

精細な文法に基づいたツリーバンク「檜」の構築

Francis Bond* 藤田 早苗* 橋本 力† 成山 重子‡§

Eric Nichols§ 大谷 朗¶ 田中 貴秋*

* 日本電信電話株式会社 NTT コミュニケーション科学基礎研究所

† 神戸松蔭女子学院大学大学院 ‡ メルボルン大学

§ 奈良先端科学技術大学院大学 情報科学研究科 ¶ 大阪学院大学 情報学部

{bond, sanae, takaaki}@cslab.kecl.ntt.co.jp chashi@sils.shoin.ac.jp

{eric-n, shigeko-n}@is.aist-nara.ac.jp ohtani@utc.osaka-gu.ac.jp

概要

我々は、意味理解を行える自然言語処理のための言語知識ベースを目指して、ツリーバンク「檜」を構築している。「檜」は、日本語 HPSG の解析に基づいた詳細な統語情報、意味情報を持つ。知識源に用いることを視野に入れ、ツリーバンクの最初の対象として辞書の語義文を選んだ。本稿では、統語・意味タグ付けと文法拡張を並行して行うことにより、効率的にツリーバンクを構築可能であることを示す。また、ドメイン依存/非依存の言語現象に適用するための文法拡張についても述べる。

キーワード: コーパス構築、意味表現、日本語、HPSG、文法拡張

Development of the Hinoki Treebank Based on a Precise Grammar

Francis Bond* Sanae Fujita* Chikara Hashimoto† Shigeko Nariyama‡§

Eric Nichols§ Akira Ohtani¶ Takaaki Tanaka*

*NTT Communication Science Laboratories, NTT Corporation

†Kobe Shoin Graduate School ‡The University of Melbourne

§Graduate School of Information Science, Nara Institute of Science and Technology

¶Faculty of Informatics, Osaka Gakuin University

Abstract

In this paper we describe the construction of the Hinoki treebank and the Japanese HPSG grammar it is based on. We then show preliminary results from building a stochastic parse-ranking mechanism, and outline plans for further work.

Keywords: Treebanking, Semantic representation, Japanese, HPSG, Grammar Engineering

1 はじめに

我々は、統語情報と意味情報を統合して扱える自然言語処理を目指して、「基本語彙知識ベース」の構築を進めている [1]。究極の目標は機械に自然言語を理解させることであり、そのためにテキストを統語的に解析するだけでなく、意味的に解析

し、意味情報を獲得することを目指している。さらに、出力された意味表現によって、システムが内部に持つ意味モデルを更新できるようにすることが理想である。その第一歩として、語の基本的な語義に関する情報を記述した基本語意味データベース Lexeed[1] と語の文中での統語情報・意味情報を記述した新しい構文木コーパスであるツリーバンク「檜」の構築を行っている。本稿では、このうち、ツリーバンクを実際に構築する過程の詳細について述べる。ツリーバンク構築の動機と応

*本稿は 03 年 7 月より NTT コミュニケーション科学基礎研究所を中心に活動してきた「檜ツリーバンクプロジェクト」について、その進捗および研究成果の一部を報告するものである。

用については文献 [2] を、基本語意味データベースについては文献 [1] を参照されたい。

近年、シンボリックなアプローチと統計的なアプローチを融合することによって、自然言語処理の諸問題に対する解決技術は飛躍的な進歩を遂げている。例えば、構文解析ではシンボリックな文法を統計モデルと融合することにより精度を顕著に向上させている [3]。また、質問応答では意味タグを統計モデルと融合した結果が報告され [4]、語義曖昧性解消の問題においても統計モデルの有効性が示されている [5]。しかしながら、現在のところ、シンボリックな文法と統計モデルに加えて、意味情報を統合した例は報告されていない。Kleinら [6] は、少数の機能語から構築した語彙化モデルによって統計的な構文解析の精度が大幅に向上させられることを示している。しかし、データスパースネスの問題から、単語の直接的な共起関係は有効な手がかりにならないことも多い。例えば、統計ベースの構文解析器の代表的な訓練/テストデータである Penn treebank では、*stocks* (株) と *skyrocket* (急騰) は同時に出現しない。そこで、それぞれの上位概念である *capital* (資本)、*move upward* (上昇) が共起することを利用すれば有効であると期待されるが、実際にはこのような意味情報は十分に利用されていない。

