

HPSG Supertagging の精度向上のための 単語クラスタリング

吉澤 智也^{†1} 松崎 拓也^{†1}
宮尾 祐介^{†1} 辻井 潤一^{†1,†2,†3}

HPSG 構文解析における supertagging の精度を向上させることを目的として、単語クラスタリングの結果を利用した Supertagging とは、一つの単語に割り当てられる語彙項目の候補を確率モデルを用いて削減する方法である。一方で、supertagging は構文解析の速度と精度を向上させるものの、間違っただ候補削減によって致命的な構文解析のエラーを引き起こしてしまうことがある。低頻度語に関するデータの少なさ（スパース性）は前述の致命的なエラーの原因となりうる。我々は大規模コーパスに対して Brown らの単語クラスタリングを実行し、得られるクラスタのビット列を supertagging への新たな素性として導入した。単語クラスタ情報は、単語のスパース性を軽減すると考えられる。Penn Treebank を用いた実験により精度を検証したが、精度向上はみられなかった。本稿ではその原因についての分析・考察を行う。

Word Clustering for HPSG Supertagging

TOMONARI YOSHIZAWA,^{†1} TAKUYA MATSUZAKI,^{†1}
YUSUKE MIYAO^{†1} and JUN'ICHI TSUJII^{†1,†2,†3}

This paper is aimed at increasing the accuracy of supertagging for a Head-Driven Phrase Structure Grammar (HPSG) parser, using class features induced from word clustering. Supertagging is a technique to eliminate possible candidates of lexical entries to a word by using a probabilistic model before parsing. Although supertagging improves the speed and accuracy of HPSG parsers, a wrong elimination of lexical entries can cause fatal errors in HPSG parsing. The sparseness of infrequent words is possibly related to these mistakes. Since previous supertaggers use simple N-grams of part-of-speech tags and words as features, examples of infrequent words are not sufficient to estimate a probabilistic model. We applied Brown's word clustering algorithm to BLIPP corpus and introduced new features with the result of clustering in a probabilistic model of a supertagger. In our experiments using the Penn Treebank, word clustering does not improve our supertagger. This paper presents our analysis

of the reason for this observation.

1. はじめに

構文解析は自然言語処理の基礎技術の一つであり、工学的・言語学的な視点から様々な枠組みが提唱されている。語彙化文法は言語の形式化の一つで、深い統語的な解析を行うことから、様々なアプリケーションへの応用が期待されている。

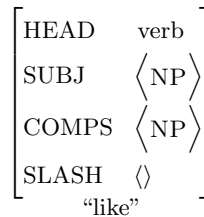
一方、語彙化文法による構文解析は、用いられるデータ構造の複雑性や、後述する語彙項目の多さにより、多大な計算機資源を必要とする。このような計算量の問題に対して、supertagging と呼ばれる手法が高速化に有効であることが示されている¹⁾²⁾³⁾。Supertagging とは、一つの単語に割り当てられる語彙項目の候補を確率モデルを用いて削減する方法である。文の持つ統語的構造は個々の単語の語彙項目によって強く制限されているため、supertagging を用いると構文解析の速度を飛躍的に向上させることができる。

しかし、supertagging は構文解析の速度と精度を向上させるものの、間違っただ候補削減によって致命的な構文解析のエラーを引き起こしてしまうことがある。したがって、supertagging の精度を上げることは、語彙化文法を用いた高速かつ高精度な構文解析にとって大変重要な課題である。

Koo ら⁴⁾ は、Brown ら⁵⁾ による単語 bi-gram を用いた単語クラスタリングの結果を係り受け解析に利用し、有意な精度向上が得られることを示した。また、Miller ら⁶⁾ は固有表現認識に用いる素性として、Brown らのクラスタリングの結果を利用し、有意なエラー率の減少を得ている。これらの結果より、構文解析における確率モデルの素性として、単語クラスタ情報を使用することの有効性が示唆される。

本研究ではこれらの既存研究に従い、Brown のアルゴリズムによって得られたクラスタリング結果が、二分木を表すバイナリビット列で与えられることを利用し、そのビット列を用いて supertagging の新たな素性テンプレートを作成した。また、我々は Penn Treebank

