節に関する規則の中に，用言を含むものは存在しない．表6.4で，この3つの補足節に関する規則がないのは，そのためである．

表 6.4: 補足節を左辺に持つ規則の分布 (名詞句以外)

助詞	A類	B類	C類	合計	出現率
*	0	0	0	0	—
が	11	10	85	104	10.58%
を	10	13	81	104	9.62%
に	27	15	62	104	25.96%
で	12	23	69	104	11.54%
と	8	13	83	104	7.69%
の	1	3	92	96	1.04%
から	11	12	81	104	10.58%
へ	1	5	90	96	1.04%
より	6	18	80	104	5.77%
にて	0	0	96	96	0.00%
など	0	0	0	0	—
まで	0	0	0	0	—
合計	87	110	819	1,016	8.56%

このように，用言の末尾に「か」や「かどうか」が出現する場合に，動詞句など，名詞句以外の句を持つ補足節に関する規則が使われる．

さらに，別の分類として，補足節中の句が括弧を含むか否かで2種類に分け，括弧を含む補足節に関する規則のみを集計した結果を表6.5に示す．これより，括弧を含む規則もコーパス中に出現していないものが多いことが分かる．実際，コーパス 16,421 文中，括弧を含む文は 2,695 文しかなく，文数が十分ではない．

係り受け関係を表す規則

ラベル付け用の文法中に，係り受け関係 (連用修飾と連体修飾) を表す規則は 1,584 規則ある[6]．このうちA類，B類，C類の規則数はそれぞれ603規則，365規則，616規則であり，出現率は低い．詳細を調査するために，これらの規則を係り先の句 (動詞，形容詞，判定詞，名詞) ごとに分類し，それぞれの規則数を表6.6に示す．これより，動詞句に係る構造を表す規則の出現率に比べて形容詞や判定詞

[6]名詞終止文や副詞終止文など用言以外の語が文末に現れる文に関する規則は含まない．これらの規則については，次の項目で述べる．

表 6.5: 補足節を左辺に持つ規則の分布 (括弧を含む補足節)

助詞	A類	B類	C類	合計	出現率
*	15	8	25	48	31.25%
が	10	7	79	96	10.42%
を	12	10	74	96	12.50%
に	16	6	74	96	16.67%
で	16	19	61	96	16.67%
と	7	6	83	96	7.29%
の	1	1	94	96	1.04%
から	6	4	86	96	6.25%
へ	2	0	94	96	2.08%
より	3	4	89	96	3.13%
にて	0	0	96	96	0.00%
など	4	3	17	24	16.67%
まで	1	2	21	24	4.17%
合計	93	70	893	1,056	8.81%

表 6.6: 係り受け関係を表す規則の分布

	A類	B類	C類	合計	出現率
動詞	229	113	130	472	48.52%
形容詞	152	115	205	472	32.20%
判定詞	155	119	198	472	32.84%
名詞	168	18	83	168	39.88%
合計	603	365	616	1,584	38.07%

に係る構造を表す規則の出現率が低いことが分かる．これは，動詞に係る補足節はガ格，ヲ格など多様であるが，形容詞や判定詞がガ格以外の補足節を受ける例は非常に少ないことが原因である．

- 従ってＵＳＴＲは，議会の圧力に弱い
- 後には何も残らないのと同じだ

また，連体修飾を表す規則の出現率も，動詞に係る構造を表す規則の出現率に比べて低い．これは，連体修飾を表す規則の中に含まれる名詞の並列構造を表す規

表 6.7: 係り受け関係を表す規則の分布 (係り先の句が括弧を含む場合)

	A類	B類	C類	合計	出現率
動詞	27	68	53	148	18.24%
形容詞	8	46	94	148	5.41%
判定詞	7	40	101	148	4.73%
名詞	15	12	57	84	17.86%
合計	57	166	305	528	10.80%

則が二つの名詞をつなぐ語のパターンの数だけ必要となることが原因である．名詞をつなぐ役割を果たす語には，並立助詞，読点，接続詞，副詞がある．

- 東京と大阪 (並立助詞)
- 東京，大阪 (読点)
- 東京そして大阪 (接続詞)
- 東京さらに大阪 (副詞)

さらに,「東京，そして大阪」のように複数の語が組み合わさることもある．これらのパターンを網羅するためには，それぞれに対応した文法規則が必要となり，それらの規則を網羅するためには，それぞれの規則を使用する文を集めなければならない．

さらに詳細に調べるため，係り受け関係を表す規則を，係り先の句が括弧を含むか否かで2種類に分け，括弧を含む場合の規則のみを集計した．結果を表6.7に示す．これより，係り先の句が括弧を含む場合の規則の出現率が非常に低いことが分かる．この規則を使用する文には，以下のような文がある．

- 超高齢化社会の進展に伴い，今後は「ある程度黒字になる」と予想する事業者は４３％に上る．

括弧外の節が括弧内の末尾の語に係る例が少ないために，対応する規則の出現率が低くなっている．

表 6.8: 用言以外の語が文末に現れる文のための規則の分布

	A類	B類	C類	合計	出現率	文数
名詞終止句	97	60	83	240	40.42%	4,022
助詞終止句	50	58	136	244	20.49%	183
副詞終止句	39	44	141	224	17.41%	100
連用終止句	34	81	161	276	12.32%	112
合計	220	243	521	984	22.36%	—

用言以外が文末に現れる文に関する規則

一般に，日本語では文末に用言の終止形が現れる．ところが，名詞や副詞などの用言以外の語や，補足節，連用節が文末に現れることもある．

名詞終止句: 研修生は３５歳までの男女が対象．

副詞終止句: 二段階の引き上げは初めて．

助詞終止句: 箱詰めは七百円から．

連用終止句: 親友と気が合わないから．

これらの構造を扱うための規則は984規則あるが，そのコーパス中の分布を表6.8に示す．これより，助詞終止句，副詞終止句，連用終止句に関する規則の出現率が特に低いことが分かる．これは，文数からも明らかであるように，コーパス中の文数が十分ではないことが原因である．

特殊規則

本研究で作成したコーパスでは，以下の文に構文構造を付与するために，特殊な構造を認めている (付録B.3.6).

～から～まで: 六十歳から八十三歳まで の三百八十七人が回答した

～から～へ: 朝鮮高級学校から日本の高校へ の編入資格はない

～に～: ２週間に１度 は酒を飲んで大騒ぎ

～から～: サンクリストバルから十キロ の陸軍基地へのゲリラ攻撃

表 6.9: 特殊な構造を表す規則の分布

	A類	B類	C類	合計	出現率	文数
～から～まで	20	12	132	164	12.20%	28
～から～へ	10	9	145	164	6.10%	13
～に～	7	7	14	28	25.00%	13
～から～	17	14	21	52	32.69%	41
合計	54	42	312	408	13.24%	—

表 6.10: 読点を右辺に含む規則の分布

A類	B類	C類	合計	出現率	文数
243	130	1,184	1,557	15.61%	11,333

これらの構造を表す規則の分布を表6.9に示す．これより，特殊な構造を持つ文の数がまだ十分ではないことが分かる．

右辺に読点を含む規則

読点を右辺に含む規則は1,557規則ある．これらの規則の分布を表6.10に示す．これより，読点を右辺に含む規則は，あまりコーパス中に出現していないことが分かる．読点を含む文は11,333文あり，一見すると十分あるように思われるが，読点はあらゆる箇所に出現可能であるため，そのすべてを網羅するためにはさらに多くの文が必要である．

6.1.3　追加すべき文

以上より，コーパスに新たに文を追加する際には，以下の文を中心に追加すべきである[7]．

- 仮定形，命令形，意志形の用言を持つ文
- 「へ」,「の」,「にて」が末尾に現れる補足節を持つ文

[7]この考察は，文法の被覆率を向上させることを目的としている．このコーパスから統計情報を抽出するためには，無作為に文を選んで追加しなければならない．

- 名詞句以外の句を含む補足節を持つ文(「～するかどうかが」など)
- 形容詞，判定詞をガ格以外の補足節が修飾する文
- 名詞の並列構造を持つ文
- 括弧を含む文
- 用言以外が文末に現れる文
- 特殊構造を持つ文
- 読点を含む文

6.2　ラベル付け作業者間の一致に関する考察

第4.1節で述べたように，コーパスは複数の作業者によって人手で作成されるため，作業者間の構文構造の不一致が生じる可能性がある．本研究では，コーパス作成支援ツールを使用し，方針に従って人手で作成した文法による解析結果(統語圧縮共有森)をもとに作成しているため(第5.1.1節)，文法による制約に反する構文構造を誤って付与する可能性は排除できる．しかし，文法による制約を満たしていても，作業者によって異なる構文構造を付与してしまう可能性は残っており，これを防ぐためにはラベル付け方針を細かく設定しなければならない．一方，方針を細かく設定したとしても，その方針が作業者の直感に反して複雑であると，作業者の混乱を招き，ラベル付けの不一致を引き起こす可能性がある．そこで，本節では，実際に作業者間のラベル付けの不一致を調査し，今回のラベル付け方針の問題点を考察する．この結果は，今後，コーパス作成方針を再検討する際の参考になる．