意味情報が利用されていない大きな原因として、二つ考えられる。一つの原因は、単にリソースが存在しないことである。統語情報と意味情報を合わせ持ったコーパスが存在しないために、両方の情報を用いた統計モデルが構築できない。もう一つの原因は、どのような意味情報が必要であり、どのように獲得すればよいかがいかに依然として明確でないことである。

我々は、これらの問題を解決する過程として次の3つの段階を考えている。第1段階では、日本語の意味情報を持ったツリーバンク「檜」を構築する。「檜」の構築には、基本語意味データベース Lexeed[1] を利用する。このツリーバンクを元に統計モデルを構成し、新たにシソーラスを構築するのに利用する。第2段階では、統語情報と意味情報を統合した言語モデルを構築する。これには、辞書の語義文に語義単位で付与した意味タグとシソーラスを用いる。さらに、選択制限、特質構造 (qualia structure) などの情報を含んだ詳細なオントロジの構築を目指す。第3段階でこれらの

辞書とオントロジを辞書収録外の語に拡張する方法を検討する。

本稿では、第1段階のツリーバンク構築過程の詳細について述べる。特にツリーバンクの構築の方針と方法の詳細、解析文法の拡張について述べる。2章で、ツリーバンクの基盤となる Lexeed について述べる。3章で、ツリーバンク構築の方針と方法について述べ、4章で、ツリーバンク構築のために行った文法拡張について報告する。5章でツリーバンクを利用した言語モデル構築の予備実験について述べ、6章でまとめと今後の方向性を述べる。

2 日本語意味データベース: Lexeed

Lexeed[1] は、一般的に使用されている語に対して意味、統語、語用情報を付与した日本語の意味データベースである。Lexeed に収録される語は、一般の日本人の語に対するなじみの度合を表す「単語親密度」[7]に基づいて選定されている。単語親密度は語に対するなじみの度合を1から7の実数で表し、7が最もなじみのある語であることを示す。このうち単語親密度が5以上である28,000語を基本語と定義し [8]、Lexeed に収録している。多くの語が複数の語義を持つので、語義の総数は46,347である。全ての語義文は、基本語のみを用いて書き換えられており、辞書の中で自己完結するように構成されている。

見出し語「ドライバー」の Lexeed の記述を図1に例示する。「ドライバー」は単語親密度が6.5であり、3つの語義を持っている。語義文1'は、語義文1の「ねじまわし」が基本語でない語のために基本語のみを使って書き換えられた文である。Lexeed 全体では、81,000文の語義文がある。第1段階での目標は、この81,000文のツリーバンクを構築することである。

3 ツリーバンク「檜」

3.1 基本方針

ツリーバンク「檜」は、Oepenら [9] が構築した動的かつ詳細な情報を持つ英語のツリーバンク Redwoods と同様の考え方に基づいて構築している。Redwoods は主辞駆動句構造文法 (Head-driven Phrase Structure Grammar, HPSG)[10, 11] による解析に基づいて構築され、文法を精錬

見出し語	ドライバー
単語親密度	6.5 [1-7]
語義 1	語義文 1 <u>ねじまわし</u>
	語義文 1' <u>ねじを差し入れたり、抜き取ったりする</u> <u>道具</u>
語義 2	語義文 2 <u>自動車</u> を <u>運転</u> する <u>人</u>
語義 3	語義文 3 <u>ゴルフ</u> で、 <u>遠距離</u> 用の <u>クラブ</u>
	文 2 <u>一番</u> <u>ウッド</u>

図 1: Lexceed の収録語の例:見出し語「ドライバー」

することによりツリーバンクの精度も動的に向上する。

我々がこの方法を採用したのには幾つかの理由があるが、最も重要な理由は、統語情報と意味情報を統合して詳細に記述できるという点である。ツリーバンクには、HPSG に基づいた統語-意味情報をもつ複数の解析結果候補から作業者が選んだ最適な解析結果が蓄積される。この解析結果の集合から、様々な粒度のあらゆる情報を獲得することができる。特に、伝統的なラベル付きの構文木で表されるような統語構造や、単語間の依存関係、さらに、最小再帰意味論 (Minimal Recursion Semantics: MRS)[12] による詳細な意味表現などが含まれている。解析器の出力を利用したツリーバンクの構築は、黒橋ら [13] も行っている。黒橋らの方法では構文解析器の選択した解析結果を作業者が修正していくのに対し、我々の方法は複数の解析結果候補の中から適切なものを作業者が選択する点が異なる。