^{†1} 東京大学大学院情報理工学系研究科コンピュータ科学専攻
Department of Computer Science, University of Tokyo
^{†2} マンチェスター大学情報学研究所
School of Computer Science, University of Manchester
^{†3} 英国国立テキストマニングセンター
National Center for Text Mining



([NP.nom<V.bse>NP.acc]_lxm-no_singular3rd_verb_rule)

図 1 語彙項目の例

Fig.1 An example of lexical entry

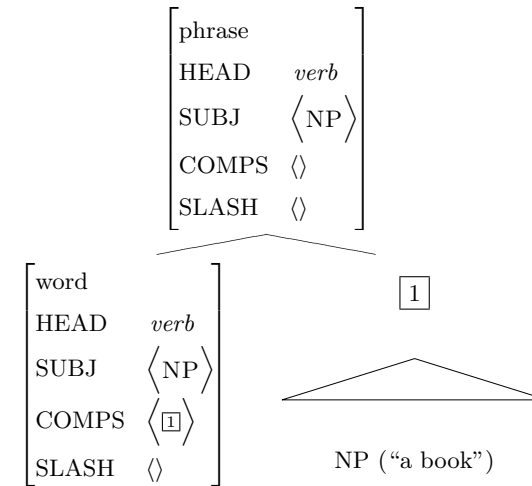


図 2 スキーマ適用の例

Fig.2 An example of schema application

を用いて実験を行い, supertagging の精度を評価した. 結果は, 期待されるような精度向上を得ることができなかった. 本稿では精度向上が得られなかった原因の分析・考察を行う.

2. 背景

2.1 HPSG

Head-driven Phrase Structure Grammar (HPSG)⁷⁾ は, Pollard と Sag によって提唱された語彙化文法の一つである. HPSG では, 単語それぞれに語彙項目 (Lexical Entry) が一つずつ割り当てられている. それらの語彙項目の要素は, スキーマと呼ばれる文法規則によって単一化され, 一つの構文木を作る.

語彙項目の例として, “I like the book.” の “like” に対応する語彙項目を図 1 に挙げる (括弧内は今回用いた文法での名称. [SUBJ < HEAD > COMPS] のように表記する). 語彙項目では, 素性構造 (feature structure) と呼ばれるデータ構造が用いられ, 単語に関する様々な情報は素性 (feature) として表される. 図 1 は単語が動詞であることが HEAD で表されており, 「この単語は NP を目的語として取る」ことが図中の COMPS と呼ばれる素性で, 「この単語は主語として NP が必要」なことが図中の SUBJ と呼ばれる素性でそれぞれ表されている.

また, 名詞句 (NP) “the book” を表す語彙項目と, 動詞 “like” の COMPS 素性 (「NP を目的語としてとる」) が単一化されることにより, 動詞句 “like the book” を作る (図 2). この文法規則は Head-Complement スキーマと呼ばれる.

HPSG をはじめとした語彙化文法は, 少数の文法規則と多数の語彙項目からなっており, 文法的説明を文法規則そのものではなく, もっぱら単語に対する語彙項目によって行って

いる.

2.2 Supertagging

HPSG をはじめとした語彙化文法を用いた構文解析では一般に, 単語に割り当てべき語彙項目の候補が多数存在している. 一つの文における語彙項目の組み合わせ数は膨大なものになり, 探索には大きな計算機資源を要する. さらに, 文法規則 (スキーマ) を適用することによって語彙項目から構文木を構築する際に語彙項目と素性の単一化を行う必要があり, CFG に比べより多くの処理を行わなければならない.

このような計算量の問題を解決するため, Bangalore と Joshi は¹⁾ は Supertagging と呼ばれる高速化手法を提案した. これは確率モデルを用いて, 割り当てべき語彙項目 (supertag) の候補を絞り込む手法である. Matsuzaki ら⁸⁾ は supertagging と, CFG-filtering と呼ばれる HPSG を CFG を用いて近似する方法を組み合わせ, HPSG 構文解析を 10 倍高速化することに成功している.

