HPSG を用いた統語解析のための統計モデル

光石 豊　鳥澤 健太郎　辻井 潤一
東京大学大学院 理学系研究科 情報科学専攻
{mitsuisi,torisawa,tsujii}@is.s.u-tokyo.ac.jp

1 はじめに

本論文では, (1) 現在開発中の HPSG に基づく日本語文法と, (2) この文法を基にした統計モデルを用いた統語構造の曖昧性解消について述べる. 我々の最終目標は, 主辞駆動句構造文法 (HPSG) [7] の枠組を用いて, 情報検索システム・対話システムなどの現実的な応用のための自然言語処理システムを構築することであるが, 本論文はその第一段階である.

HPSG は, 素性構造 [1] をデータ構造に用いた, 制約に基づく語彙化された文法枠組であり, 言語学者の間で近年注目されている. HPSG には計算言語学の観点からも注目すべき特徴がある. 主な特徴として以下の二つが考えられる.

1. 部分的な文法でも解析が可能である (アンダースペシフィケーション)
2. 数学的に厳密な枠組である

部分的な文法でも解析が可能であるという第一の性質により, 柔軟な文法の設計を行うことができる. 一般に, 語彙化された枠組では, 各語に関する言語現象について詳細な文法記述ができる反面, 全ての語に対する文法記述をする必要があり, 実用的な文法を設計するには設計コストが膨大である. HPSG では, 部分的な文法でも解析が可能なため, 人手で記述可能な大雑把な文法から出発することが可能である. 例えば, 最初に各品詞に対する記述をし, それから例外的な語・現象に対する詳細な記述を追加するということも容易である.

数学的に厳密な枠組であるという第二の性質により, 理論的に妥当な応用が可能である. 例えば, 機械学習などを数学的に正しく実現することが可能であろう.

従来, HPSG は, 素性構造の単一化のコストが高いため, 速度の面から現実的な応用には向かないとされていた. しかし, 我々の研究室で開発された, 素性構造処理のための高速なプログラミング言語 LiLFeS[5] を用いることにより, 現実的な時間で解析を行うことが可能になった.

我々は, HPSG の枠組を用いた実用的な日本語文法を作成した (第 2 節). 統語構造の曖昧性解消の精度を高めるため, 元の文法に, 「経験に基づく文法記述」と呼ぶ記述を追加した (第 3 節). この文法を基に, 簡単な統計モデルを用いて統語構造の曖昧性解消を行った (第 4 節).

2 HPSG の枠組による日本語文法

我々は, HPSG の枠組を用いた日本語文法を開発中である. 目標とする文法は, カヴァレッジの広さ, 解析時間の短さ, 発展性の大きさの三つの観点で実用的な文法である. 広いカヴァレッジと短い解析時間を得るには, できるだけ単純な文法の方が良いが, 単純化しすぎた文法では, 文法を改良・詳細化する手段がなく, 発展性がない. そこで, 将来的に文法の改良・詳細化が容易なように, 我々は各品詞・各機能語レヴェルに対する記述をした文法を設計した.

現在, 我々の日本語文法は,

- 6 個のルールスキーマ[1]
- 104 個の語彙項目テンプレート

から成っている.

2.1 ルールスキーマ

HPSG におけるルールスキーマは, 文脈自由文法 (CFG) の書き換え規則に相当するが, HPSG は語彙化された枠組であるため, 前者は後者と比べて抽象化されている. 例えば, ルールスキーマの 1 つである head-modifier schema は, 「非主辞側の子が主辞[2]側の子を修飾する」「主辞側の子はマークされていない」などの, 具体的な品詞・語に依存しない制約から成っている.

2.2 語彙項目テンプレート

HPSG では文法の大半を各語に対する語彙項目として記述する. しかし, 全ての語に対する語彙項目を詳細に記述するのは現実的には不可能である. そこで我々は, 各語に対してではなく, 何らかの語の集合に対して文法記述を行う. この記述を語彙項目テンプレートと呼ぶ.