もう一つの重要な理由は、大規模で十分な頑健性を持つ日本語 HPSG 文法 JaCY [14] が利用できる点である。加えて、文法開発のためのオープンソースのツール類が広範囲にわたって整備されている。JaCY を辞書語義文のドメインに適応させるため、LKB(Linguistic Knowledge Builder)[15] を全般的に使用した。LKB は、HPSG を含む Typed Feature Structure に基づいた文法の開発環境であり、解析器やデバッグツールを含んでいる。レグレッションテストやツリーバンクの構築、知識獲得 [2] などで、大量のテキストを解析する際には、LKB の高速版解析器である PET[16] を利用した。実際のツリーバンク構築には、これらのツールをプロファイリング環境である [incr tsdb()] [17] 上

で統合して行った。[incr tsdb()] は、ツリーバンクの構築、更新ができるだけでなく、同じ文法の異なるバージョン間の比較や同じ文法を異なる解析器で使った結果の比較を簡単に行ったり、ツリーバンクからの統計モデルの学習を行うことができる。ここで得られた統計モデルは、PET の解析候補を尤度順に順位づける際に利用できる。

3.2 構築の方法

ツリーバンクの構築は 2 段階で行われる。第 1 段階で、解析器によりコーパスを解析し複数の解析結果候補を出力する。「檜」の場合は、解析器に PET を使用し、日本語文法 JaCY に基づいて解析した。解析結果は、木構造で表された統語情報と MRS で表された意味情報を持っている。第 2 段階で、作業者が解析結果候補の中から正しいものを選択することによって、各文にタグ付けを行う (図 2)。場合によっては全ての候補が誤りであると判定する。この解析結果候補の選択は [incr tsdb()] 上で行った。候補が適切であるかの判定は、(1) 構文木の形、(2) 統語ラベル、すなわち適用された文法規則、(3) 意味表現 (MRS) の妥当性を見て行われる。

解析結果は、構文木と、構文木を構築するために使用された文法規則の情報から成る。これを解析木と呼ぶ。図 3 は「ドライバー」(語義 2) の語義文の解析木の例である。解析木のルート及び中間ノードは解析時に適用された文法規則を表している。構文木の形が同じ候補でも、解析木が異なることが有り得る。