Supertagging は系列ラベリング問題に分類され, 機械学習を用いたアプローチがなされている. Bangalore と Joshi¹⁾ は 3-gram 隠れマルコフモデルを, Clark ら³⁾ は最大エントロピーモデルを用いて Supertagging を実装した.

本論文では Clark らに従い、最大エントロピーモデルを用いて Supertagger を実装する。文 s と supertag (ここでは語彙項目) l に対する条件付き確率は次のように表現される。

$$p(l|s) = \frac{1}{Z(s)} \exp\left(\sum_i \lambda_i f_i(l, s)\right) \quad (1)$$

ここで $f_i(l, s)$ は文 s と語彙項目 l についての素性であり、 $Z(s)$ は正規化定数である。 λ_i は素性の重みであり、 l と s の対からなる訓練データを用いて推定される。

今回我々は、確率モデルの推定に Broyden-Fletcher-Goldfarb-Shanno (BFGS) アルゴリズムを実装した Amis^{*1}を用いた。

2.3 単語クラスタリング

我々は Brown ら⁵⁾ の提唱した単語クラスタリングアルゴリズムを用いた。このアルゴリズムは、bi-gram モデルを用いて単語を階層的にクラスタリングするもので、全ての単語を葉として持つ二分木を作成する。二分木の任意のノードは、その下に属する葉(単語)全てからなるクラスタを意味している。

Brown らのアルゴリズムは、まず各々の単語自体がそれぞれ一つのクラスタに属している状態からスタートする。つまり、 V 個の単語が存在しているとき、 V 個のクラスタ c_i ($1 \leq i \leq V$) が存在しているとみなす。そして、ある二つのクラスタをまとめて木(クラスタ)を作ったとき、対象となる文における全クラスタ間の相互情報量

$$I(c_i, c_j) = \sum_{c_1, c_2} P(c_1 c_2) \log \frac{P(c_2|c_1)}{P(c_2)} \quad (2)$$

の減少が一番少なくなるような二つのクラスタを選び、木(クラスタ)をマージする。ここで、 $P(c_1 c_2)$ はクラスタ c_1, c_2 に属する単語が $c_1 c_2$ と連続して出現する確率を表す。この操作を繰り返して、最終的に全クラスタが一つの木となるようにする。

単語クラスタリングアルゴリズムの結果は、意味的なまとまりを含むことを Brown らは論文中で指摘している。たとえば、{"sell", "buy", "selling", "buying", "sold"} などが一つのクラスタとして挙げられている。クラスタリングにより得られる木は二分木であり、最終的に得られた単語クラスは、"010010001" などのようなバイナリ・ビット列で表現できる。このクラスタリング結果を、未知語・低頻度語に対するスムージングに利用する研究

がなされている。Miller ら⁶⁾ は固有表現認識に用いる素性として、Brown らのクラスタリングの結果を利用した。具体的には、クラスタリング結果から得られるビット列のうち、8, 12, 16, 20 ビットからなる接頭辞を素性として追加した結果、25%エラーを削減した。また、Koo ら⁴⁾ は係り受け解析に、Brown らの単語クラスタリングの結果を素性として使用し、英語とチェコ語の両方で有意な精度向上を得た。Koo らは Miller らとは異なり、以下の二つの素性を用いている。

- クラスタリング結果から得られるビット列のうち、4-6 ビットからなる接頭辞。これを POS の代わりに用いる。
- 全ビット列。ただし、クラスタ数が 1000 になるようにしている。

Koo らはこの手法により、英語とチェコ語の双方でそれぞれ 1 ポイント程度の有意な改善(英語で 92.02%から 93.16%, チェコ語で 86.13%から 87.13%)を得ている。

3. 提案手法

3.1 Supertagger

今回我々は最大エントロピーモデルを用いて supertagger を実装した。この理由として、最大エントロピーモデルは素性の追加が容易であり、単語クラスタリングの結果を簡単に組み合わせることができると考えられるからである。