ここで, 語の集合を定める手段として, JUMAN[4] の形態素情報を用いる. 基本的には, 各品詞に対して一つまたは複数の語彙項目テンプレートを用意する. 語彙項目テンプレートにはその品詞に共通の緩い制約のみを記述する. 例えば, 図 1 は, 副詞に対する語彙項目テンプレートの例[3]であるが, 大雑把に言うと, これには, 品詞が副詞で (`MAJ`), 助詞によってマークされる可能性があり (`NOMINAL`), 動詞や形容詞を副詞を修飾する可能性がある (`MODIFY`) という, 副詞共通の制約が書かれている. また, 動詞[4]の語彙項目テンプレート中の格フレームについては, ガ格が最大一個, ヲ格が最大一個, その他の格については制限なし, という単純な格フレームのみを用意している.

品詞によっては, 品詞に対する記述のみでは不十分な場合もある. その場合は, 各品詞に対してではなく, 各品詞細分類や各語に対して語彙項目テンプレートを用意する. 例えば, 名詞の場合は, 副詞的に使われる「副詞的名詞・時相名詞」, 「する」と結合して動詞になる「サ変名詞」, 補足節を作る「の」, 補足節を作るか普通名詞的に用いられる「こと」, のそれぞれに対して語彙項目テンプレートを用意している. また, 「が」「を」

[1] ここでいうルールスキーマとは, ID スキーマとプリンシプルを単一化したものを指している.

[2] 我々の日本語文法では, 全てのルールスキーマにおいて右側の子が主辞となっている.

[3] 簡単のため, 実際の語彙項目テンプレートと比べて単純化してある.

[4] サ変動詞は除く.

$$
\left[\begin{array}{ll}
\mathrm{SYN|LOC} & \left[\begin{array}{ll}
\mathrm{HEAD} & \left[\begin{array}{ll}\mathrm{MAJ} & \text{副詞} \\ \mathrm{NOMINAL} & + \end{array}\right] \\
\mathrm{SPECIAL} & [\mathrm{LEX}\ +] \\
\mathrm{ADJACENT} & \textit{non-sign} \\
\mathrm{ADJACENT\text{-}1} & \textit{non-sign} \\
\mathrm{MARK} & \textit{non-sign} \\
\mathrm{SUBCAT} & \langle\rangle \\
\mathrm{MODIFY} & [\mathrm{SYN|LOC|HEAD|MAJ}\ \text{述語性 or 副}]
\end{array}\right]
\end{array}\right]
$$

図 1: 副詞に対する語彙項目テンプレート

「は」「の」「と」などの主要な助詞については，助詞ごとに大きく異なる振舞いをするので，それぞれの助詞に対して語彙項目テンプレートを用意している．

格フレームを変化させる語については，各語に対して語彙項目テンプレートを用意している． 例えば，受動を表す接尾辞「れる」「られる」は，動詞と結合して，格フレーム中のガ格をニ格に，ヲ格をガ格に変化させるので，「れる」「られる」に対する語彙項目テンプレートを用意して，それに格フレームを変化させる記述をしている．そのほか，使役を表す接尾辞「せる」「させる」，サ変動詞「する」「できる」「させる」に対しても語彙項目テンプレートを用意して同様の記述をしている．

3 経験に基づく文法記述

前節で述べた文法に，常に成り立つとは限らないが，冗長な曖昧性を減らすのに貢献する記述を追加する． 将来的には，スコアづけをするような制約として記述してその制約を統計モデルで考慮する予定であるが，現時点では簡単のため決定的な制約として記述している．