6.2.1　二人の作業者間のラベル付け一致度

作業者間のラベル付けがどの程度一致するかを調査するために，2人の作業者に，同じ文(495文)に対するラベル付けを依頼した．一致度の評価にはevalb[8]を

[8]http://nlp.cs.nyu.edu/evalb/

表 6.11: ラベル付け可能か否かの判定の一致度

		作業者 A		
		可	不可	合計
作業者 B	可	383	36	419
	不可	26	50	76
合計		409	86	495

利用し[9]，F 値 (bracketing recall と bracketing precision から計算) と完全一致率 (complete match) で評価する．

まず，ラベル付け可能か否か (方針により対象外となるか否か) の判定の一致度 (文数) を表 6.11 に示す．これより，87.47%にあたる 433 文について，両者の判定が一致していることが分かる．次に，両者がラベル付けした 383 文について，evalb により F 値と完全一致率を求めたところ，それぞれ 0.98 と 75.72%であった．

6.2.2　矛盾した構造の分析

まず，一方の作業者がラベル付けし，もう一方の作業者がラベル付けしなかった 62 文について分析する．主な例を示す．

(1) 荷台が前後に 5 メートル，左右に 1 メートル広がる

(2) 大蔵，自治両省と税調が選んだ

(3) 平和を願い，大切に生きること，厳しさをバネにして生きる姿勢

(4) 上の子の病気で予定通り行動できない

(5) 六人が死亡，二十五人が負傷した

(6) このような言葉が使われるのは，どうも野球選手だけのような気がする

(1) は，「前後に 5 メートル」と「左右に 1 メートル」が並列関係にあるが，今回の方針では，4 つの連用修飾句が別個に動詞「広がる」を修飾する構造としている．しかし，一方の作業者は，「前後に 5 メートル」と「左右に 1 メートル」それ

[9]デフォルトでは，ラベルの長さが最長 30 バイトとなっている．本研究で作成したコーパスでは，30 バイトを超える長さのラベルも存在するため，最大値を 100 バイトに変更して利用した．

ぞれをまとめようとしたものの文法の制約によりラベル付けできず，ラベル付け対象外と判断した．(2) は,「大蔵」と「自治」が並列関係にあり,「大蔵，自治」に対して「両省」が結合すると考えることが自然であるが，今回の方針では，複合名詞の内部に名詞句を持つ構造を認めず,「大蔵」と「自治両省」が並列関係にあると判断し，連体修飾関係と同じ構造を付与することとしている．ところが，この場合も，一方の作業者は,「大蔵」と「自治」を先にまとめられないとして，ラベル付け対象外と判断した．このように，意味的に考えて自然な構造と異なる場合，その自然な構造のイメージにとらわれてしまうことでラベル付けできないと判断してしまうことがある．

(3) は連体修飾句が連続する例であり，今回の方針では右下がりの構造に制限される．ところが，連続する連体修飾句の数が多くなったり，連体修飾句の中に用言が含まれたりすると，右下がりの構造に制限される範囲が分からなくなり，結果として，ラベル付けできないと判断してしまう可能性がある．

(4) は,「予定通り」が一般名詞であるために連用修飾句になれず，構文構造を付けることはできない．ところが，一方の作業者はそれに気付かず,「予定通り行動」で一つの複合名詞になる構造を付与している．今回使用しているコーパス作成支援ツールは，構文解析結果の集合を入力とし，一つの正しい構文構造に絞り込めるまで正しい構文構造が満たすべき条件を作業者が指定していくものである．このように，構文解析結果の集合をスタート地点とすることで，文法の制約に反する構造を誤って付けることを防いでいるが，そのために，作業者は，解析結果の集合の中から一つに絞り込むことができると，ラベル付け対象外となる文であるにも関わらず，細かい構造の確認をせずそのまま付けてしまう可能性がある．

(5) は，サ変名詞「死亡」が単独で動詞になる例である．サ変名詞は直後に「する」が結合することで動詞となるが，今回の方針では,「する」が省略されていても，サ変名詞は単独で動詞になれるとしている．ところが，一方の作業者は「する」が省略されているためにラベル付けできないと判断した．これは，以下の例文のように，助詞や用言が省略されている場合はラベル付け対象外とする方針を拡大解釈してしまったことが原因である (括弧で囲まれた部分が省略されている語句)．

- ところがこの丸刈り (は)，日本が韓国を併合した直後の一九一二年に定められた旧監獄令から生まれたものだ
- 縁故米などの形でヤミ取引される可能性も (ある)

このように，方針の適用範囲を誤解してしまう可能性がある．

(6)では，一方の作業者は,「このような言葉が使われるのは」の係り先は「野球選手だけのような」であり,「どうも」の係り先は「気がする」であると考え，係り受け関係が交差するとしてラベル付け対象外と判断している．今回の方針では，連用修飾句の係り先の決定は意味を考えた上で作業者が判断することになっているが，その判断が一致しない可能性がある．

次に，どちらの作業者も構文構造を付与したが，その構造が一致しなかったものについて分析する．これは全部で93文あった．以下にその例を挙げる．

(1) 私は，ツシマヤマネコもトキのたどった道を強制されていると思えてならない

(2) また篤実なキリスト者としても知られ，在日韓国・朝鮮人問題や天皇制をめぐる市民運動でも活躍してきた

(3) 今年六月に再開された包括協議の協力分野を除く個別分野の中で，両国が合意に達したのは初めて

(4) お金とはお国柄が出るものだなあ

(5) 六・四八倍から四・八一倍に縮小する

(6) 赤字予算を組むのは天安門事件以降，五年連続

(7) ベンツェン財務長官以下四閣僚が勢ぞろいして記者会見した

(8) 申し込み多数の場合は抽選

(9) 今後都市部周辺の黒人居住区の開票が進めば，さらにＡＮＣ支持票が増える見込み

(1), (2)では，連用修飾句「私は」と「また」の係り先が一致していない．先に述べたように，連用修飾句の係り先は作業者が判断するため，その判断が食い違う可能性が高い．実際，構文構造が一致しなかった93文のうち，連用修飾句の係り先が異なるものは81文あった．

(3)は,「今年六月に再開された包括協議の協力分野を除く個別分野の中」が名詞句となるが,「包括協議の」の係り先が「協力分野」であるか,「(個別分野の)中」になるかで異なっている．先に述べたように，連体修飾句が連続する場合は右下が

りの構造に制限されるが，途中に用言が入る場合には構造が複雑になるため，構造を間違えやすくなる．

(4) は，形式名詞に助動詞「だ」が結合するパターンの例である．このパターンは2種類に分類できる．

- 豆腐は大豆から作られたものだ
- 昔はよく遊んだものだ

前者は,「大豆から作られたもの」が名詞句になるのに対し，後者は「ものだ」が助動詞的な役割を果たして「遊んだものだ」が先にまとめられる．これは，作業者が意味を考慮して判断することになっているが，その判断が異なる場合がある．

(5) は,「六・四八倍から」と「四・八一倍に」が「縮小する」を修飾する構造が正しいが，一方の作業者は「～から～」というパターンを扱うための特殊ルール (付録B.3.6) を誤って適用している．

(6) は,「天安門事件以降」が連用修飾句となるが，一方の作業者は，誤って「五年連続」と並列関係となる構造 (すなわち，連体修飾関係) を付けている．

(7) は,「ベンツェン財務長官以下四閣僚」が一般名詞であるか数量詞であるかの違いである．一般に，数字と単位 (後置助数詞) が結合すると数量詞となるが,「一単語」のように一般名詞が結合する場合でも数量詞的な意味になることがある．そのために，数字と一般名詞が結合する場合には，それが一般名詞になる規則と数量詞になる規則の二つが用意されているため，どちらを選ぶかで構造が一致しない場合がある．さらに，この例では，複合名詞が一般名詞であるか数量詞であるかで非終端記号名が異なるだけで，それ以外の構造はすべて同じであるため，作業者が誤りに気付かない可能性もある．

(8) は，サ変名詞が文末に出現する例である．サ変名詞が文末に出現する場合，その直後に「する」が省略されていると考えられる場合は動詞,「だ」や「です」が省略されていると考えられる場合は名詞終止文 (付録B.3.5) としている．この例は「する」ではなく「だ」が省略されていると考える方が自然であるが，一方の作業者は「する」が省略されていると判断したために，異なる構造が付けられた．

(9) では,「今後」が連用修飾句となるが，一方の作業者は「今後都市部周辺」で一つの複合名詞としていた．これは，構造を細かく確認しなかったことによる単純な誤りであると考えられる．

6.2.3　分析のまとめ

今回の方針では，連用修飾句の係り先は意味を考慮して決定することとなっているため，作業者間で不一致が生じる可能性が高い．中には，複数ある係り先の候補のうち，どれを修飾する構造であっても良いと考えられるものも少なくない．これらについては，具体例を挙げながら一つ一つ方針を明確にしていく必要がある．