表 1 に今回実装した supertagger で用いた素性を示す。 w_i および p_i は、ターゲットとなる単語から数えて、 i 番目の単語と POS をそれぞれ表す。例えば "The/DT teacher/NN provided/VBD a/DT solution/NN to/IN the/DT students/NNS" という文において、"provided/VBD" の supertag を推定する場合を考える。この場合、 w_0 が "provided", w_{-1} が "teacher", p_0 が 'VBD' などとなる。実際に使用される素性は表 2 のようになる。実装では、語彙項目の各候補の場合について、素性テンプレートを用いてその重みを足し合わせ、一番スコアが大きくなるような語彙項目を求める

3.2 クラスタリング

単語クラスタリングについて詳細を述べる。我々は単語そのものではなく「read/VBZ」「read/VBD」など「単語 + POS」の組をクラスタリングする。単語自体ではなく「単語 + POS」の組を用いた理由は、今回用いた HPSG 文法では語彙項目が(大まかに言って)POS の細分となるように作られているという事実に基づいている。例として、今回用いた文法における語彙項目を表 3 に挙げる。このような事実から、一般的に言えば、同じ単語であっても POS が異なる場合には、割り当てられるべき語彙項目の候補が大きく異なる可能性が

*1 <http://www-tsuji.is.s.u-tokyo.ac.jp/amis/>

表 2 素性の例
Table 2 An example of features

素性	意味
$w_0 = \text{"provided"}$ & "provided" = "[NP.nom<V.bse>NP.acc]_lxm-past_verb_rule"	「今見ている単語が provided で、 provided の語彙項目が語彙項目 [NP.nom<V.bse>NP.acc]_lxm-past_verb_rule である。
$w_{-1} = \text{"teacher"}$ & $w_0 = \text{"provided"}$ & "provided" = "[NP.nom<V.bse>NP.acc]_lxm-past_verb_rule"	「teacher provided と続き、 provided の語彙項目が語彙項目 [NP.nom<V.bse>NP.acc]_lxm-past_verb_rule である。

表 3 語彙項目の例
Table 3 Examples of lexical entries

対応する POS	語彙項目	例
動詞一人称単数 (VBP)	[NP.nom<V.bse>NP.acc]_lxm-no_singular3rd_verb_rule	"I love you."
動詞現在進行形 (VBG)	[NP.nom<V.bse>NP.acc]_lxm-prp_verb_rule	"I was reading a book."
形容詞 (JJ)	[<ADJP>]N_lxm	"last year"

表 1 Supertagger で用いられる素性
Table 1 Features used in the supertagger

詳細	素性テンプレート
周辺の単語	w_{-1}, w_0, w_1
周辺の POS	$p_{-2}, p_{-1}, p_0, p_1, p_2, p_3$
n-gram	$w_{-1}w_0, w_0w_1, p_{-1}w_0, p_0w_0, p_1w_0,$ $p_{-2}p_{-1}p_0p_1, p_{-1}p_0p_1p_2, p_0p_1p_2p_3,$ $p_{-2}p_{-1}p_0, p_{-1}p_0p_1, p_0p_1p_2, p_1p_2p_3,$ $p_{-2}p_{-1}, p_{-1}p_0, p_0p_1, p_1p_2, p_2p_3$

表 4 クラスタリング結果のビット列から作られる素性の例
Table 4 An example of features created from a cluster bit string

詳細	素性テンプレート
接頭辞 2 ビット	$b_{i2} = 10$
接頭辞 4 ビット	$b_{i4} = 1001$
接頭辞 8 ビット	$b_{i8} = 10010110$
全ビット列	$b_{i0} = 100101101010$

ある。そのため、単語の形だけでなくその POS の情報も入れる必要があることから、このような組を導入した。

3.3 単語クラスタリング結果の supertagger への導入

本節では前節の単語クラスタリングによって得られたバイナリビット列を、supertagger の確率モデルの新たな素性として追加する方法について述べる。

本実験では素性の追加方法として、次のような三通りの方法を提案する。

- (1) 全ビット列を素性として追加する
- (2) 全ビット列について、2, 4, 8 ビットからなる接頭辞を素性として追加する
- (3) 単語を全ビット列で置き換える（ただし、低頻度語に限る）