一般に，日本語文の係り受け構造を考えたとき，各文節は距離の近い文節に係りやすいという性質があり，後述する統計モデルにもこの性質が組み込まれている． しかし，現実のコーパスにおいて，この性質に高い割合で背く典型的な言語現象がいくつかある．例えば，「は」でマークされた助詞句は，近い文節に係らず (図 2 左)，文末の文節に係る (図 2 右) という強い傾向がある (ただし，後方に他の「は」でマークされた助詞句がある場合，その直前の文節に係る傾向がある (図 3))．こういった言語現象を扱うために，文法に記述を追加する．この記述を「経験に基づく文法記述」と呼ぶ． 「経験に基づく文法記述」は，冗長な構文木を出さないための制約を記述したものである． 例えば，「は」でマークされた助詞句の例では，「太郎は遊ぶ花子を見る」という文に対して，図 2 左の構文木を返さず，図 2 右の構文木のみを返すような制約を記述する． 「経験に基づく文法記述」で表される制約は，常に成り立つとは限らない言語現象についての制約であるので文法のカヴァレッジを狭くする危険性があるが，それ以上に曖昧性解消の精度を高くする効果が得られると期待される． 本論文では「は」の傾向についての他に，「関係節に読点は含まれない」「読点を伴う時を表す名詞・名詞性名詞助数辞は常に副詞的に使われる」という「経験に基づく文法記述」を文法に追加した．

4 統計モデル

HPSG の枠組を用いた日本語文法を基にした，日本語の統語構造の曖昧性を解消するための統計モデルについて議論する．統語構造の曖昧性は，入力形態素列 W が与えられた時に出力される構文木 T のうち，確率最大の構文木 T_r を選ぶことによって解消されるとする．即ち，

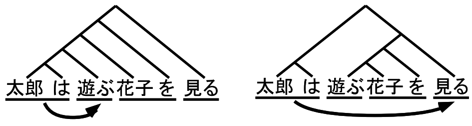

図 2: 「太郎は遊ぶ花子を見る」に対する誤った構文木 (左) と正しい構文木 (右)

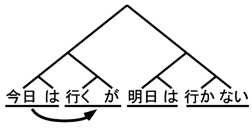

図 3: 「今日は行くが明日は行かない」に対する正しい構文木

$$
\begin{aligned}
T_r &= \mathrm{argmax}_T P(T|W) \\
&= \mathrm{argmax}_T \frac{P(W|T)P(T)}{P(W)} && \text{(ベイズの定理)} \\
&= \mathrm{argmax}_T \frac{P(T)}{P(W)} && (P(W|T)=1) \\
&= \mathrm{argmax}_T P(T) && (P(W) \text{ は } T \text{ に依存せず})
\end{aligned}
$$

$P(T)$ が問題となる統計モデルである． 本来ならば，この統計モデルにおいて構文木に含まれる全ての素性を考慮すべきであるが，本研究では以下のように単純化した統計モデルを用いる． まず，構文木の根ノードに対する素性構造を F_{root} とする．また，構文木に含まれる高さ一段の部分木を $\langle R, F_M, F_N, F_H\rangle$ で表す．ただし，R はその部分木で使われているルールスキーマの名前を表し，F_M，F_N，F_H はそれぞれ親ノード，非主辞 (左) 側の子ノード，主辞 (右) 側の子ノードに対する素性構造を表す[5]． 本論文で我々が採用した統計モデル $P(T)$ は次のようなモデルである．

$$
\begin{aligned}
P(T) = {} & P(c(F_{root}), p(F_{root}), w(F_{root})) \\
& \times \prod_{\langle R, F_M, F_N, F_H\rangle} f(R, F_M, F_N, F_H) \qquad (1)
\end{aligned}
$$

ただし，

$$
\begin{aligned}
& f(R, F_M, F_N, F_H) = \\
& \quad P(R, p(F_N), w(F_N) \mid p(F_M), w(F_M)) \\
& \quad \times P(c(F_N), c(F_H), b(F_H) \mid R, p(F_N), p(F_M), c(F_M)) \\
& \hfill (2)
\end{aligned}
$$

$p(F)$，$w(F)$，$c(F)$，$b(F)$ は，構文木のノードの素性構造 F を引数とする関数である[6]．$p(F)$ と $w(F)$ は，それぞれ主辞素性中の品詞素性と単語素性の値[7]を返す．$c(F)$ は，

[5] R, F, M, N, H はそれぞれ Rule schema, Feature structure, Mother, Non-head daughter, Head daughter の頭文字である．

