文法抽出のための日本語構文構造付きコーパスの開発

野呂 智哉 † 小池 千万人 ‡ 橋本 泰一 † 徳永 健伸 † 田中 穂積 †

† 東京工業大学 大学院情報理工学研究科 ‡ 東京工業大学 工学部

{noro,chimato,taiichi,take,tanaka}@cl.cs.titech.ac.jp

1 はじめに

構文構造付きコーパスは，自然言語処理において重要な知識資源の一つである．その使用目的は，(1) 確率モデル等のパラメータの学習，(2) 解析システム等の精度評価，(3) 用例に基づく解析のための直接利用，(4) 文法，共起情報，格フレーム等，他の知識資源の自動獲得 — 等がある．構文構造付きコーパスからの文法抽出については，Charniak が，句構造付きコーパスである Penn Treebank コーパス [9] から抽出した文脈自由文法 (tree-bank grammar) による解析精度が比較的良いことを示し [1]，それ以降，構文木 (句構造) 付きコーパス[*1]から抽出した文法に関する様々な研究が進められ，他言語においても Penn Treebank コーパスと同様の構文木付きコーパスが開発されている．しかし，日本語では Penn Treebank コーパスのような構文木付きコーパスが存在せず，他言語による研究成果を日本語に適用することが困難となっている．そこで，まず始めに，日本語の構文木付きコーパスを開発する必要がある．

しかし，仮に構文木付きコーパスが存在したとしても，そのコーパスから抽出した文法 (tree-bank grammar) [*2]を使用して構文解析を行うと，しばしば膨大な量の構文解析結果 (曖昧性) [*3]が出るという問題がある．曖昧性の増大は，解析精度の低下，解析所要時間や使用メモリ量の増大の要因となる．この問題の最大の要因は，構文木付きコーパスを開発する際，抽出した文法による曖昧性を考慮していないことである．曖昧性を抑えるためには，曖昧性を抑えるためにはどのような構造をつけるべきかを，コーパス開発段階で検討する必要がある．

構文木付きコーパスを開発する際，各文に対して意味を考慮した構造を付与することが一般的である．このようなコーパスから抽出した文法を使用して構文解析を行うと，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．その理由は，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からである．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) [5] を想定し，構文構造を制限することで構文解析の段階の曖昧性を極力抑えることを考え，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．

我々は，文法抽出のための構文木付きコーパスを，(1) 既存のコーパスから文法を抽出，(2) 曖昧性を増大させる要因を分析，(3) コーパス変更方針を作成，(4) コーパスを変更し，文法を再抽出，(5) 手順 (2)〜(4) を繰り返す — という手順で開発を進めている (図 1)[11]．本稿では，我々のコーパス作成方針を紹介し，作成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，その方針が容易に適用できることを示す．

2 コーパス作成方針

コーパス作成方針を決定する際に検討すべき問題は，以下の 2 点である．

構文情報の欠落: 抽出した文法が構文解析に必要な構文情報を持たないことにより，曖昧性が発生する．構文的に誤った構造が多く，不必要に曖昧性を増大させる要因となる．

意味的曖昧性: 曖昧性の解消に，構文情報だけでなく，意味情報も必要とする場合がある．意味情報を利用しない構文解析では，この曖昧性を解消することは困難である．

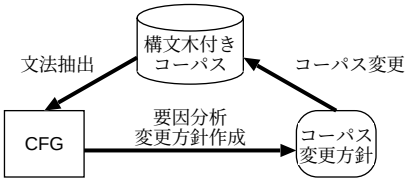

図 1 構文木付きコーパス開発手順

[*1] 本稿では，句構造付きコーパスを，構文木付きコーパスまたはコーパスと呼ぶ．

[*2] 本稿では，tree-bank grammar[1] を，単に文法と呼ぶ．

[*3] 本稿では，構文解析結果の曖昧性を，単に曖昧性と呼ぶ．

前者の解決には，どの構文情報が必要かを検討し，その情報を非終端記号に追加する．後者の解決には，意味情報を使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で表現する．すなわち，このコーパスから抽出した文法による構文解析結果では，一部の意味的曖昧性は区別されない．意味的曖昧性は，構文解析後の意味解析で解消することになる．

主なコーパス作成 (変更) 方針は以下の通りである．

用言の活用形: 日本語において，動詞句等が連体修飾するか連用修飾するかは，用言の活用形や末尾の助詞で決まる．用言の活用形に関する情報を上位ノードに引き継ぐことで，用言が連体形であるにも関わらず連用修飾句となるような誤った構造を生成することを防げる．この考えは，Klein ら [6] の”SPLIT-VP”，Schiehlen[12] の”Verb-Form” と類似している．

複合名詞内の構造: 複合名詞内の構造の曖昧性は，単一の構造で表現し，構文解析の段階では区別しない．この考えは，白井ら [13] の「同一品詞列の取り扱い」と同じであるが，我々の方針では，名詞列だけでなく，接頭語，接尾語を含む場合も同様に単一の構造で表現する．

連用修飾句の係り先: 連用修飾句 (助詞句, 副詞, 接続詞，従属節等) の係り先の曖昧性は，従来通り別の構造として区別する．

連体修飾句の係り先: 連体修飾句の係り先の曖昧性は，複合名詞内の構造の曖昧性と同様，単一の構造で表現する．ただし，「10 年の 歴史を持つ祭り」の場合，「10 年の」が「歴史」を修飾するか「祭り」を修飾するかで，「持つ」を修飾する句の範囲が変わる (前者の場合の連用修飾句の範囲は「10 年の歴史を」であるのに対し，後者の場合は「歴史を」になる)．連体修飾句の係り先が変わることで，周辺の連用修飾句の範囲が変わる場合は，従来通り別の構造として区別する．

並列構造: 並列構造の曖昧性 (注目する二つの句が並列関係にあるか否か，あるいは，注目する句と並列関係にある句はどれか) は単一の構造で表現する．並列名詞句は連体修飾関係, 並列述語句, 並列助詞句は連用修飾関係として扱う．この考えは，Schiehlen[12] の”Coordinated Categories” と類似している．

我々は，意味的曖昧性を連用修飾関係と連体修飾関係に大別している．これは，日本語において，連用修飾関係の解析と連体修飾関係の解析では，解析手法が異なると考えるからである．例えば，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，格や呼応関係等を考慮しながら進める．一方，連体修飾関係の解析は，体言を中心に進めることになる．この方針で作成したコーパスから抽出した文法で構文解析を行うと，連用修飾関係の曖昧性は構文解析結果の曖昧性として現れ，連体修飾関係の曖昧性は区別されない．構文解析後の意味解析では，まず，連用修飾関係の解析によって，複数の構文解析結果の中から一つの解析結果を選択し，連体修飾関係の解析によって，その構文木が包含する連体修飾関係の曖昧性の中から一つを決定することになる (図 2).

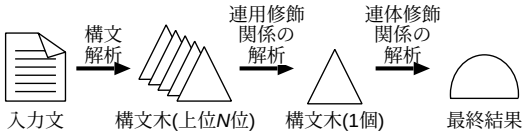

図 2 想定する意味解析の流れ

表 1 構文解析結果の数

	文法規則数	構文解析結果数
EDR	1,949	9.355×10^5
RWC	2,565	9.599×10^4

3 評価実験

我々は，前節で述べた方針の検討を EDR コーパス [10] で行い，その方針による構文木の付与の有用性を，構文解析結果の曖昧性と解析精度の二つの観点から確認した [11]．本節では, RWC コーパス [2] で同様の実験を行い，我々の方針が他のコーパス (異なる品詞体系を持つコーパス) に対しても容易に適用できることを示す．

RWC コーパスは品詞タグ付きコーパスであり，品詞体系は IPA 品詞体系を利用している．EDR コーパスと違い，RWC コーパスには構文情報がないため，第 1 節で述べたコーパス開発手順のように，構文木付きコーパスを出発点とすることができない．そこで，EDR コーパスを対象に作成したコーパス作成方針をそのまま適用することで，手順 (4) から開始し，直接，構文木付きコーパスを作成した．文数は 16,421 文 (1 文あたり平均 21.71 形態素) である．

表 1 に，作成した構文木付きコーパスから抽出した文法で，全 16,421 文を構文解析した結果を示す．参考として，EDR コーパス 8,911 文に対し，我々の方針によって構文木を付与したコーパスから抽出した文法による結果を併記する．ただし，構文解析には MSLR パーザ [14] を使用した．コーパス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，曖昧性は十分抑えられている．

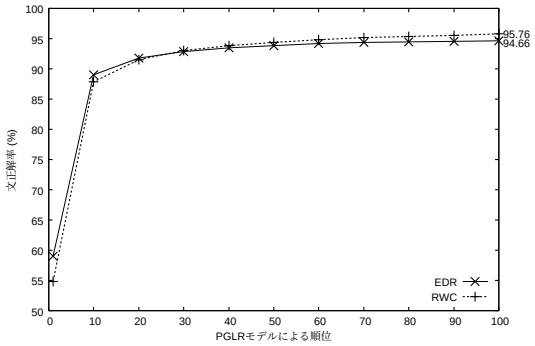

図 3　文正解率

表 2　被覆率と再現率

	被覆率	再現率
EDR	97.32%	95.88%
RWC	98.38%	97.18%

次に，構文解析結果を確率一般化 LR(PGLR) モデル [4] でランク付けし，解析精度を調べた．ただし，評価は 10 分割交差検定で行った (文法は学習用データのみから抽出した)．抽出した文法による被覆率，再現率，文正解率を表 2，図 3 に示す．ただし，被覆率，再現率，文正解率は以下のように定義する．

被覆率:　少なくとも 1 個の解析木を出力する文の割合
再現率:　出力した全ての解析木の集合の中に，正解の構造と完全に一致するものが含まれる文の割合
文正解率:　上位 n 個の解析木の集合の中に，正解の構造と完全に一致するものが含まれる文の割合

これらも，EDR コーパスの場合と大差のない結果が得られている．

我々のコーパス作成方針では，一部の曖昧性を単一構造で表現することにより，構文解析結果の曖昧性を抑えているため，文正解率が向上することは当然であるという疑問が残る．さらに，日本語の構文解析では，文節係り受け解析による係り受け精度を比較することが多く，上述の結果では他の手法との比較が難しい．そこで，EDR コーパスによる実験 [11] と同様，意味的に正しい構造と正解データとした場合の文節係り受け精度を調べた．

EDR コーパスによる実験では，我々の方針による変更前の構文構造を正解データとして文節係り受け精度を調べた．しかし，今回は「変更前」のコーパスが存在しないため，京大コーパス [8] 中の 3,764 文を評価用データとして評価を行う (図 4)．ただし，京大コーパスの品詞体系は RWC コーパスと異なるため，構文解析の前に，品詞の自動変換を行っている [3].

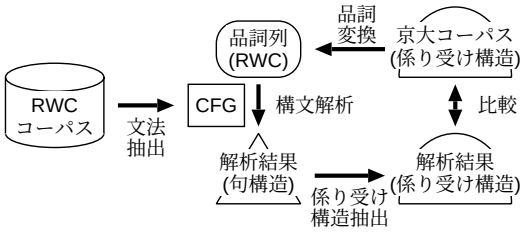

図 4　文節係り受け精度の計算

表 3　文節係り受け精度

	係り受け精度	文正解率	文節不一致
EDR	91.32%	61.54%	9
RWC	85.76%	52.38%	1,280

本実験では，文法抽出，PGLR モデルの学習には構文木を付与した全 16,421 文を利用し，PGLR モデルによる生成確率が 1 位である解析木の係り受け精度を調べた．我々がコーパスに付与した構文木は句構造であるため，解析結果の句構造から文節係り受け関係を抽出する．ただし，構造を制限している連体修飾関係については，曖昧性を含む連体修飾句は最も近い名詞を修飾することにする[*4]．結果を表 3 に示す．参考のため，EDR コーパスの場合の結果を併記する．ただし，係り受け精度，文正解率は以下のように定義する．

係り受け精度:　全ての係り受け関係のうち，正しい係り受け関係の数の割合
文正解率:　1 文中の全ての係り受け関係が正しく決定された文の割合

「文節不一致」とは，文節区切りが正解と一致しなかった文の数を表す．EDR コーパスの場合と比べ，結果が悪くなっているが，その原因として，(1) 品詞の自動変換の精度が低い，(2) 京大コーパスと我々の方針で文節区切りの決定方針が異なる[*5]—等が考えられる．

我々のコーパス作成方針は，抽出した文法による構文解析後に意味解析を行うことを前提としている．そこで，予備実験として，構文解析後の意味解析が最適に行われた場合，文節係り受け精度はどこまで向上し得るかを調べた．第 2 節で述べたように，我々の方針で作成したコーパスから抽出した文法で構文解析を行うと，連用修飾関係の曖昧性は構文解析結果の曖昧性として現れ，

[*4] 並列構造の曖昧性は無視し，評価の対象外とする．複合名詞内の構造は，文節の係り受けとは無関係であるため，考慮しない．
[*5] 例えば，京大コーパスでは「2005 年 3 月 15 日」を「2005 年」，「3 月」，「15 日」の 3 文節とするが，我々は 1 文節としている．

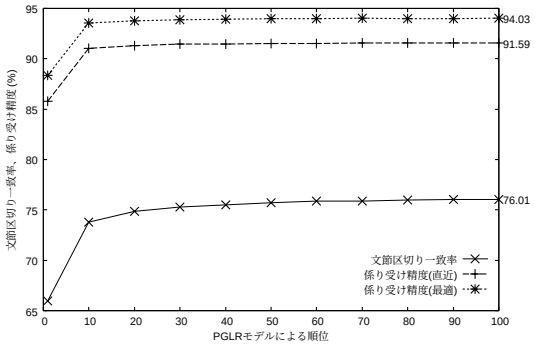

図 5　意味解析が最適に行われた場合の係り受け精度

連体修飾関係の曖昧性は区別されない．予備実験では，PGLR モデルによる生成確率の上位 n 個の解析結果について文節係り受け関係を抽出し，係り受け関係が正解と最も一致するものを選択した場合の係り受け精度を求める．結果を図 5 に示す．ただし，「係り受け精度 (直近)」は，連体修飾句の係り先を最も近い名詞とした場合の係り受け精度を，「係り受け精度 (最適)」は，連体修飾関係の曖昧性解消が最適に行われた場合の係り受け精度を表す．「文節区切り一致率」は，1 文中の全ての文節区切りが一致した文の割合を表す．結果より，文節区切り一致率は 10% 程度向上し，PGLR モデルによる生成確率の上位 100 個について意味解析が最適に行われた場合の係り受け精度は 94% に達する．使用しているコーパスや実験規模が異なるため，公平な比較にはならないが，この文節係り受け精度は，最大エントロピー法や Support Vector Machine を利用した他の文節係り受け解析 [7, 15] の精度を上回る．この結果より，今後，意味解析を進めることで，従来の文節係り受け解析手法の精度を上回ることが期待できる．

4　おわりに

本稿では，文法抽出のための日本語の構文木付きコーパスの作成方針を紹介した．我々は，EDR コーパスを利用してコーパス作成方針の検討を行ったが，品詞体系の異なる RWC コーパスに対しても，同じ方針を要因に適用できる．さらに，作成したコーパスから抽出した文法で構文解析を行うと，構文解析結果の曖昧性が抑えられるだけでなく，本格的な意味解析により，他の係り受け解析による文節係り受け精度を上回る可能性が十分あることを示した．

第 2 節で述べたように，我々のコーパス作成方針では意味的曖昧性を 4 種類に分類している．構文解析後の意味解析ではこれらの曖昧性の解消を分けて行うことを想定しているが，その手法の検討は今後の課題となる．

参考文献

[1] Eugene Charniak. Tree-bank grammars. In *AAAI-96*, 1996.

[2] Koichi Hashida, Hitoshi Isahara, Takenobu Tokunaga, Minako Hashimoto, Shiho Ogino, and Wakako Kashino. The RWC text database. In *LREC '98*, 1998.

[3] 橋本泰一, 野呂智哉, 徳永健伸, 田中穂積. 品詞タグ付きコーパスのための品詞体系変換ツール: Conbu. 言語処理学会第 11 回年次大会, 2005.

[4] 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, 1998.

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

[6] Dan Klein and Christopher D. Manning. Accurate unlexicalized parsing. In *ACL 2003*, 2003.

[7] 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, 2002.

[8] 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, 1997.

[9] 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, 1993.

[10] 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.

[11] 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. 自然言語処理, Vol. 12, No. 1, 2005.

[12] Michael Schiehlen. Annotation strategies for probabilistic parsing in German. In *COLING 2004*, 2004.

[13] 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, 1997.

[14] 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, 2000.

[15] 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, 2000.