手法 (1) および (2) については、前節において列挙した素性テンプレートのうち、 w_i

の代わりにクラスタのビット列 b_i を使った素性を新たに追加する。例として $w_i = \text{"text"}$ の場合を考える。これに対応するビット列が "100101101010" であったとき、これを使って表 4 のような素性テンプレートを導入する。つまり、それぞれの単語に対し、対応するビット列を素性として追加するのである。

手法 (3) については、単語を補完する形ではなく、単語そのものをビット列で置き換えてしまう。例として "wight" (勇敢な) という低頻度出現単語を挙げる。これを、対応する "11011010" というビット列に置き換え、 $w_i = 11011010$ という素性に変換する。

4. 実 験

4.1 データセット

実験に使用した HPSG 文法には、Miyao ら⁹⁾ によって作成された Enju HPSG Grammar

を使用した。この文法は、主に Pollard と Sag ら⁷⁾ の形式化に従い、Penn Treebank のセクション 02-21 から辞書を自動獲得したものである。我々は supertagging の確率モデルの推定・評価に、セクション 02-21, セクション 22, セクション 23 を訓練用・開発用・評価用に用いた。各々のセクションに含まれる文数は、それぞれ 39,832 文 (852,918 語), 1,648 文 (37,147 語), 2,291 文 (50,844 語) である。

また、単語クラスタリングには BLIPP コーパスを用いた。これは Charniak¹⁰⁾ によって Penn Treebank と重なりが無いように Wall Street Journal から作られた構文木付きコーパスである*1。このうち、1987 年と 1988 年分を使用した。全 1,593,238 文 (38,627,057 語) からなる。また、コーパス中の異なり語数は 318,753 語 (後述するように数字を一つにまとめると 262,783 語) である。

4.2 Supertagger の実装の詳細

Supertagger の確率モデルの推定の際には、Ratnaparkhi¹¹⁾ に従い、観測事象における出現回数が三回未満の素性について足切りを行い、確率モデルから削除した。

4.3 単語クラスタリング

Miller ら⁶⁾ に従い、我々はクラスタ数として 500 および 1000 を設定し、BLIPP コーパスに対してクラスタリングを実行した*2。クラスタリングの前処理として、“1,200/CD” や “0.5/CD” など、単語が数字のみからなるトークンについては、一律に “-NUMBER-/CD” に置換した。これにより、単語の異なり語数が減り、クラスタリングを高速化することができる。

表 5 にクラスタリングによって得られたクラスタの例を示す。この例では、主として副詞がまとまってクラスタに属していることが見て取れる。

4.4 実験結果

Supertagging は、Penn Treebank の gold-standard の POS タグを入力として用いて精度を評価した。評価には評価用のセクション 23 の全ての文を使用した。提案手法およびベースラインの精度を表 4.4 に示す。ここで低頻度語とは、セクション 02-21 における出現回数が 3 回以下の単語を指す。

結果は、いずれの手法においても精度が低下した。全単語に対してクラスタリング結果のバイナリビット列を素性として追加した (1) と (2) において、特に低頻度語において正

*1 ただし、本研究で利用した情報は単語および POS のみであり、構文木部分は利用していない。

*2 Brown のクラスタリングの実行には brown-cluster を用いた。

<http://www.eecs.berkeley.edu/~Epliang/software/>

表 5 単語クラスタリングの結果の一部
Table 5 Examples of Word Clustering

単語	出現回数
predictably/RB	104
sadly/RB	119
nowadays/RB	131
theoretically/RB	156
notwithstanding/IN	188
frankly/RB	225
naturally/RB	446
occasionally/RB	568
lately/RB	1081
obviously/RB	1222
sometimes/RB	3215
now/RB	32451

表 6 セクション 23 における supertagging の精度の比較
Table 6 Accuracy of the supertagging on Section 23

モデル	クラスタ数	正解率 (正解/全トークン)	低頻度語 (正解/全トークン)
ベースライン	-	88.97% (45236/50844)	89.64% (2164/2410)
(1) 全ビット文字列	500	86.56% (44013/50844)	75.64% (1823/2410)
	1000	86.57% (44015/50844)	75.31% (1815/2410)
(2) 全ビット + 接頭辞	500	85.98% (43717/50844)	75.89% (1829/2410)
	1000	86.04% (43746/50844)	75.52% (1820/2410)
(3) 置換	500	84.62% (43026/50844)	74.36% (1792/2410)
	1000	84.45% (42944/50844)	71.49% (1723/2410)