連体修飾句の係り先は右下がりの構造になるように制限されているが，途中に用言が入ったり，連続する連体修飾句の数が多くなったりすると，本来の意味的に正しい構造のイメージにとらわれ，誤った構造をつけやすくなる傾向がある．今回の方針では，周辺に存在する連用修飾句の範囲を変える場合と変えない場合の2通りに分類して，連体修飾句の係り先に関する制約を決定している．しかし，作業者間の一致度を向上させるためには，別の分類により制約を決定する必要があるかもしれない．

作業者間の一致度を向上させるための手段として，ラベル付け方針の再検討のほかに，コーパス作成支援ツール側の改善も考えられる．例えば，文が長ければ長いほど，構造が矛盾する可能性が高くなる．これは，現在使用しているツールでは，構文構造全体を一画面に表示することができないことが原因であると考えることもできる．また，複合名詞内の構造や，連体修飾関係を表す構造は右下がりに制限しているが，その構造が作業者の直感に合わないために，混乱を招く可能性がある．意味に関係なく制限している構造をツール上で表示する際には，そのまま右下がりの構造で表示するのではなく，深さ1のフラットな構造で表示するなど，表示方法を工夫することで構造の矛盾を防ぐことができる可能性がある．さらに，ある構造を決定するために与えた条件が，作業者の想定外の構造まで決定してしまうことがある．その構造が誤っていても作業者が気付かない場合，その誤った構造がそのまま付けられてしまう可能性もある．直前に与えた条件によって確定した部分が分かれば，誤った構造を付与する可能性を減少させられると考えている．

過去にラベル付けされたデータを参照することも，構文構造の一貫性を保つ手段の一つである．そのためには，類似文の検索や一貫性をチェックするためのシステムなどが必要となる．これらのシステムを開発し，コーパス作成支援ツールと統合することで，作業者間のラベル付けの一致度を向上させることができると考えている．

第7章　結論

7.1　本研究のまとめ

大規模な構文構造付きコーパスは，自然言語処理技術にとって重要な知識資源のひとつである．実際，Penn Treebank コーパスなどの構文構造付きコーパスの作成により，様々な研究成果が生まれている．ところが，日本語では Penn Treebank コーパスのような構文構造付きコーパスが存在せず，これまで同様の研究を日本語で行うことが困難であった．そこで，本研究では，日本語の構文構造付きコーパスを作成した．その際，コーパスから抽出した文法による構文解析結果の曖昧性が非常に大きいという問題点に着目し，曖昧性を極力抑えることを目的として付与すべき構文構造の検討を行った．

これまでにも，コーパスに対して様々な情報を追加，変更した上で文法を再抽出し，解析精度がどの程度向上するかを調査した研究はいくつか報告されている [26, 47]．ところが，いずれの研究も，抽出した文法が出力する構文解析結果の曖昧性について明確に検討していない．曖昧性の増大は，解析速度の低下，使用メモリ量の増大を招くだけでなく，解析精度の低下の原因にもなる．解析途中で枝刈りを行うことで，解析速度の低下や使用メモリ量の増大を防ぐことは可能であるが，枝刈りによって，正しい解析結果が途中で排除される可能性もある．特に，解析の初期の段階では，それまでの結果に対する生成確率に大きな差が生じない上に，ごく一部の構造の生成確率に過ぎないために大小が逆転する可能性も高く，正しい解析結果が排除される可能性が高くなると考えている．曖昧性を増大させる要因を分析し，その分析結果をもとにコーパス変更 (作成) 方針を決定し，それに従ってコーパスを変更 (作成) する必要がある．白井ら [48] は，構文解析結果の曖昧性を考慮しながら構文構造の変更を行っているが，その変更は，機械的に行える部分が中心であり，まだ十分であるとは言えない．時間と労力を要するが，曖昧性を十分に抑えるためには，人手による変更が必要である．

本研究では，まず，構文構造付きコーパスから抽出した文脈自由文法 (tree-bank

grammar) 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示し，それを解決するためのコーパス変更方針を提案した．具体的には，曖昧性を増大させる要因として，以下の4点を指摘した．

(1) 作業者の誤り

(2) 構文構造の不一致

(3) 構文解析に必要な情報の不足

(4) 意味的曖昧性

このうち，要因3と要因4について，以下のようにコーパス変更方針を決定した．

(1) 用言の活用形に関する情報を非終端記号に追加

(2) 複合名詞内の構造を，意味や語構成に関係なく同一の構造で表現

(3) 連体修飾関係の構造を，連用修飾句の範囲を変えない場合に限り，意味に関係なく同一の構造で表現

(4) 並列関係であるか否かの区別をしない

この方針の有用性を確認するため，EDR コーパスと RWC コーパスで評価実験を行った．まず，EDR コーパス 8,911 文に対して「変更前」と「変更後」の構造を付与した2種類のコーパスを用意し，曖昧性と解析精度を調べたところ，構文解析結果の数は 10^{12} オーダから 10^{5} オーダまで減少し，PGLR モデルによる生成確率上位 100 位までの文正解率は 8%以上向上した．また，RWC コーパス 16,421 文に対して「変更後」の構造を付与したコーパスから抽出した文法で同様の実験を行ったところ，同様の結果が得られた．これより，本研究で提案したコーパス作成 (変更) 方針は，他のコーパスに対しても適用可能であることを確認した．

さらに，EDR 変更後コーパスから抽出した文法について，EDR 変更前コーパスの構造を正解とした場合の PGLR モデルによる生成確率1位の解析結果の係り受け精度を調べたところ，意味的な情報を使用していないにも関わらず，89.61%の精度が得られた．一方，RWC 変更後コーパスから抽出した文法について，京大コーパスの構造を正解として同様の評価を行ったところ，82.88%の精度が得られた．今後，本格的に意味解析を行うことで，解析精度をさらに向上させることができると考えている．

最後に，コーパスから抽出した文法やコーパス変更(作成)方針について考察した．本研究では，コーパスを作成するために，大雑把な文法を人手で作成したが，その文法とコーパスから抽出した文法を比較することで，コーパスから抽出できなかった文法規則の特徴を調査した．その結果，以下の文法規則が十分抽出できていないことが分かった．

- 「の」,「へ」,「にて」が末尾に出現する補足節に関する規則
- 名詞句以外の句を含む補足節に関する規則
- 係り受け関係を表す規則
- 括弧を含む句に関する規則
- 用言以外の語が文末に出現する文のための規則
- 読点を含む構造に関する規則
- 命令形，意志形，仮定形の用言に関する規則

今後，コーパスに新たな文を追加する際には，それまでの文から抽出できなかった文法規則を含む構造を持つ文を中心に追加する必要がある．

さらに，コーパス変更(作成)方針について考察するために，二人の作業者が付与した構文構造の一致度を調査した．その結果，連用修飾句の係り先の決定が一致しない可能性が高いことが分かった．その他，連体修飾句が多数連続する名詞句の構造，特に，その中に用言が含まれる場合なども矛盾を生じやすいことが判明した．これらの考察結果は，今後，コーパス作成方針の再検討を行う際に有用となる．

本研究で作成したコーパスは，文法抽出を目的とし，抽出した文法による構文解析結果の曖昧性を抑えるために，複合名詞内の構造の曖昧性，連体修飾関係の曖昧性などを単一の構造で表現している．文法抽出以外の目的でコーパスを利用する場合，これらの曖昧性を単一の構造で表現することは適切ではない．しかし，本研究で作成したコーパスについて，意味に関係なく単一の構造で表現している部分だけを対象に構文構造を付与し直せば，意味的曖昧性を区別した構文構造を持つコーパスを作成できる．この方法は，意味的曖昧性を区別した構文構造付きコーパスを一から作成するよりも少ない労力で作成可能である．先に意味的曖昧性を区別したコーパスを作成してから，特定の意味的曖昧性を単一の構造に置換

することで文法抽出のためのコーパスを作成することも可能であるが，曖昧性の少ない作成方針に基づいて，先に文法抽出のためのコーパスを作成し，その後，意味的曖昧性を区別した構文構造を付与し直す方が，作成に要する労力が少ないと考えている．

7.2　今後の課題

本研究では，構文解析後に意味解析を行うことを前提とし，意味的曖昧性を極力抑えた構文構造をコーパスに付与している．今後，後処理として想定している意味解析の手法を検討しなければならない．第4.3節で述べたように，本研究で提案したコーパス作成方針では，曖昧性を連用修飾関係と連体修飾関係に大きく分けている．これにより，意味解析を連用修飾関係と連体修飾関係に分けて検討することが可能となる．現在の方針では，連用修飾関係の曖昧性は異なる構造で表現され，連体修飾関係の曖昧性はコーパスに付与する構文構造では区別していない．このことから，以下の手順で意味解析を行うことを想定している．

(1) 連用修飾関係の曖昧性を解消 (複数の構文解析結果の中から尤もらしい構造を選択)

(2) 連体修飾関係の曖昧性を解消 (選択した構造が包含するすべての連体修飾関係から尤もらしいものを選択)

そのため，語の共起関係や意味属性など様々な情報を大量に収集する必要があり，効率的な収集方法の検討も今後の課題となる．

現在，構文構造を付与した文は16,421文であるが，これはPenn Treebankコーパス (約5万文)，Negraコーパス (約2万文)，京大コーパス (約4万文) など他のコーパスに比べると少ない．今後，コーパスの規模を大きくする必要があるが，そのために，以下の二つの側面から，最終的にどの程度の規模まで大きくすれば充分であるかの検討も必要である．

(1) 多様な言語現象を網羅するために必要な規模 (抽出した文法の被覆率)

(2) 確率モデルの学習に必要な規模 (構文解析精度)

(1) は，第6.1節の考察結果を参考にすることができる．

第6.2節では，作業者間の構文構造の矛盾が起きやすい構造について考察した．今後，矛盾を極力防ぐために，コーパス作成方針の再検討が必要である．具体的には，曖昧性を極力抑えながらも，作業者に違和感を与えない，判断に迷わない構造を検討することで，矛盾を防ぐことができると考えている．本研究で提案したコーパス作成方針では，意味的曖昧性を複合名詞内の構造の曖昧性，連体修飾関係の曖昧性，連用修飾関係の曖昧性，並列構造の曖昧性の4種類に分類しているが，細分類や別の観点からの分類を検討することも有用である．コーパス作成方針の再検討のほかに，コーパス作成支援ツールを改善することによって作業者間の構文構造の不一致を防ぐことも可能である．現段階で想定している具体的な改善点を以下に示す．

(1) 複合名詞内の構造，連体修飾関係など意味に関係なく構造を制限している箇所について，その構造をそのまま表示するのではなく，作業者に違和感を与えないような表示方法を考える必要がある

(2) 作業中，付与すべき構文構造に迷いが生じた際に，過去にラベル付けされたデータを参照することで，作業効率が向上するだけでなく，構造の不一致を未然に防げる．

謝辞

本研究を進めるにあたり，多くの方々から御指導，御支援を頂きました．ここに心からの感謝を捧げます．

まず，研究テーマの設定から学生生活全般に至るまで終始様々な御助言を賜わりました指導教官の田中穂積教授に心より感謝いたします．また，日々の議論を通じて数多くの貴重なご意見を頂きました徳永健伸助教授，橋本泰一助手に深く感謝いたします．

本研究は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」で行ってきました．古井貞熙教授をはじめとする本プログラムの推進担当者の皆様，博士フォーラムの皆様には様々な意見を頂きました．心より感謝いたします．

小林正博氏，大久保佳子氏をはじめとする日本システムアプリケーションの皆様には，コーパス作成および修正作業において多大な御協力を頂きました．奈良先端科学技術大学院大学の松本裕治教授には，RWCテキストコーパスの修正版を提供して頂きました．ワーズビークルの渕武志氏には，京大コーパス，RWCテキストコーパス間の品詞体系の変換のためのプログラムを提供して頂きました．富士通研究所の松井くにお氏，斉藤孝広氏，麻岡正洋氏には，本研究で作成したコーパスと文法を利用していただき，様々な意見を頂きました．厚く御礼を申し上げます．

田中・徳永研究室の皆様には，ミーティングなどを通して，折に振れ貴重な御意見を頂きました．また，コーパス作成，修正作業を手伝って頂きました．Slaven Bilac氏には英語論文の校正で大変お世話になりました．秘書の方々にも，煩雑な事務手続きを引き受けて頂きました．ここに深く感謝いたします．

CNRS-LIMSIのMichael Zock氏には，来日中，英語論文の校正で大変お世話になりました．改めて御礼を申し上げます．

その他，研究を支えてくださった皆様に改めて感謝いたします．

参考文献

[1] Susanna Alfonso, Eckhard Bick, Renato Haber, and Diana Santos. "Floresta Sintá(c)tica": A treebank for Portuguese. In *the 3rd International Conference on Language Resources and Evaluation*, pp. 1698–1703, 2002.

[2] Leonoor van der Beek, Gosse Bouma, Jan Daciuk, Tanja Gaustad, Robert Malouf, Gertjan van Noord, Robbert Prins, and Bego na Villada. The Alpino Dependency Treebank. In *Algorithms for Linguistic Processing NWO PIONIER Progress Report*, chapter 5. Groningen, 2002.

[3] Daniel M. Bikel and David Chiang. Two statistical parsing models applied to the Chinese treebank. In *the 2nd ACL Workshop on Chinese Language Processing*, 2000.

[4] Alena Böhmová, Jan Hajič, Eva Hajičová, and Barbora Hladká. The Prague Dependency Treebank: Three-level annotation scenario. In Anne Abeillé, editor, *Treebanks: Building and Using Syntactically Annotated Corpora.* Kluwer Academic Publishers, 2001.

[5] Sabine Brants, Stefanie Dipper, Silvia Hansen, Wolfgang Lezius, and George Smith. The TIGER Treebank. In *the 1st Workshop on Treebanks and Linguistic Theories*, pp. 24–41, 2002.

[6] Glenn Carroll and Mats Rooth. Valence induction with a head-lexicalized PCFG. In *the Conference on Empirical Methods in Natural Language Processing*, 1998.

[7] Eugene Charniak. Tree-bank grammars. In *the 13th National Conference on Artificial Intelligence*, pp. 1031–1036, 1996.

[8] Eugene Charniak. A maximum-entropy-inspired parser. In *the 1st Conference of the North American Chapter of the ACL*, 2000.

[9] Noam Chomsky. *Lectures on Government and Binding.* Foris Publications, 1981.

[10] Kenneth Church and Ramesh Patil. Coping with syntactic ambiguity or how to put the block in the box on the table. *American Journal of Computational Linguistics*, Vol. 8, No. 3–4, pp. 139–149, 1982.

[11] Martin Čmejrek, Jan Cuřín, and Jiří Havelka. Prague Czech-English Dependency Treebank: Any hopes for a common annotation scheme? In *the HLT/NAACL 2004 Workshop: Frontiers in Corpus Annotation*, pp. 47–54, 2004.

[12] Michael Collins. Three generative, lexicalized models for statistical parsing. In *the 35th Annual Meeting of the ACL and the 8th Conference of the European Chapter of the ACL*, 1997.

[13] Michael Collins, Jan Hajič, Lance Ramshaw, and Christoph Tillmann. A statistical parser for Czech. In *the 37th Annual Meeting of the ACL*, 1999.

[14] Markus Dickinson and W. Detmar Meurers. Detecting errors in part-of-speech annotation. In *the 11th Conference of the European Chapter of the ACL*, 2003.

[15] Markus Dickinson and W. Detmar Meurers. Detecting inconsistencies in treebanks. In *the 2nd Workshop on Treebank and Linguistic Theories*, 2003.

[16] Amit Dubey and Frank Keller. Probabilistic parsing for German using sisterhead dependencies. In *the 41st Annual Meeting of the ACL*, pp. 96–103, 2003.

[17] Jason Eisner. Efficient normal-form parsing for combinatory categorial grammar. In *the 34th Annual Meeting of the ACL*, pp. 79–86, 1996.

[18] Liliane Haegeman. *Introduction to Government and Binding Theory.* Blackwell Publishers, 1994.

[19] Koichi Hashida, Hitoshi Isahara, Takenobu Tokunaga, Minako Hashimoto, Shiho Ogino, and Wakako Kashino. The RWC text database. In *the 1st International Conference on Language Resource and Evaluation*, pp. 457–461, 1998.

[20] Chung hye Han, Na-Rae Han, Eon-Suk Ko, Heejong Yi, and Martha Palmer. Penn Korean Treebank: Development and evaluation. In *the 16th Pacific Asia Conference on Language, Information and Computation*, 2002.

[21] Kentaro Inui, Virach Sornlertlamvanich, Hozumi Tanaka, and Takenobu Tokunaga. Probabilistic GLR parsing: A new formalization and its impact on parsing performance. 自然言語処理, Vol. 5, No. 3, pp. 33–52, 1998.

[22] Mark Johnson. PCFG models of linguistic tree representations. *Computational Linguistics*, Vol. 24, No. 4, pp. 613–632, 1998.

[23] Aravind K. Joshi and Yves Schabes. Tree-adjoining grammars. In Grzegorz Rozenberg and Arto Salomaa, editors, *Handbook of Formal Languages*, Vol. 3, chapter 2. Springer-Verlag, 1997.

[24] Daniel Jurafsky and James H. Martin. *Speech and Language Processing.* Prentice-Hall, 2000.

[25] Ronald M. Kaplan and Joan Bresnan. Lexical-functional grammar: A formal system for grammatical representation. In Joan Bresnan, editor, *The Mental Representation of Grammatical Relation*, chapter 4, pp. 173–281. MIT Press, 1982.

[26] Dan Klein and Christopher D. Manning. Accurate unlexicalized parsing. In *the 41st Annual Meeting of the ACL*, pp. 423–430, 2003.

[27] Nobo Komagata. Efficient parsing for CCGs with generalized type-raised categories. In *IWPT 97*, pp. 135–146, 1997.

[28] Alexander Krotov, Mark Hepple, Robert Gaizauskas, and Yorick Wilks. Evaluating two methods for treebank grammar compaction. *Natural Language Engineering*, Vol. 5, No. 4, pp. 377–394, 1999.

[29] 工藤拓, 松本裕治. 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, Vol. 43, No. 6, pp. 1834–1842, 2002.

[30] 黒橋禎夫, 長尾眞. 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, Vol. 33, No. 8, pp. 1022–1031, 1992.

[31] 黒橋禎夫, 長尾眞. 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, 1997.

[32] 黒橋禎夫. 日本語構文解析システム KNP version 2.0 b6, 1998.

[33] 黒橋禎夫, 長尾真. 日本語形態素解析システム JUMAN version 3.61, 1999.

[34] K. Lari and S. J. Young. The estimation of stochastic context-free grammars using the Inside-Outside algorithm. *Computer Speech and Language*, Vol. 4, pp. 35–56, 1990.

[35] Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, Vol. 19, No. 2, pp. 313–330, 1993.

[36] W. A. Martin, K. W. Church, and R. S. Patil. Preliminary analysis of a breadth-first parsing algorithm: Theoretical and experimental results. In Leonard Bolc, editor, *Natural Language Parsing Systems*, pp. 267–328. Springer-Verlag, 1987.

[37] 益岡隆志, 田窪行則. 基礎日本語文法 –改訂版–. くろしお出版, 1992.

[38] 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. 形態素解析システム『茶筌』 version 2.3.0, 2003.

[39] 美野秀弥, 橋本泰一, 徳永健伸, 田中穂積. 日本語の連体修飾関係に関する研究. 言語処理学会第10回年次大会, pp. 600–603, 2004.

[40] 日本電子化辞書研究所. EDR電子化辞書2.0版仕様説明書, 2001.

[41] 西尾悠. PCFGモデルとPGLRモデルの比較に関する一考察. Master's thesis, 東京工業大学大学院情報理工学研究科計算工学専攻, 2004.

[42] 野村雅昭, 石井正彦. 複合動詞資料集. 国立国語研究所, 1987.

[43] 野呂智哉, 橋本泰一, 徳永健伸, 田中穂積. 大規模日本語文法の開発. Technical Report 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, 2003.

[44] 岡崎篤, 白井清昭, 徳永健伸, 田中穂積. 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, 2001.

[45] Carl Pollard and Ivan A. Sag. *Head-Driven Phrase Structure Grammar.* University of Chicago Press, 1994.

[46] Michael Schiehlen. Combining deep and shallow approaches in parsing German. In *the 41st Annual Meeting of the ACL*, pp. 112–119, 2003.

[47] Michael Schiehlen. Annotation strategies for probabilistic parsing in German. In *the 20th International Conference on Computational Linguistics*, pp. 390–396, 2004.

[48] 白井清昭, 徳永健伸, 田中穂積. 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, Vol. 4, No. 1, pp. 125–146, 1997.

[49] 白井清昭, 植木正裕, 橋本泰一, 徳永健伸, 田中穂積. 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, Vol. 7, No. 5, pp. 93–112, 2000.

[50] Wojciech Skut, Brigitte Krenn, Thorsten Brants, and Hans Uszkoreit. An annotation scheme for free word order languages. In *the 5th Conference on Applied Natural Language Processing*, 1997.

[51] Masaru Tomita. *Efficient Parsing for Natural Language.* Kluwer Academic Publishers, 1986.

[52] 内元清貴, 関根聡, 井佐原均. 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, Vol. 40, No. 9, pp. 3397–3407, 1999.

[53] 内元清貴, 村田真樹, 関根聡, 井佐原均. 後方文脈を考慮した係り受けモデル. 自然言語処理, Vol. 7, No. 5, pp. 3–18, 2000.

[54] Nianwen Xue, Fu-Dong Chiou, and Martha Palmer. Building a large-scale annotated Chinese corpus. In *the 19th International Conference on Computational Linguistics*, 2002.

[55] 八木豊, 野呂智哉, 橋本泰一, 徳永健伸, 田中穂積. 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, 2003.

付 録A　変更方針の検討に使用するコーパス

本章では，コーパスに付与されている構文構造の変更方針を検討するための出発点として使用している構文構造付きコーパスについて述べる．

本研究で使用しているコーパスは，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コーパス中の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．本章では，その相違点を中心に述べる．

A.1　基本構造

本研究で使用しているコーパスに付与されている構文構造は，以下の3層に分かれている(図A.1)．

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

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，非常に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[48]，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A.1)．ただし，

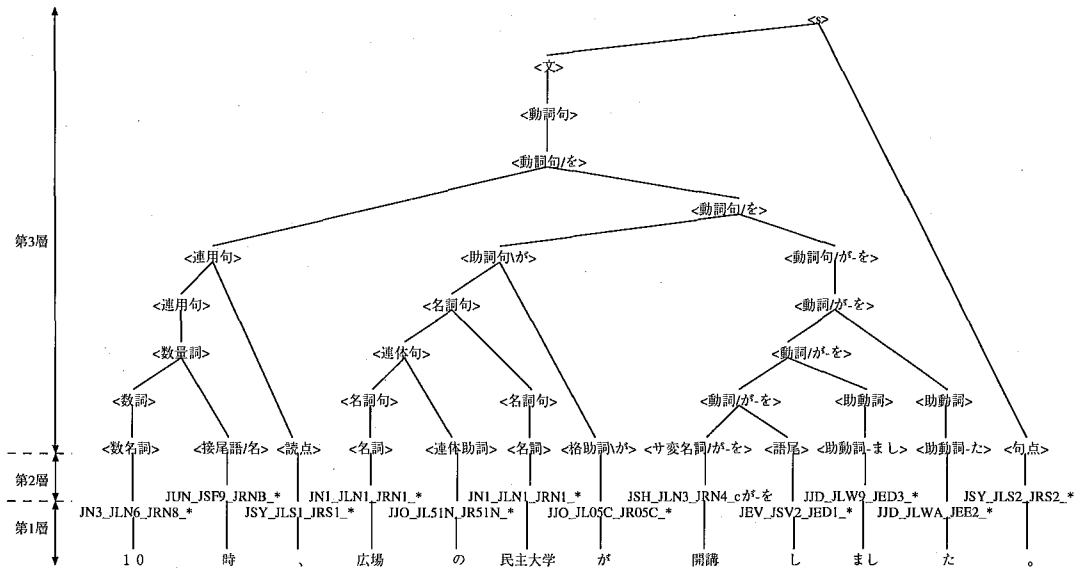


図 A.1: 構文構造を構成する三つの層

表 A.1: 品詞の細分化の例

単語	EDR コーパス	使用しているコーパス
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_*
(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

「開講」など「する」を伴なって動詞を形成するもの(サ変名詞)は，EDR日本語単語辞書では“JN1;JVE”という二つの品詞が割り当てられているが，本研究で使用しているコーパスでは“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．また，EDRコーパスでは,「不安感を強めている」の「てい(る)」は助詞「て」，動詞語幹「い」の2単語に分かれているが，本研究で使用しているコーパスでは，1語の助動詞相当句としている.「によって」などの助詞相当句も同様である[1].

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種類に達する(存在し得る品詞を含めると優に1,000種類を超える)．この品詞の上に直接構文構造を付与すると，そのコーパスから抽出した文法規則が必要以上に細かくなる．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層と第2層の間の品詞を省略し，第2層と第3層の間の品詞を終端記号とする．

A.3　構文構造

先に述べたように，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，本研究で使用したコーパスでは一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」という二つの後置詞句(助詞句)に対して，白井ら[48]の場合は図A.2(a)のような構造になり，本研究で使用しているコーパスでは図A.2(b)のような構造になる[2]．(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二つになるが，(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．その代わり，名詞句を構成するまでの部分木が深くなるが，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

[1]助詞相当句の左右連接属性は，先頭の語の左連接属性と末尾の語の右連接属性で決まるが，この決定方法では，格助詞「に」で始まり接続助詞「て」で終わる助詞相当句(「によって」,「にとって」など)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，さらに形態素ごとに分類している．

[2]白井らが「後置詞句」と呼んでいる句は，本研究で使用しているコーパスでは「助詞句」と呼んでいる．

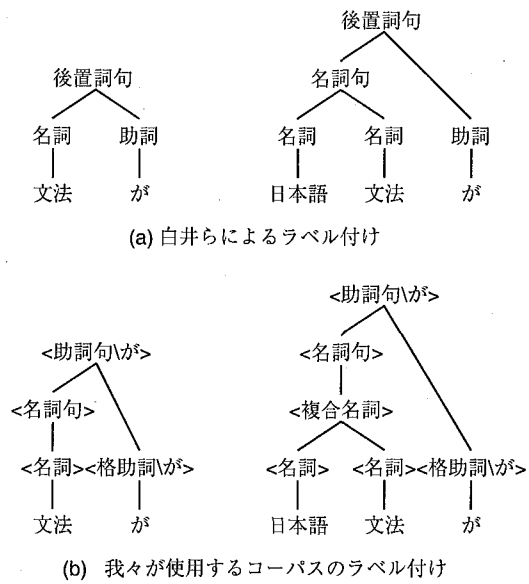


図 A.2: 白井ら [48] のラベル付けとの違い

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る．

構造は基本的にEDRコーパスの括弧付けに従うが，一部，EDRコーパスの括弧付けとは異なる構造を付与する場合がある．以下に，そのラベル付け方法を述べる．

A.3.1　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 [48]．本研究で使用しているコーパスも，同様の構造となっている．

A.3.2　フラットな構造

EDRコーパスの括弧付けの中には，細かい括弧付けがなく，多くの要素を一つの括弧でまとめてしまっているものがある [48]．その場合には，さらに細かい構造を付与する．

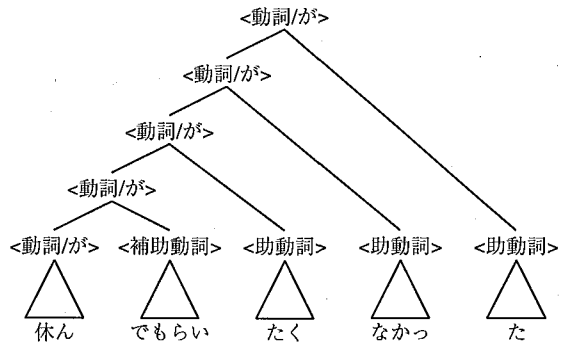


図 A.3: 動詞に複数の助動詞が結合する場合の構造

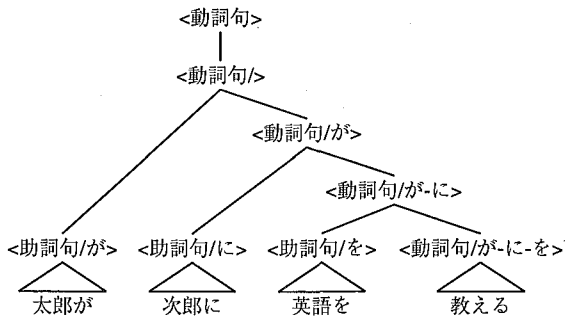


図 A.4: 用言のとる表層格を考慮した構造

A.3.3　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 コーパスでは一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 A.3)．この部分を EDR コーパスに従ってフラットな構造のままにすると，結合する助動詞や語尾のパターンだけ文法規則が必要となるが，左下がりの構造にすることで，少数の規則でより多くのパターンをカバーできるようになる．

A.3.4　用言がとる表層格情報の扱い

本研究で使用している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 A.4)．これにより，二重ヲ格などの制約を取り入れることが可能になる．

付 録B　RWCコーパスに対する構文構造の付与

本章では，RWCコーパスに対する構文構造の付与について述べる．

RWCコーパスは品詞タグ付きコーパスであり，品詞体系は形態素解析器の茶筌[38]が採用している体系と同じである．このコーパス中の文に対し，本研究で提案している方針に従って，コーパス作成支援ツール[44]を用いて構文構造を付与することになるが，それ以前に，一部の品詞，形態素区切りを自動的に変換している．茶筌の品詞体系，形態素区切りは，形態素解析には適しているが，構文解析には適さない部分があるからである．

RWCコーパスに対して構文構造を付与する手順を以下に示す．

(1) コーパスの一部を編集(除去)する

(2) 一部の品詞，形態素区切りを自動的に変換する

(3) 変換後の品詞体系に準拠した文法(コーパス作成用文法)を人手で作成する

(4) MSLRパーザで構文解析し，結果を統語圧縮共有森の形式で獲得する

(5) コーパス作成支援ツールで正しい構文構造を選択する

上記の手順を見れば分かるように，まえもって文法(コーパス作成用文法)を人手で作成しているが，この文法はコーパス作成のためだけに使用するものである．最終的に使用する文法は作成後のコーパスから抽出するため，不要な文法規則がコーパス作成用文法に含まれていても問題はない．逆に，必要な文法規則が含まれていないと，統語圧縮共有森の中に正しい構文構造が含まれないため，コーパス作成支援ツールで正しい構文構造を選択することができない．このような理由から，コーパス作成用の文法は，「この文法規則を必要とする構造が日本語文に存在するかもしれない」と直感的に感じた文法規則をすべて列挙したものになっている．

本章の残りでは，コーパスの編集，品詞や形態素区切りの自動変換と構文構造の付与について詳細を述べる．

B.1　コーパスの編集

RWCコーパスは新聞記事中の文を形態素列に区切り，品詞を割り当てたものである．新聞記事は，人間が読むことを前提に作成されているので，視覚的な分かりやすさを考慮して括弧，記号，空白を利用している．しかし，機械的に解析を行う上で，これらをそのまま扱うことは得策ではないため，その分析は構文解析とは別に行うべきである．そこで，今回のラベル付けでは，それらの情報を機械的に除去してから構文構造を付与することにしている．

以下に，前もって除去する部分を具体的に示す．

B.1.1　空白

日本語では，文章を書く場合，段落の先頭に1文字分の空白をあけることが一般的である．コーパスにも，段落の先頭の空白がそのまま残っている．しかし，この空白は構文構造とは関係がないので，構文構造を付与する前に除去する．

また，スポーツのスコアや俳句(川柳)などを記述する際，前後の行とレイアウトを揃えるために空白を入れることがある．

- 秋田工（秋田）　　反　9
  2　1　0　0　12　1　1　0　0　　7　19
  T　G　P　D　　前　T　G　P　D　　後　　計
  1　0　1　0　　8　2　2　0　0　14　22
  長崎北（長崎）　　反10

- 与野党もわからぬ妻と政談し　　相模原　水野タケシ
  好きなもの飲むときコップを大きくし　　久喜　　宮本愛子

これらは表の一種と考えるべきであり，構文解析が対象とするものではないため，途中に空白を含む文[1]はラベル付けの対象から除外する．

[1]RWCコーパスでは，表もすべて文として扱われ，品詞が割り当てられている．例に挙げたラグビーのスコアは，各行が1文として扱われている．

B.1.2　記号

文頭の記号は，記事の始まりやヘッドラインを表すことが多い．

- ◇ベストを尽くすだけ
- ○…女子飛び板飛び込みの元渕は三位で予選通過．
- ★日本大使館員，現地へ

これらも，文頭の空白と同じく構文構造とは関係がないので，構文構造を付与する前に除去する．

さらに，記号を箇条書きの区切りとして使用することもある．新聞記事では，紙面の都合上，箇条書きを一つずつ改行して表記せず，1行に続けて記述し，区切りを表すために記号を使用している．

- 具体策としては，風致地区条例や市街地景観条例などの現行規制の緩和▽一定規模以上の施設は都市計画法の地区計画精度を導入，合理的な土地利用を図る▽大学側との相談窓口の設置――が三本柱となる．

箇条書きの分析は構文解析とは別に行うべきである．そこで，途中に記号を含む文はラベル付けの対象から除外する．

B.1.3　括弧に囲まれた部分

RWCコーパスでは8種類の括弧が使われているが，除去せずにそのまま構文構造を付与すべきものと，構文解析とは別で扱うべきという考えのもと，除去してから構文構造を付与すべきものに分類することができる．ここでは，以下の3種類に分類し，扱いを述べる．

(1)　[] , 【】 , 《》 , ＜＞

(2)　()

(3)　「」 , 『』 , “”

(1)類の括弧は，文頭や文末で使われることが多く，その記事の見出しや記者名などを表す．

- ＣＤカード偽造の容疑で東大阪の男性を逮捕　【大阪】
- 【ロンドン５日黒岩徹】英国の閣僚が，愛人に子供を生ませたことが発覚し，辞任した．
- ［ロック情報］エアロスミスがやってくる
- ［みんなの広場］小学校焼却炉事故に思う

これらは構文解析とは別で扱うべきであり，文頭や文末に現れる場合に限り構文構造を付与する前に除去する．

(2) 類の括弧は，人名の読みや年齢，組織名の略称や別称など，付加的な情報を表すことが多い．

- 一九三五年，旧ソ連国境に近い旧満州（現中国東北部）のサンガに入植，牧場を経営していた岡部勇雄会長（８４）が，一緒に働いた白系ロシア人たちから教わった技術を伝えた．
- 主人と私は，昨日無事に娘の結婚式を終え，安堵（あんど）が不安と寂しさに変わっていく複雑な気持ちで，公園を歩いていました．
- アメリカンフットボール日本一をかけ学生と社会人の王者が戦う第十一回日本選手権（第四十七回ライスボウル）は三日，東京ドームで行われる．
- 横田町では，一九七六年に国営横田地区農地開発事業（三七五ヘクタール）がスタート．

これらの情報も，構文解析とは別で扱うこととし，構文構造を付与する前に除去する．この除去は機械的に行うが，以下の文で不都合が生じる．

- （５）スープを煮立て，ハクサイを加え（１）を直径２センチのダンゴにして入れる．

レシピにおいて，それ以前に作っておいたものを参照する際に，その番号 (括弧付き番号) を使用することがある．それを機械的に除去すると，文として成立しなくなってしまう．このような文はラベル付けの段階で作業者がチェックし，ラベル付けの対象から除外することにしている．

(3) 類の括弧は，発言内容，本や映画のタイトル，強調などで使われる．

- 三日午前八時四十分ごろ，通行人から「鉄棒に犬が鎖でつられ，ぶら下がっている」と同署に通報があった．
- 昨年，ベストセラーになったアメリカの小説「マディソン郡の橋」を読んだ．
- “保険”として外国籍を取得した後，香港に戻った．
- 巻末には同じ編者による『多話戯草』も収録している

これらは除去せずにそのまま構文構造を付与する．詳細は後述する (B.3.4節)．

B.2　品詞や形態素区切りの自動変換

先に述べたように，茶筌の品詞体系や形態素区切りは，構文解析には適さない部分がある．例えば，助詞は形態素ごとに細分化した方がよい[48]．また,「のだ」,「のです」のように助動詞的な役割を果たす形態素列は，1語にまとめておいた方が構文解析結果の曖昧性の抑制に役立つ．実際に変換する主な箇所を以下に示す．

数字列: 茶筌は2桁以上の数字は「2/0/0/5/年」のように，1文字ずつに区切る．しかし，構文解析において数字が1文字ずつに区切られている必要はないので，数字が連続する部分は1語にまとめ,「２００５/年」とする．

格助詞の細分化: 格助詞は形態素ごとに細分化する．

助詞相当句: 連語の格助詞のうち,「に関する」など名詞に係るものは連体助詞として区別する．また,「という」など引用の助詞「と」を含むものも区別する．

タ系語尾: 「(食べ) た」,「(食べ) て」,「(食べ) たら」,「(食べ) たり」などのタ系語尾[37]は直前の用言，助動詞に統合する．

アルファベット: アルファベットは一般名詞とする．

助動詞相当句: 「のだ」,「んだ」,「のです」,「んです」など形式名詞「の」,「ん」と助動詞「だ」,「です」が連続する場合は，それを結合する．

サ変名詞語尾: 茶筌はサ変名詞の直後の「する」,「できる」を動詞としているが，これをサ変名詞語尾とする[2]．

[2]茶筌では一般名詞としていても，その直後に「する」が結合してサ変名詞的な使われ方をすることは頻繁に起きるので，一般名詞の直後の「する」,「できる」もサ変名詞語尾とする．

文末記号: 疑問符，感嘆符など文末に出現しやすい記号を文末記号として他の記号と区別する [48].

連体副詞: 副詞の中には,「少し 前」のように名詞を修飾するものがある．これらを他の副詞と区別する (Schiehlen[47] の "Adverbial Classification" に類似).

複合動詞: 「吹き消す」のような複合動詞の中には，後半の動詞「消す」が自立動詞となっている場合がある．自立動詞の間には係り受け関係が生じ，非自立動詞は助動詞的な役割をすると決めているが，その場合,「消す」は非自立動詞である方が良い．そこで，複合動詞リスト [42] にある複合動詞について，後半の動詞が自立動詞となっている場合は非自立動詞に変換する．

B.3　構文構造の付与

基本的には，益岡ら [37] の定義に従いながら構文構造を決定していく．ただし，非終端記号名は品詞レベル，句レベル，節レベル，文レベルの四つのレベルに分けてつける．品詞レベルは，構文構造における最小単位であり，名詞，動詞，助詞，副詞などのまとまりを決定するレベルである (助動詞は直前の動詞などとまとめられる)．このレベルには，複合名詞などの複合語を含む．句レベルは，名詞句，動詞句など連体修飾や連用修飾を受けるまとまりを決定するレベルである．修飾を受けた後の全体のまとまりも句とする．節レベルは，句を修飾するまとまりを決定するレベルである．動詞句などがそのまま連体節，連用節に置換される場合と,「Aが」のように名詞句と助詞が結合して節になる場合がある．文レベルは，単文，複文，重文のまとまりを決定するレベルである.「１０時に出発する と言った」のように，引用の助詞が受けるまとまりも文とする．

構文構造は，節が句を修飾して句になるという法則に従って付けられる．図B.1に，構文構造の例を示す．しかし，一部の特殊な構造を扱うために，例外的なラベル付けを行っている．以下で，そのラベル付け方法を述べる．

B.3.1　「など」の扱い

助詞「など」が末尾に出現する節の係り先は，ラベル付けの際に迷いやすい．例えば，以下の文を考える．

(1) いま話題の食管制度など 食料問題を扱った本

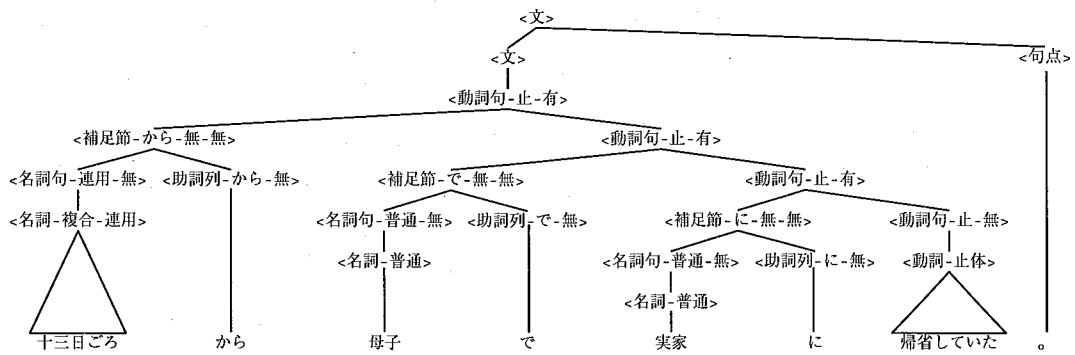


図 B.1: RWCコーパスに付与する構文構造の例

(2) 体内の水分や栄養管理，電解質の補正などスタッフはほとんど眠らなかった

(3) 豊富な資源を抱えるなど明るい将来への展望を持つ

例文1は「食管制度」を修飾する連体修飾節，例文2，例文3は，それぞれ「眠らなかった」,「持つ」を修飾する連用修飾節である．このように,「など」が末尾に出現する節は，連体修飾節になる場合と連用修飾節になる場合がある．

京大コーパス [31] では，以下のような方針となっている．

- 「名詞＋など」の場合，対応する名詞と同格関係になる．適当な名詞が文中に存在しない場合は，用言に係る．
- 「用言＋など」の場合，用言に係る．

本研究で作成するコーパスも，この方針に従う．ただし，本研究では，名詞の同格関係は並列関係と同様に扱う (すなわち，変更方針によって，連体修飾関係として扱われることになる)．

B.3.2　「うち」,「ほか」の扱い

(形式) 名詞「うち」,「ほか」でも「など」と同様の現象が起きる．

(1) １０人のうち２人が欠席した

(2) １０人のうち欠席したのは２人だ

例文1は「2人」に係る連体修飾節，例文2は「2人だ」に係る連用修飾節となる．

「うち」や「ほか」が末尾に出現する場合のラベル付け方針は以下のようにする．

- 「名詞＋の＋うち(ほか)」,「連体詞＋うち(ほか)」は対応する名詞に係る．適当な名詞が文中に存在しない場合は，用言に係る．
- それ以外の場合，用言に係る．

B.3.3　「ら」,「たち」の扱い

接尾語「ら」,「たち」でも同格関係が生じる場合がある．

- 教師ら六十人が参加

京大コーパスでは,「ら」,「たち」を手掛り語として同格を扱っている．本研究で作成するコーパスも，この方針に従う．

B.3.4　括弧の扱い

EDRコーパスに対するラベル付けでは，括弧を含む文は除外していた．しかし，RWCコーパスでは約46%の文が括弧を含み，より多くの文を扱うためには括弧の扱いを検討する必要がある．そこで，鉤括弧，二重鉤括弧，ダブルクォートの3種類の括弧を扱うことにする(図B.2)．ただし，句レベルのまとまりを囲む括弧のみを対象とし，以下の場合はラベル付け対象外とする．

節レベルのまとまりを囲む場合:　侵攻作戦を「数日以内に」開始する

括弧外の節が括弧内の末尾以外の句を修飾する場合:　アジア市場との「相互補完的な役割を果たすべきだ」と，協調関係を強める努力を要請している[3]

B.3.5　名詞終止文，助詞終止文，副詞終止文，連用終止文

通常の文は用言の終止形で終わるが，それ以外の語で終わる文も存在する．

(1) 重要なのは時間を守ること．

[3]「クロアチア国防省はクライナ共和国領内から「ロケット二発が発射された」との声明を発表した」のように，括弧内の末尾の語を修飾する場合は除外せずに構文構造を付与する．

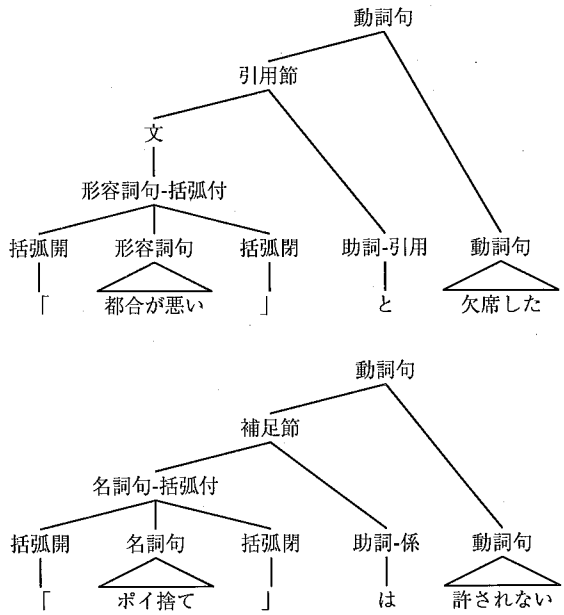


図 B.2: 括弧を含む文の構造

(2) 試合開始は午後六時から.

(3) 同長官の訪中は初めて.

(4) おばさんがお手伝いしてくれるから.

本研究で使用するコーパスでは，名詞句，補足節，副詞節，連用節で終わる文を，それぞれ名詞終止文，助詞終止文，副詞終止文，連用終止文としている．ただし，扱う文は，末尾に助動詞「だ」が省略されていると判断できる場合に限り，それ以外の用言が省略されている場合はラベル付け対象外とする．

- 明日は東京へ. (動詞「行く」が省略されている)

B.3.6　その他の特殊構造

前節までに述べた点以外に特殊な構造を付与する場合を述べる．

AからBまで(へ):　東京から大阪まで の距離はどのくらいですか (図B.3(a))

[数量詞] に [数量詞]:　その会議は 1年に1回 の頻度で開かれる (図B.3(b))

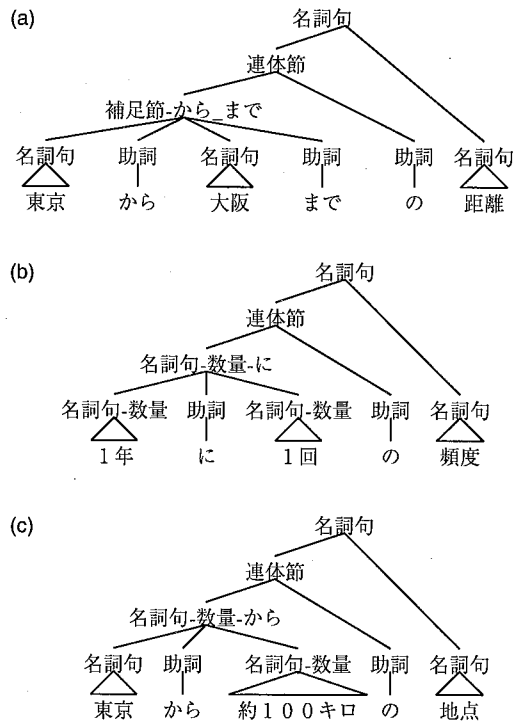


図 B.3: 特殊な構造

Aから [数量詞]: 東京から１００キロの地点にいる (図B.3(c))

ただし，この方針は直後に「の」などの助詞が結合する場合に限る．つまり，直後に助詞が出現しない以下の場合は，この方針は適用されず，各連用修飾節が別個に用言を修飾する構造となる．

(1) 東京から大阪へ向かう

(2) その会議は１年に１回開かれる

(3) 東京から１００キロ離れた地点にいる

付 録C　京大コーパスを正解データとした場合の評価結果に関する考察

第5.2.3節において，RWC変更後コーパスから抽出した文法で京大コーパス中の文を構文解析した結果の文節係り受け精度を示した．しかし，以下の2点について，EDRコーパスを使用した場合 (第5.2.2節) より結果が悪くなった．

(1) 評価用データの文数が大幅に減少

(2) 文節区切りが一致しない文が多い

本章では，結果が悪くなった要因について述べる．

C.1　評価用データの文数が大幅に減少した要因

当初，評価用データとして8,835文を用意したが，表5.6 (49ページ) に示したように，評価用データの文数は8,835文から3,764文まで減少した．その理由の一つとして，品詞体系の自動変換の精度が低いことが挙げられる．そこで，品詞体系の自動変換の精度を調べた．ただし，正解データが存在しないため，以下の2通りの方法で変換精度を求めた．

(1) 京大コーパスを茶筌で解析した結果を正解データとし，京大コーパスの品詞体系をRWCコーパスの品詞体系に変換した結果を評価

(2) RWCコーパスを正解データとし，RWCコーパスをJUMANで解析した後，RWCコーパスの品詞体系に変換した結果を評価

評価に使用したコーパスは，係り受け精度の評価に使用するデータだけでなく，京大コーパス，オリジナルのRWCコーパス (1994年版3,000記事) 全文である．ど

表 C.1: 品詞体系の変換精度の推定

	形態素区切り		主品詞のみ		主品詞＋細品詞	
	再現率	適合率	再現率	適合率	再現率	適合率
(1)	87.51%	91.42%	85.40%	89.22%	54.69%	57.13%
茶筌	99.13%	95.62%	98.75%	95.26%	84.02%	81.05%
(2)	86.75%	89.01%	84.16%	86.36%	68.38%	70.17%
JUMAN	97.69%	95.97%	95.91%	94.22%	94.23%	92.57%

ちらの評価方法も，形態素解析の精度が影響するため厳密な評価にはならないが，形態素解析の精度と比較することで，簡単な推定はできると考える．結果を表C.1に示す．(1) と (2) は，上述の方法1と方法2の結果であり，「茶筌」と「JUMAN」は，それぞれRWCコーパスを茶筌で解析した結果と京大コーパスをJUMANで解析した結果を表す．また，「形態素区切り」，「主品詞のみ」，「主品詞＋細品詞」は，それぞれ形態素区切り，主品詞，細品詞が一致していれば正解とした場合の結果である．表C.1より，この自動変換の精度は，形態素区切りは90%程度であるが，細品詞まで正解するものは70%程度しかないことが分かる．実際，京大コーパス中の100文について人手で変換結果を確認したところ，解析に失敗した文は50文あり，そのうち42文は変換誤りが原因であった．当然のことであるが，最終的に評価用データとして残った3,764文についても，変換誤りが含まれている可能性はあるが，今回はこのデータで評価を行った．

解析結果が一つも出力されない文が多い別の理由として，文境界の認定の違いがある．RWCコーパスでは茶筌の解析結果をそのまま利用しているが，基本的に，句点で文が区切られると判断される[1]．一方，京大コーパスでは，以下の文のように，括弧内に複数の文がある場合などにおいて，句点が出現しても文境界としない場合がある．

- 党内の議論や党関係者の意見は「保守二党論はよろしくない．市民の側に立った平和と民主主義を担う政党が必要」というものだ．

この文は，京大コーパスでは1文となるが，茶筌は2文と判断する．RWCコーパスから抽出した文法は，文の途中に句点が入ることを認めておらず，途中に句点を含む文を解析すると，解析に失敗する．京大コーパス中の8,835文のうち，句点

[1]通常，茶筌は改行をもって文の区切りと判断するが，実行時にjオプションを指定した場合は，句点，感嘆符，疑問符を文の区切りと判断する．

を2個以上含む文は336文あった.

C.2　文節区切りが一致しない文が多い要因

表5.7に示したように，構文解析結果の文節区切りが正解データと一致しなかった文が多い．これは，正解データである京大コーパスと文法抽出やPGLRモデルの学習に使用したRWC変更後コーパスで文節区切りの決め方が異なることが要因の一つとして挙げられる．以下に例を挙げる．

(1) 三十一日午後九時四十三分ごろ，北海道桧山管内奥尻町で，震度3の地震があった.

(2) 船長ら乗組員二十七人は同船を放棄した

(3) 宝剣山荘の支配人が六人が下山するのを見送った

例文(1)の「三十一日午後九時四十三分ごろ,」は，京大コーパスでは「三十一日」,「午後」,「九時」,「四十三分ごろ,」の4文節に分かれるが，RWC変更後コーパスでは1文節としている．例文(2)の「乗組員二十七人は」も，京大コーパスでは「乗組員」と「二十七人は」の2文節に分かれるが，RWC変更後コーパスでは1文節としている．逆に，例文(3)の「下山するのを」は，京大コーパスでは1文節としているが，RWC変更後コーパスでは「下山する」と「のを」の2文節としている．

複合名詞内の構造は，文節係り受け構造とは無関係であるとして，今回の実験では無視している．しかし，(1)や(2)の例を見ると，実際には，複合名詞内の構造と文節係り受け構造は無関係ではないことが分かる．この問題は，複合名詞内の構造を厳密に解析し，文節区切りを与えることで解決できる可能性がある．今後，複合名詞内の構造の解析手法を考える際には，文節区切りが入る可能性も考慮する必要がある．