[6] p, w, c, b はそれぞれ part of speech, word, comma, bunsetsu の頭文字である．

[7] 以下，それぞれ単に「品詞」「単語」と呼ぶ．

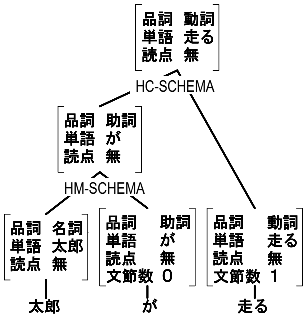

図 4: 「太郎が走る」に対する構文木 T_{ex} (分かりやすさのため, 素性名や値は実際とは変えてある)

そのノードの支配する句に読点が含まれているかどうかを表す素性の値 (有 / 無) を返す. $b(F)$ はそのノードが支配する句に含まれる文節数を返す. $\prod$ は, 構文木中の全ての部分木に対する f の値を掛け合わせることを表す. 例えば, 図 4 の構文木 T_{ex} に対する $P(T_{ex})$ は以下のように求められる.

$$\begin{aligned} P(T_{ex}) = {} & P(\text{無}, \text{動詞}, \text{走る}) \\ & \times P(\texttt{HC}, \text{助詞}, \text{が} \mid \text{動詞}, \text{走る}) \\ & \times P(\text{無}, \text{無}, 1 \mid \texttt{HC}, \text{助詞}, \text{動詞}, \text{無}) \\ & \times P(\texttt{HM}, \text{名詞}, \text{太郎} \mid \text{助詞}, \text{が}) \\ & \times P(\text{無}, \text{無}, 0 \mid \texttt{HC}, \text{名詞}, \text{助詞}, \text{無}) \end{aligned}$$

式 (1), (2) で示される統計モデルでは以下のような前提をおいている.

- 構文木はトップダウンに生成され, 構文木中の高さ一段の部分木はそれぞれ独立に構成される (式 (1)).
- ノードの素性構造は, 品詞, 単語, 読点の有無, 文節数に関する素性のみで表される (式 (2)).
- 各部分木において, 適用されるルールスキーマ, 非主辞側の子の品詞・単語は, 親の品詞・単語のみに依存する (式 (2) 右辺第 1 行目).
- 各部分木において, 子が支配する句に含まれる読点の有無 (主辞側と非主辞側の両方) と, 主辞側の子の支配する句に含まれる文節数は, 適用されるルールスキーマ, 非主辞側の子の品詞, 親の品詞・親の支配する句に含まれる読点の有無に依存する (式 (2) 右辺第 2 行目).

また, 主辞側の子の支配する句に含まれる文節数を統計モデルに取り入れることにより, 係り受けの距離の概念を確率値に反映させている.

現在の我々の日本語文法を用いて解析した場合, 同じ文に対する異なる構文木間で $c(F_{root})$, $p(F_{root})$, $w(F_{root})$ の値はそれぞれ常に等しくなっている. 従って, $P(c(F_{root}), p(F_{root}), w(F_{root}))$ の値は一定となり, 構文木の確率比較の際この値は無視できる. 各部分木の条件付き確率 (式 (2) の右辺第 1 行目と第 2 行目) は, 訓練コーパスから最尤推定することにより学習される. ただし, データスパースネスの問題があるため, 式 (2) の右辺第 1 行目の確率については, 次のようにバックオフ [3] を行う.

$$P(R, p(F_N), w(F_N) \mid p(F_M), w(F_M)) = \left\{ \begin{array}{l} \frac{C(R, p(F_N), w(F_N), p(F_M), w(F_M))}{C(p(F_M), w(F_M))} \\ \frac{C(R, p(F_N), p(F_M), w(F_M))}{C(p(F_M), w(F_M))} \times \frac{C(p(F_N), w(F_N))}{C(p(F_N))} \\ \frac{C(R, p(F_N), w(F_N), p(F_M))}{C(p(F_M))} \\ \frac{C(R, p(F_N), p(F_M))}{C(p(F_M))} \times \frac{C(p(F_N), w(F_N))}{C(p(F_N))} \\ \epsilon \ \text{(十分小さい値)} \end{array} \right\}$$
$$\text{のうち, 最初に求まる } 0 \text{ でない値}$$

ただし, $C(X)$ は, 訓練コーパス中に X が出現する頻度を表す. 式 (2) の右辺第 2 行目の確率については, 値が求まらない場合は, 十分小さい値 ϵ を与えた.

5 実験

実験は, EDR 電子化辞書 [2] の日本語コーパスを用いて行った. 統計モデルは, このコーパス中の 108928 文を訓練コーパスとして学習した. テストコーパスにはランダムに選んだ 1000 文を用いた[8].

5.1 文法のカヴァレッジと解析時間

(a) 第 2 節で設計した日本語文法と, (b) その文法に第 3 節の「経験に基づく文法記述」を追加した文法を, テストコーパス 1000 文に対するカヴァレッジと解析時間の二つの観点から評価した. カヴァレッジについては, 解析に成功する (最低一つ以上の構文木が出力される) 文の割合と, 出力される構文木の中に正解を含む文の割合とを用いて評価し, 解析時間については解析に成功する文の一文あたりの平均解析時間を用いて評価する. パーザには, プログラミング言語 LiLFeS で実装された CKY ベースのボトムアップパーザを用い, 実験は, DEC AlphaServer 4100 (400MHz, 4GB RAM) 上で行った.

文法	解析に成功する文の割合	解析結果中に正解を含む文の割合	一文あたりの平均解析時間
(a)	91.8%	72.4%	1.30 秒
(b)	87.5%	62.9%	1.12 秒

表 1: (a) 元の文法と (b) 「経験に基づく文法記述」を追加した文法の, カヴァレッジと解析時間

実験結果を表 1 に示す. 元の文法は, 比較的広いカヴァレッジを持ち, また解析も現実的な時間で実現されている. また, 「経験に基づく文法記述」を追加した文法は, 多少カヴァレッジを狭くしている.

5.2 統語解析の曖昧性解消の精度

統語解析の曖昧性解消の精度を, (a) の文法 + ベースラインモデル, (a) の文法 + 統計モデル, (b) の文法 + ベースラインモデル, (b) の文法 + 統計モデル, の 4 つに対して評価した. 評価基準の精度には, 文節単位の係り受けの精度を用いた. 即ち, 各文の最後の 2 文節を除く全ての文節のうち, 正しい係り先文節を推定された文節

[8] 訓練コーパスとテストコーパスとは互いに排他的である.

モデル	部分精度	全体精度
a) + ベースライン	74.0%	72.0%
a) + 統計	77.7%	75.1%
b) + ベースライン	78.8%	75.1%
b) + 統計	80.0%	76.1%

表 2: 曖昧性解消の精度の評価

の割合である．精度は部分精度と全体精度の二通り求めた．部分精度とは，解析に成功する文の集合における精度であり，全体精度とは，テストコーパスの全ての文の集合における精度である．全体精度を求める際，解析に失敗した文については，全ての文節が隣の文節に係るとした．現在の実装の都合上，解析に成功する文のうち 41 語以上の長い文 ((a), (b) の文法で各々 11 文と 8 文) については，解析に失敗する文として扱った．また，ベースラインモデルとは，各文節に対して，可能な最も近い係り先文節を選ぶアドホックなモデルのことを指し，統計モデルとは第 4 節で定義した統計モデルのことを指す．

実験結果を表 2 に示す．単純な統計モデルでもある程度高い精度が得られている．また，「経験に基づく文法記述」が精度向上に貢献している．「経験に基づく文法記述」を追加した文法を用いると，アドホックなベースラインモデルでも高い精度を実現している．

6 考察

カヴァレッジについては，文法を改良することによりある程度広くすることが可能である．しかし，任意の文に対して必ず解析結果を与える文法を設計しようとすると，低頻度な言語現象を扱うような文法記述を増やすと，他の高頻度な言語現象の解析に余計な曖昧性を増やすという悪影響を生む危険性がある．カヴァレッジを広くするには，単に文法記述を増やすのではなく，ロバストパージングの手法を導入する必要となろう．

解析時間については，現在の実装でも十分実用的な解析時間を実現している．カヴァレッジを高めるため文法を改良することにより速度低下を招く危険性があるが，我々の研究室ではさらに高速な解析をするための手法をいくつか提案しており [8, 6, 9, 10]，速度面での心配はいらないものと思われる．

精度については，単純な統計モデルでもある程度高い精度を得られることが分かった．これは，人手で記述した基本的な文法を用いたことによると考えている．また，単語のシソーラス上のクラスや深層格などの他の素性を考慮した統計モデルを用いることによりさらに高い精度を得ることを期待している．

現在の実装では，日本語の，各文節が距離の近い文節に係りやすいという性質に反する言語現象に対して，「経験に基づく文法記述」を単に文法に追加している．これは曖昧性解消の精度を向上させるものの，カヴァレッジの低下を招いている．これを避けるため，将来的には，「経験に基づく文法記述」を単に文法に追加するのではなく，統計モデルで扱うことを考えている．

7 おわりに

本論文では，現在開発中の HPSG の枠組に基づく日本語文法と，この文法を基にした統計モデルを用いた統語構造の曖昧性解消方法について述べた．我々の文法は，広いカヴァレッジと現実的な解析時間を実現している．基本的な文法を記述することにより，単純な統計モデルでもある程度高い曖昧性解消の精度が得られることが分かった．また，「経験に基づく文法記述」を文法に追加することによって，曖昧性解消の精度が向上した．今後は，文法・統計モデルの改善をする予定である．

参考文献

[1] Bob Carpenter. *The Logic of Typed Feature Structures*. Cambridge University Press, 1992. ISBN 0-521-41932-8.

[2] EDR (Japan Electronic Dictionary Research Institute, Ltd.). EDR electronic dictionary version 1.5 technical guide, 1996. Second edition is available via http://www.iijnet.or.jp/edr/E_TG.html.

[3] Slava M. Katz. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. ASSP-35, No. 3, pp. 400–401, 1987.

[4] Sadao Kurohashi and Makoto Nagao. Japanese morphological analysis system JUMAN version 3.4 manual, 1997. The system is available via ftp://pine.kuee.kyoto-u.ac.jp/pub/juman/juman3.4.tar.gz.

[5] Takaki Makino, Kentaro Torisawa, and Jun'ichi Tsujii. LiLFeS — practical unification-based programming system for typed feature structures. In *Proceedings of the Natural Language Processing Pacific Rim Symposium 1997 (NLPRS'97)*, pp. 239–244, 1997. http://www.is.s.u-tokyo.ac.jp/~mak/lilfes/lilfes.ps.gz.

[6] Kenji Nishida, Takaki Makino, Kentaro Torisawa, Yuka Tateisi, and Tsujii Jun'ichi. Extension of a feature structure abstract machine for partial unification. In *PACLING '97*, pp. 232–243, 1997.

[7] Carl Pollard and Ivan A. Sag. *Head-Driven Phrase Structure Grammar*. The University of Chicago Press, 1994. ISBN 0-226-67447-9.

[8] Kentaro Torisawa and Jun'ichi Tsujii. Computing phrasal-signs in HPSG prior to parsing. In *Proceedings of the 16th International Conference on Computational Linguistics (COLING-96)*, pp. 949–955, August 1996.

[9] 吉田稔，牧野貴樹，鳥澤健太郎，辻井潤一．素性構造処理言語 LiLFeS の最適化技術．言語処理学会第 4 回年次大会発表論文集．言語処理学会，1998.

[10] 宮尾祐介，鳥澤健太郎，建石由佳，辻井潤一．実用的な HPSG 文法のための二つの手法：型の combining と選言的素性構造の packing．言語処理学会第 4 回年次大会発表論文集．言語処理学会，1998.