解率の低下が著しい。

5. 結果の分析

今回の実験では、三種類の素性追加方法のいずれにおいても、精度向上を得ることができなかった。特に低頻度語について著しい精度低下が見られた。

今回用いた Penn Treebank は Wall Street Journal から作られたコーパスであるため、低頻度語は名詞が多く、2410 語中 1518 語 (63.0%) を占める (表 5)。特に固有名詞が一番多い。名詞の取り得る語彙項目の候補は、動詞の取り得る語彙項目の候補に比べ、とても少ない。実際、POS が名詞 (NN/NNS/NNP/NNS) である単語の取り得る平均語彙項目数は 46.0 個であるのに比べ、POS が動詞 (VB/VBD/VBG/VBN/VBP/VBZ) である場

表 7 低頻度語の POS 分類
Table 7 Parts-of-speech of infrequent words

POS	出現数
固有名詞 (単数・複数; NNP/NNPS)	883 (36.6%)
普通名詞 (単数・複数; NN/NNS)	635 (26.3%)
形容詞 (比較級・最上級含む; JJ/JJR/JJS)	462 (19.2%)
動詞 (VB/VBD/VBG/VBN/VBP/VBZ)	357 (14.8%)
副詞 (RB)	53 (2.2%)
外国語 (FW)	13 (0.5%)
数字の英語表記 (CD)	4 (0.1%)
感嘆詞 (UH)	1 (0.01%)
合計	2410

合は 143.8 個である。

今回の実験では、確率モデルに対して素性の足切りを実行したため、低頻度語そのものを使用した素性はほとんどモデル中に存在しない。そのためベースラインについては、低頻度語の語彙項目の判定にその単語の POS および周辺の単語・POS の情報が使われている。低頻度語に名詞が多く、そもそも語彙項目の候補が少ない点、および周辺の単語がしばしば非低頻度語であるという事実が、ベースラインにおける低頻度語のスコアの高さを説明していると考えられる。

前節において、今回使用した文法では語彙項目が POS の細分になっているということを述べた。このことはすなわち、語彙項目 (supertag) の判別にその単語の POS の情報が大きく貢献しているということを意味する。加えて、語彙項目に含まれる周辺の単語の制約情報 (COMPS 素性や SUBJ 素性) も POS の細分になっているため、周辺の単語の POS の情報も、語彙項目の判別に役立っている。

しかるに、今回のクラスタリング結果では、一クラスタに含まれる POS の種類数の平均および標準偏差 (表 8) を見ると、多くの種類の POS が一クラスタに含まれてしまっている。表 9 に、例として、あるクラスタに含まれていた POS の種類を示す。多くの POS が一つのクラスタに入っていることがわかる。この原因の一つとして、Brown らのアルゴリズムでは意味的なクラスタ、例えば “accounts/VBZ”, “account/VBP”, “accounted/VBD” や “evangelists/NNPS”, “predicts/NNS”, “observes/VBZ” などのようなクラスタを構成してしまうということが挙げられる。このように、複数の POS の情報をまとめてしまったクラスタ素性によって、重要な情報を担っていた POS の素性重みが相対的に下がったため、精度が下がってしまったと考えられる。

表 8 一クラスタ中の POS の種類数
Table 8 A number of Parts-of-speech in one cluster

	平均	標準偏差
500 クラスタ	10.968	3.247
1000 クラスタ	8.756	3.062

表 9 一クラスタに含まれる POS の種類例
Table 9 An example of Parts-of-speech in one cluster

POS	単語数
動詞三単現 (VBZ)	81 (42.4%)
動詞過去 (VBD)	38 (19.9%)
固有名詞単数 (NNP)	30 (15.7%)
名詞単数 (NN)	16 (8.4%)
名詞複数 (NNS)	15 (7.9%)
固有名詞複数 (NNP)	5 (2.6%)
副詞 (RB)	2 (1.0%)
形容詞 (JJ)	1 (0.5%)
動詞原型 (VB)	1 (0.5%)
動詞一人称単数現在 (VBN)	1 (0.5%)
動詞進行形 (VBG)	1 (0.5%)
計	191