候補選択は、解析木に適用されている文法規則の差異ごとに妥当性を判断することによって行う。実際の作業では、ラベル付きの構文木と、解析結

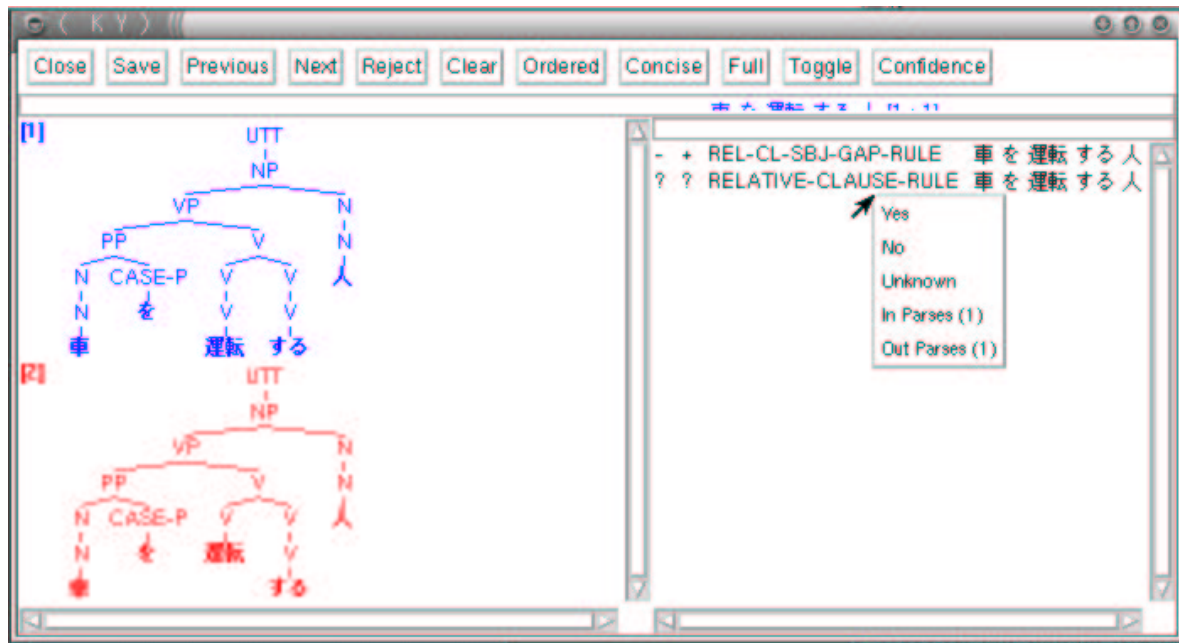


図 2: 解析結果の選択

果候補間で差異がある文法規則を提示される (図 2)。この構文木は、解析木より情報が簡略化されている。作業者は、[incr tsdb()] が提示した文法規則の差異に対する正誤の判定を、正しい解析結果が一つ選択されるまで繰り返す。図 2 は、解析結果を選択する例である。ウィンドウの左側に構文木の候補、右側に二つの解析結果で異なっている文法規則が提示されている。作業者が、右側の文法規則リストから適切なものを選択 (Yes) したり、不適切なものを排除 (No) することによって、左側の構文木の候補が自動的に絞り込まれていく。解析木が一つに選択されると MRS による意味表現が自動的に生成される。図 4 は図 3 に対する MRS である。ただし、簡単のため数量詞など詳細な情報は省略している。

各文で必要となる判定回数は、解析木候補数 N に対して概ね $\log_2 N$ である。つまり、5000 候補が出力された文であっても、約 12 回の判定で正しい解析木を絞り込むことができる。作業に慣れた段階では、平均 10 語の長さの文を対象にしたタグ付けを、1 時間当たり 50 文程度行うことが可能であった。黒橋ら [13] は、辞書語義文より長い文を対象にした構文解析タグを付けを、1 時間あたり 40 文で行ったと報告している。しかし、付与した

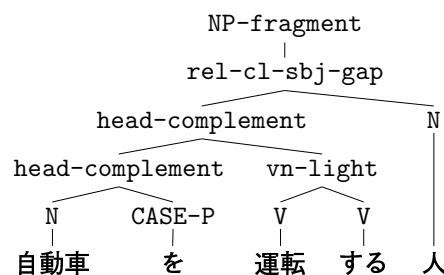


図 3: ドライバー₂ の解析結果 (解析木)

$$\langle h, x_1 \{ h : \text{prpstn_rel}(h_1) \\ h_1 : \text{hito}(x_1) \\ h_2 : \text{jidosha}(x_2) \\ h_3 : \text{unten}(u_1, x_1, x_2) \} \rangle$$

図 4: ドライバー₂ の解析結果 (意味表現)

情報は品詞タグと係り受けの関係であり、檜より簡略な情報である。

ツリーバンク構築の初期段階では、構文木候補に正しいものが存在しない場合や、冗長な曖昧性を持つ構文木が存在する 경우가少なくない。そこで、一度にタグ付けを進めるのではなく、最初は

解析結果を分析して、4章で述べるような文法拡張を並行して行った。特に、同じ長さの文を対象とすることにより類似した現象をまとめて分析することができ、効率的に文法を拡張することができた。

文法を変更したときに、作業者が解析木の選択を最初からやり直さずに済むように、[incr tsdb()]は、以前の作業者の判定をもとにツリーバンクを自動的に更新することができる[9]。解析木は文法に依存しているが、文法の変更によって作業者が再選択しなければならないのは、解析木の曖昧性が増大して新たな判断が必要なときか、既存のルール/語彙項目が大幅に変更されてシステムが自動的に解析木を再構築できなくなったときである。

また、文法、解析結果や作業者の判定など全ての情報はプレインテキストとして保存されている。そのため、文法中の表記やデータベースをテキスト処理のスク립トを使って容易に変更できる。文法記述とツリーバンク中のラベルを簡単なスク립トで変更しても、新しいツリーバンクは問題なく構築される。

このような「Redwoods型」のツリーバンク構築で問題となる点は、構築されたツリーバンクがある特定の文法に密接に結び付いていることである。しかし、文法を変更した際にも、[incr tsdb()]でツリーバンクを自動的に更新することができるため、大抵の場合この問題は軽減できると考えている。別の問題点は、タグ付けできる文が解析器によって解析できたものに限られるということである。文法で実装されていない現象を含む文や、システムの制約で解析に失敗した文はタグ付けできない。つまり、ツリーバンクの構築にとって、文法が高い網羅性を持つことが重要な条件となる。次章では、ツリーバンクを構築することを目的として、文法を拡張した過程と結果について述べる。

4 ツリーバンク構築のための文法拡張

4.1 辞書語義文ドメインへの拡張

JaCYは、もともと対話コーパスを対象としていたため、辞書語義文をドメインとした解析を行うためには文法を拡張する必要があった。これまでのJaCYの開発では、スピーチデータから銀行取り引きの分野への適用の際にカバー率を3カ月で48%向上させ、電子メール応答の分野への適用

の際に、2週間で51%向上させるなど、短期間で新しい分野に適応させることに成功している[14]。

一方、詳細な情報を持つ文法は一般的にその記述自身も複雑であるため、文法を短期間で拡張できるのは、内容を熟知したごく少数の開発者に限られる場合が多い。これは文法開発固有の問題ではなく、ソフトウェア管理一般の問題として知られている。

我々は、JaCYの拡張可能性を調べるために、Lexeedの語義文を対象として4週間で解析カバー率80%以上を達成することを試みた。このタスクには、オリジナルのJaCY開発に関わっていない6人が参加し、そのうち半数はHPSGを使った経験がほとんどない。

4.2 文法拡張の実際

PETで、初期状態のJaCYと未知語処理機能を使用して、Lexeedの全語義文81,000を解析した結果カバー率は39.3%であった。これに対して、解析に失敗していた数種類の表記を加えるだけで46.2%に向上した。

短期間で文法拡張の効果の傾向を捉えるため、単語親密度が6以上の語の各語義について、第一語義文のみをツリーバンク構築の対象とした。第一語義文では語の基本的な意味が記述され、第二文以降で補足的な説明がなされる場合が多い。対象となったのは約10,000文であり、1文の平均の長さは10.1語である。Lexeedの全ての語義に完全な統語情報を付与するため、この段階で未知語処理機能を使用しないようにした。その結果、カバー率の初期値は約10%程度になった。

実際の文法の拡張は以下のように行った。4週間での文法のカバー率と曖昧性の推移を図5と図6に示す*。

語彙の追加 カバー率の初期値が低いのは、JaCYの辞書に我々の定義した基本語の多くが含まれていなかったことによる。Lexeedから基本語の語彙を自動的に追加したなどにより、語彙数を31,000語まで到達させた。その結果、カバー率が55%まで大幅に向上した(図5の2点目)。

文法規則の追加・修正 その後、少量のツリーバンク構築を行いながら、その分析をもとに語彙と

*拡張したJaCYはwww.delph-in.netからダウンロードできる。

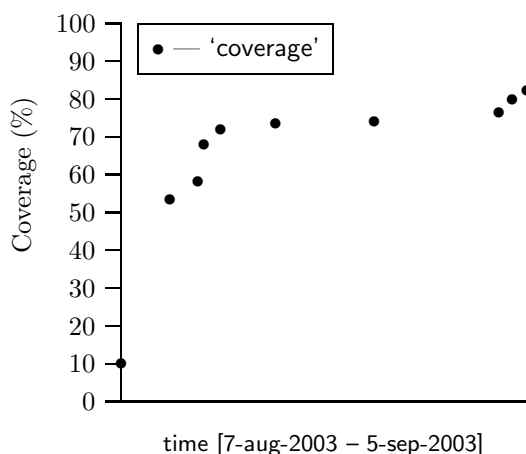


図 5: カバー率の推移

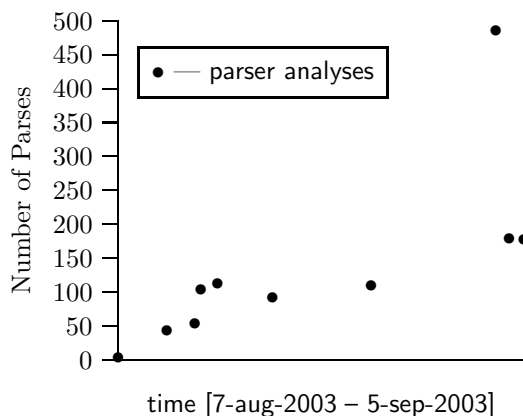


図 6: 曖昧性の推移

文法の追加・修正を行った。既存の文法規則の条件を緩和させる修正が主であったが、4.3章で述べるように複合動詞の解析規則など幾つかの文法規則を追加している[†]。この段階では、一般的な日本語文法の規則のみを追加している。これによりカバー率は70%以上に達した。

辞書に特有な表現の文法規則の追加 この時点まで、辞書語義文に特化した文法規則は追加していない。2週間ツリーバンク構築を続け、一般的な規則の追加・修正でのカバー率の向上が一段落した後に、辞書語義文に特有な規則を追加した。詳

[†]文法規則作成にあたっては JaCY の開発者から有益な助言をいただいた

細については 4.3 章で述べる。

文法体系の調整 文法規則を追加・修正しながら、文法体系全体の調整を行った。ツリーバンク構築の観点では、解析の曖昧性を低く抑えることが重要である。文法が不必要に曖昧性を含んでいると、解析候補数が増え、最適な解析の選択に手間がかかる。解析の網羅性を上げようとするれば、本質的な曖昧性が増加するのは避けられないが、同時に不必要な曖昧性が生じるのを防がなければならない。辞書に特徴的な表現 (4.3 章の (5)) に対応するため、構文木のルートの名詞句に副詞句の修飾を認めようとしたところ、副詞句があらゆる名詞句を修飾することを許してしまい、一文あたりの解析候補数が一気に約 500 にまで増大した (図 6 の最大値)。その後適切に規則の修正をした結果、一文あたりの解析候補数は 180 までに抑えられた。

以上の過程を経て、JaCY に習熟していない 6 人の作業者のみで、最終的に 4 週間でカバー率を 80%以上に向上させることに成功した。文法を全く異なるドメインに適用してカバー率を大幅に向上させたにも関わらず、文法規則の数は、文法拡張前の 114 から 122 に 5%弱増えただけである。さらに、文法体系の調整によって解析の曖昧性の増加もツリーバンク構築可能な範囲に抑えられている。これらは、JaCY のドメイン適用性の高さを示すとともに、日本語の一般的性質を適切に捉え、精細な文法拡張を行った結果と考えられる。

現在、我々は単語親密度 6 以上の単語に付けられた語義文 10,000 のうち 5,000 文のツリーバンク構築が終わっている。語義文の 636 文をサンプルとして調べたところ、94%が解析候補に正しいものを含んでおり、完全な解析木を選択することができた。約 5%は、解析器が正しい解析結果を出していない。これは主に、限定詞を含んだ文や並列構造を含んだ文の意味解析が誤っていたことが原因である。他に、作業者が正しい解析木を一つに絞り込めなかったものも 1%弱あった。

4.3 解析エラーの傾向と対策

ツリーバンク構築の際に見られた解析エラーの原因は、前述のように主に語彙や文法規則の不足によるものであるが[‡]追加した文法規則は次のよう

[‡]この他にも長文で解析可能性が膨大になりメモリ不足となる問題が生じた。現時点では極端に長い文を除外することで回

に大別できる。

1. 日本語文法の一般的な規則
2. 辞書語義文特有の構文規則

1はドメインに依存しない規則であり、日本語の性質を適切に捉えている必要がある。一方、2はドメイン依存の規則として形式化することも考えられるが、独立した日本語の表現として解釈できる文であれば、他の一般的な構文と並行する形で解析できるようにすることが望ましい。本研究では文法構築の見通しを良くするために両者を区別しつつも、2が一般的な文法規則の体系に沿うことを意識して規則を作成した。

以下では、追加した文法規則の例を挙げる。

日本語文法の一般的な規則：複合動詞規則

JaCYには複合動詞に関わる規則は存在せず、一部の複合動詞を辞書に直接記載しているのみである。しかし、日本語の複合動詞には生産的なものが存在するので、そのための規則を2つ追加した。

統語的複合動詞の例として(1)が挙げられる。

- (1) V₁/V₂: 書き/続ける、書き/終わる、書き/かける、書き/損ねる

統語的複合動詞は生産的かつ意味的にも統語的にも規則的である。よって全てを辞書に列挙するのは不適切であり、この種の複合動詞を生成する規則の追加が重要となる。

統語的複合動詞のV₂は、V₁を主辞としたVPを下位範疇化し、意味的にもV₁を主辞とした事象を取る。例えば、「健が日記を書き終える」の意味構造は(2)に相当するものになる。このような解析が可能となるように文法を拡張した。

- (2) a. $\langle oeru(kaku(ken, nikki)) \rangle$

語彙的複合動詞の例として(3)が挙げられる。

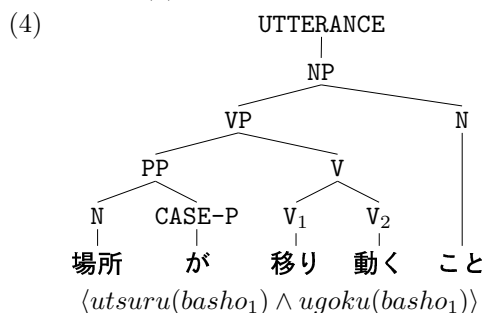
- (3) V₁/V₂: 立ち/並ぶ、打ち/殺す、指し/示す、繰り/返す

語彙的複合動詞は、統語的なものに比べると、意味的にも統語的にも不規則である。しかし、項構造が「似ている」もの同士が複合しやすい、という傾向がある。この大まかな傾向をとらえるために、結合価が同じ動詞2つを複合する規則を追加した[§]。意味的には2つの動詞の意味を単に結合

避した。文法を精練して解析曖昧性を減少させることにより、長文の解析カバー率も上がると期待される。

[§]現象を正確に扱うためにはさらに詳しい分析が必要であるが、ツリーバンク構築を進めるための暫定的対策として規則を追加した。

したものとなっている。例えば「移動₁」の語義文の構文木は(4)のようになる。



一部の語彙的複合動詞には、意味的、統語的に全く不規則なものがある。これらについては、一語として辞書に登録するべきであると考えている。

辞書ドメイン特有の問題

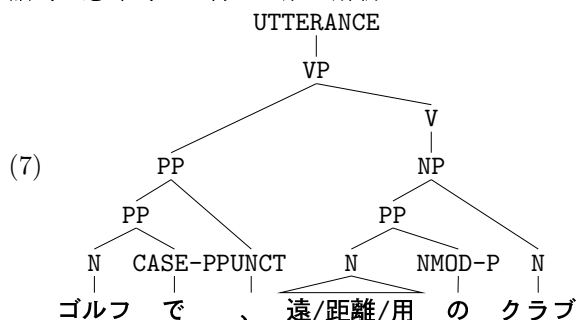
辞書では語義文に特徴的な文が頻繁に現れる。文(5)は「ドライバー」の語義文の1つである。

- (5) ゴルフで、遠距離用のクラブ

(5)は、『『ドライバー』はゴルフの用語では『クラブ』の一種を意味する』という意味で用いられている。このような文は辞書語義文特有の表現であると言える。しかし一方で、(6)のような、日本語としてごく自然なコピュラ文に対応するものと考えることができる。

- (6) ゴルフで、遠距離用のクラブだ。

従って(5)のような文を、辞書語義文特有の例外的な現象として扱うよりも、(6)のような文と統語的、意味的に並行する形で解析するべきである。



そこで、(5)のような文を空コピュラ規則により解析した。これにより、(5)を(6)のコピュラ文と並行的に解析できるようになった。(5)の統語構造は(7)のようになる。(7)の「遠距離用のクラブ」は空コピュラ規則の適用を受けて名詞から動詞に変わる。これが(6)における[V [NP 遠/距離/用/の/クラブ] [V だ]]に相当する。

5 言語モデルの構築

大規模なツリーバンクが完成すれば、ツリーバンク中の詳細な統語-意味情報から、最大エントロピー法で解析木選択の言語モデルを構築することができる [3]。予備的に簡単な統計モデルを構築し、[incr tsdb()] で再度解析を行い精度を調べた。檜の2,000文を訓練データとして用い、別の1,000文をテストデータとして解析を行った結果、72%の文で正しい解析結果が選択できた (ランダムに選択した場合は19%である)。訓練データを1,000文に減らした場合は、正解率は58%に留まった。ツリーバンク構築を進めた後、より詳細かつ規模の大きなデータで評価を行う予定である。

6 まとめと今後の方向性

システムを実用的なものにする目的においても、また言語 (資源) 全体に関する見通しのよい文法を記述する上でも、大規模なコーパスを処理することは、文法開発における重要な課題である。特にコーパスから知識獲得をおこなう場合は、統語処理と意味処理の両面から一定水準の解析精度が要求される。こうした問題に対し、文法開発ツールを用いて効率的な文法拡張とツリーバンク構築を並行して効率的に行うことで対処した。

現時点までで文法拡張と並行して約5,000文のツリーバンクを構築しているが、今後はLexeedの基本語の全語義文約80,000文のツリーバンクを完成させたい。言語モデルにより解析器の精度も向上するので、初期段階よりもツリーバンク構築の速度が速くなると考えている。また、構築手法の汎用性を検証するため、辞書語義文以外を対象にしてツリーバンクを構築を行いたい。

参考文献

- [1] 笠原要, 佐藤浩史, Francis Bond, 田中貴秋, 藤田早苗, 金杉友子, 天野昭成. 「基本語意味データベース:lexeed」の構築. In *2003-NLC-159*, 1/13-1/14 2004.
- [2] Francis Bond, 藤田早苗, 橋本力, 笠原要, 成山重子, Eric Nichols, 大谷朗, 田中貴秋, 天野成昭. 日本語ツリーバンク「檜」: 言語理解のためのコーパス. In *2003-NLC-159*, 1/13-1/14 2004.
- [3] Kristina Toutanova, Christopher D. Manning, and Stephan Oepen. Parse ranking for a rich HPSG grammar. In *Proceedings of The First Workshop on Treebanks and Linguistic Theories (TLT2002)*, Sozopol, Bulgaria, 2002.
- [4] Chiori Hori, Takaaki Hori, Hideki Isozaki, Eisaku Maeda, Shigeru Katagiri, and Sadaoki Furui. Deriving disambiguous queries in a spoken interactive ODQA system. In *ICASSP-2003*, pp. 624-627, 2003.
- [5] Mark Stevenson. *Word Sense Disambiguation*. CSLI Publications, 2003.
- [6] Dan Klein and Christopher D. Manning. Accurate unlexicalized parsing. In Erhard Hinrichs and Dan Roth, editors, *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pp. 423-430, 2003.
- [7] 天野成昭, 近藤公久. 日本語の語彙特性. 三省堂, 東京, 1999.
- [8] 金杉友子, 笠原要, 稲子希望, 天野昭成. 単語親密度に基づく基本的語彙の選定策. 第 NLC2002 巻, pp. 21-26, 2002.
- [9] Stephan Oepen, Dan Flickinger, Kristina Toutanova, and Christopher D. Manning. LinGO redwoods: A rich and dynamic treebank for HPSG. In *Proceedings of The First Workshop on Treebanks and Linguistic Theories (TLT2002)*, Sozopol, Bulgaria, 2002.
- [10] Carl Pollard and Ivan A. Sag. *Head Driven Phrase Structure Grammar*. University of Chicago Press, Chicago, 1994.
- [11] Ivan A. Sag and Tom Wasow. *Syntactic Theory: A Formal Introduction*. CSLI Publications, Stanford, 1999.
- [12] Ann Copestake, Dan Flickinger, Carl Pollard, and Ivan A. Sag. Minimal recursion semantics: An introduction. (manuscript <http://www-csli.stanford.edu/~aac/papers/newmrs.ps>), 1999.
- [13] Sadao Kurohashi and Makoto Nagao. Building a Japanese parsed corpus — while improving the parsing system. In Anne Abeillé, editor, *Treebanks: Building and Using Parsed Corpora*, chapter 14, pp. 249-260. Kluwer Academic Publishers, 2003.
- [14] Melanie Siegel and Emily M. Bender. Efficient deep processing of Japanese. In *Proceedings of the 3rd Workshop on Asian Language Resources and International Standardization at the 19th International Conference on Computational Linguistics*, Taipei, 2002.
- [15] Ann Copestake. *Implementing Typed Feature Structure Grammars*. CSLI Publications, 2002.
- [16] Ulrich Callmeier. PET - a platform for experimentation with efficient HPSG processing techniques. *Natural Language Engineering*, Vol. 6, No. 1, pp. 99-108, 2000.
- [17] Stephan Oepen and John Carroll. Performance profiling for grammar engineering. *Natural Language Engineering*, Vol. 6, No. 1, pp. 81-97, 2000.