一方、Miller らの固有表現タギング、Koo らの係り受け解析では有意な精度向上を得ている。この原因として、HPSG supertagging では割り当てるべき語彙項目がかなり多い (今回用いた文法では 1700 種類) のに比べ、固有表現タギングや dependency で用いられるタグ数は比較的少ないことが挙げられる。また、タスクそのものの性質が大きく異なることも原因の一つと考えられる。つまり、supertagging は単語に統語範疇を割り当てるタスクであり、固有表現のような意味情報の割り当てタスクと比べ、意味に関する素性が入れにくいと推測される。一方で Koo らの係り受け解析で精度向上がみられたのは、係り受け解析においては、係り先や係り元となる動詞や名詞の意味の情報が有効であったからと考えられる。

また、Brown らの単語クラスタリングアルゴリズムは bi-gram モデルを使っていることに注意したい。たしかに、bi-gram モデルによるクラスタリング結果は、ある種の統語的情報を含む。しかしその情報は、supertagging に必要な情報とは一致していなかったと考えられる。

6. おわりに

本研究では Brown らの単語クラスタリングを利用して、supertagging における新たな素性の導入方法を提案した。残念ながら、今回の提案手法では supertagging の精度改善には至らなかった。

今後の改良点として、Brown らの単語クラスタリングアルゴリズムに制約を加え、「同じ

POS が同じクラスに属するようにする」ということが挙げられる。また、クラス数を増やすことで、クラス数の粒度を上げることも考えられる。先に表 8 で挙げたように、クラス数が増えれば POS の種類は減少するため、語彙項目が POS の細分であるという特徴を活かすことができる。

133–142 (1996).

参 考 文 献

- 1) Bangalore, S. and Joshi, A.K.: Supertagging: an approach to almost parsing, *Computational Linguistics*, Vol.25, No.2, pp.237–265 (1999).
- 2) Clark, S.: Supertagging for combinatory categorial grammar, *Proceedings of the 6th International Workshop on Tree Adjoining Grammars and Related Frameworks (TAG+ 6)*, pp.19–24 (2002).
- 3) Clark, S. and Curran, J.R.: The importance of supertagging for wide-coverage CCG parsing, *COLING '04: Proceedings of the 20th international conference on Computational Linguistics*, Morristown, NJ, USA, Association for Computational Linguistics, p.282 (2004).
- 4) Koo, T., Carreras, X. and Collins, M.: Simple Semi-supervised Dependency Parsing, *Proceedings of ACL/HLT* (2008).
- 5) Brown, P.F., deSouza, P.V., Mercer, R.L., Pietra, V. J.D. and Lai, J.C.: Class-based n-gram models of natural language, *Computational Linguistics*, Vol.18, No.4, pp.467–479 (1992).
- 6) Miller, S., Guinness, J., Zamanian, A., Dumais, S., Marcu, D. and Roukos, S.: Name Tagging with Word Clusters and Discriminative Training, *HLT-NAACL 2004: Main Proceedings*.
- 7) Pollard, C. and Sag, I.: *Head-Driven Phrase Structure Grammar*, University Of Chicago Press (1994).
- 8) Matsuzaki, T., Miyao, Y. and Tsujii, J.: Efficient HPSG Parsing with Supertagging and CFG-filtering, *Proceedings of the International Joint Conference on Artificial Intelligence 2007*, pp.1671–1676.
- 9) Miyao, Y. and Tsujii, J.: Probabilistic disambiguation models for wide-coverage HPSG parsing, *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, Morristown, NJ, USA, Association for Computational Linguistics, pp.83–90 (2005).
- 10) Charniak, E.: A maximum-entropy-inspired parser, *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference*, San Francisco, CA, USA, Morgan Kaufmann Publishers Inc., pp.132–139 (2000).
- 11) Ratnaparkhi, A.: A maximum entropy model for part-of-speech tagging, *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp.