Extremely Lexicalized Models for Accurate and Fast HPSG Parsing

Takashi Ninomiya
Information Technology Center
University of Tokyo

Takuya Matsuzaki
Department of Computer Science
University of Tokyo

Yoshimasa Tsuruoka
School of Informatics
University of Manchester

Yusuke Miyao
Department of Computer Science
University of Tokyo

Jun'ichi Tsujii
Department of Computer Science, University of Tokyo
School of Informatics, University of Manchester
SORST, Japan Science and Technology Agency
Hongo 7-3-1, Bunkyo-ku, Tokyo, 113-0033, Japan
{ninomi, matuzaki, tsuruoka, yusuke, tsujii}@is.s.u-tokyo.ac.jp

Abstract

This paper describes an extremely lexicalized probabilistic model for fast and accurate HPSG parsing. In this model, the probabilities of parse trees are defined with only the probabilities of selecting lexical entries. The proposed model is very simple, and experiments revealed that the implemented parser runs around four times faster than the previous model and that the proposed model has a high accuracy comparable to that of the previous model for probabilistic HPSG, which is defined over phrase structures. We also developed a hybrid of our probabilistic model and the conventional phrase-structure-based model. The hybrid model is not only significantly faster but also significantly more accurate by two points of precision and recall compared to the previous model.

1 Introduction

For the last decade, accurate and wide-coverage parsing for real-world text has been intensively and extensively pursued. In most of state-of-the-art parsers, probabilistic events are defined over phrase structures because phrase structures are supposed to dominate syntactic configurations of sentences. For example, probabilities were defined over grammar rules in probabilistic CFG (Collins, 1999; Klein and Manning, 2003; Charniak and Johnson, 2005) or over complex phrase structures of head-driven phrase structure grammar (HPSG) or combinatory categorial grammar (CCG) (Clark and Curran, 2004b; Malouf and van Noord, 2004; Miyao and Tsujii, 2005). Although these studies vary in the design of the probabilistic models, the fundamental conception of probabilistic modeling is intended to capture characteristics of phrase structures or grammar rules. Although lexical information, such as head words, is known to significantly improve the parsing accuracy, it was also used to augment information on phrase structures.

Another interesting approach to this problem was using supertagging (Clark and Curran, 2004b; Clark and Curran, 2004a; Wang and Harper, 2004; Nasr and Rambow, 2004), which was originally developed for lexicalized tree adjoining grammars (LTAG) (Bangalore and Joshi, 1999). Supertagging is a process where words in an input sentence are tagged with 'supertags,' which are lexical entries in lexicalized grammars, e.g., elementary trees in LTAG, lexical categories in CCG, and lexical entries in HPSG. Supertagging was, in the first place, a technique to reduce the cost of parsing with lexicalized grammars; ambiguity in assigning lexical entries to words is reduced by the light-weight process of supertagging before the heavy process of parsing. Bangalore and Joshi (1999) claimed that if words can be assigned correct supertags, syntactic parsing is almost trivial. What this means is that if supertags are correctly assigned, syntactic structures are almost de-

Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP 2006), pages 155–163,
Sydney, July 2006.

termined because supertags include rich syntactic information such as subcategorization frames. Nasr and Rambow (2004) showed that the accuracy of LTAG parsing reached about 97%, assuming that the correct supertags were given. The concept of supertagging is simple and interesting, and the effects of this were recently demonstrated in the case of a CCG parser (Clark and Curran, 2004a) with the result of a drastic improvement in the parsing speed. Wang and Harper (2004) also demonstrated the effects of supertagging with a statistical constraint dependency grammar (CDG) parser. They achieved accuracy as high as the state-of-the-art parsers. However, a supertagger itself was used as an external tagger that enumerates candidates of lexical entries or filters out unlikely lexical entries just to help parsing, and the best parse trees were selected mainly according to the probabilistic model for phrase structures or dependencies with/without the probabilistic model for supertagging.

We investigate an extreme case of HPSG parsing in which the probabilistic model is defined with only the probabilities of lexical entry selection; i.e., the model is never sensitive to characteristics of phrase structures. The model is simply defined as the product of the supertagging probabilities, which are provided by the discriminative method with machine learning features of word trigrams and part-of-speech (POS) 5-grams as defined in the CCG supertagging (Clark and Curran, 2004a). The model is implemented in an HPSG parser instead of the phrase-structure-based probabilistic model; i.e., the parser returns the parse tree assigned the highest probability of supertagging among the parse trees licensed by an HPSG. Though the model uses only the probabilities of lexical entry selection, the experiments revealed that it was as accurate as the previous phrase-structure-based model. Interestingly, this means that accurate parsing is possible using rather simple mechanisms.

We also tested a hybrid model of the supertagging and the previous phrase-structure-based probabilistic model. In the hybrid model, the probabilities of the previous model are multiplied by the supertagging probabilities instead of a *preliminary probabilistic model*, which is introduced to help the process of estimation by filtering unlikely lexical entries (Miyao and Tsujii, 2005). In the previous model, the preliminary probabilistic model is defined as the probability of unigram supertagging. So, the hybrid model can be regarded as an extension of supertagging from unigram to n-gram. The hybrid model can also be regarded as a variant of the statistical CDG parser (Wang, 2003; Wang and Harper, 2004), in which the parse tree probabilities are defined as the product of the supertagging probabilities and the dependency probabilities. In the experiments, we observed that the hybrid model significantly improved the parsing speed, by around three to four times speed-ups, and accuracy, by around two points in both precision and recall, over the previous model. This implies that finer probabilistic model of lexical entry selection can improve the phrase-structure-based model.

2 HPSG and probabilistic models

HPSG (Pollard and Sag, 1994) is a syntactic theory based on lexicalized grammar formalism. In HPSG, a small number of schemata describe general construction rules, and a large number of lexical entries express word-specific characteristics. The structures of sentences are explained using combinations of schemata and lexical entries. Both schemata and lexical entries are represented by typed feature structures, and constraints represented by feature structures are checked with *unification*.

An example of HPSG parsing of the sentence "*Spring has come*" is shown in Figure 1. First, each of the lexical entries for "*has*" and "*come*" is unified with a daughter feature structure of the Head-Complement Schema. Unification provides the phrasal sign of the mother. The sign of the larger constituent is obtained by repeatedly applying schemata to lexical/phrasal signs. Finally, the parse result is output as a phrasal sign that dominates the sentence.

Given a set $\mathcal{W}$ of words and a set $\mathcal{F}$ of feature structures, an HPSG is formulated as a tuple, $G = \langle L, R \rangle$, where

$$L = \{l = \langle w, F \rangle | w \in \mathcal{W}, F \in \mathcal{F}\} \text{ is a set of lexical entries, and}$$
$$R \text{ is a set of schemata; i.e., } r \in R \text{ is a partial function: } \mathcal{F} \times \mathcal{F} \rightarrow \mathcal{F}.$$

Given a sentence, an HPSG computes a set of phrasal signs, i.e., feature structures, as a result of parsing. Note that HPSG is one of the lexicalized grammar formalisms, in which lexical entries determine the dominant syntactic structures.

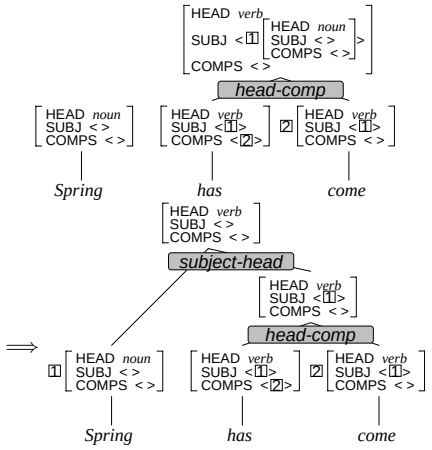

Figure 1: HPSG parsing.

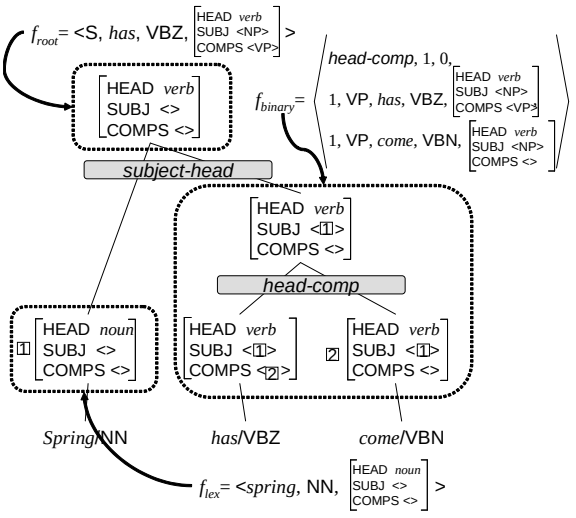

Figure 2: Example of features.

Previous studies (Abney, 1997; Johnson et al., 1999; Riezler et al., 2000; Malouf and van Noord, 2004; Kaplan et al., 2004; Miyao and Tsujii, 2005) defined a probabilistic model of unification-based grammars including HPSG as a *log-linear model* or *maximum entropy model* (Berger et al., 1996). The probability that a parse result T is assigned to a given sentence $\mathbf{w} = \langle w_1, \ldots, w_n \rangle$ is

$$p_{hpsg}(T|\mathbf{w}) = \frac{1}{Z_{\mathbf{w}}} \exp\left(\sum_u \lambda_u f_u(T)\right)$$

$$Z_{\mathbf{w}} = \sum_{T'} \exp\left(\sum_u \lambda_u f_u(T')\right),$$

where λ_u is a model parameter, f_u is a feature function that represents a characteristic of parse tree T, and $Z_{\mathbf{w}}$ is the sum over the set of all possible parse trees for the sentence. Intuitively, the probability is defined as the normalized product of the weights $\exp(\lambda_u)$ when a characteristic corresponding to f_u appears in parse result T. The model parameters, λ_u, are estimated using numerical optimization methods (Malouf, 2002) to maximize the log-likelihood of the training data.

However, the above model cannot be easily estimated because the estimation requires the computation of $p(T|\mathbf{w})$ for all parse candidates assigned to sentence $\mathbf{w}$. Because the number of parse candidates is exponentially related to the length of the sentence, the estimation is intractable for long sentences. To make the model estimation tractable, Geman and Johnson (Geman and Johnson, 2002) and Miyao and Tsujii (Miyao and Tsujii, 2002) proposed a dynamic programming algorithm for estimating $p(T|\mathbf{w})$. Miyao and Tsujii (2005) also introduced a *preliminary probabilistic model* $p_0(T|\mathbf{w})$ whose estimation does not require the parsing of a treebank. This model is introduced as a reference distribution of the probabilistic HPSG model; i.e., the computation of parse trees given low probabilities by the model is omitted in the estimation stage. We have

(Previous probabilistic HPSG)

$$p_{hpsg'}(T|\mathbf{w}) = p_0(T|\mathbf{w}) \frac{1}{Z_{\mathbf{w}}} \exp\left(\sum_u \lambda_u f_u(T)\right)$$

$$Z_{\mathbf{w}} = \sum_{T'} p_0(T'|\mathbf{w}) \exp\left(\sum_u \lambda_u f_u(T')\right)$$

$$p_0(T|\mathbf{w}) = \prod_{i=1}^{n} p(l_i|w_i),$$

where l_i is a lexical entry assigned to word w_i in T and $p(l_i|w_i)$ is the probability of selecting lexical entry l_i for w_i.

In the experiments, we compared our model with the probabilistic HPSG model of Miyao and Tsujii (2005). The features used in their model are combinations of the feature templates listed in Table 1. The feature templates f_{binary} and f_{unary} are defined for constituents at binary and unary branches, f_{root} is a feature template set for the root nodes of parse trees, and f_{lex} is a feature template set for calculating the preliminary probabilistic model. An example of features applied to the parse tree for the sentence "*Spring has come*" is shown in Figure 2.

$$
\begin{aligned}
f_{binary} &= \left\langle \begin{array}{l} r, d, c, \\ sp_l, sy_l, hw_l, hp_l, hl_l, \\ sp_r, sy_r, hw_r, hp_r, hl_r \end{array} \right\rangle \\
f_{unary} &= \langle r, sy, hw, hp, hl \rangle \\
f_{root} &= \langle sy, hw, hp, hl \rangle \\
f_{lex} &= \langle w_i, p_i, l_i \rangle
\end{aligned}
$$

combinations of feature templates for f_{binary}

$\langle r, d, c, hw, hp, hl \rangle$, $\langle r, d, c, hw, hp \rangle$, $\langle r, d, c, hw, hl \rangle$, $\langle r, d, c, sy, hw \rangle$, $\langle r, c, sp, hw, hp, hl \rangle$, $\langle r, c, sp, hw, hp \rangle$, $\langle r, c, sp, hw, hl \rangle$, $\langle r, c, sp, sy, hw \rangle$, $\langle r, d, c, hp, hl \rangle$, $\langle r, d, c, hp \rangle$, $\langle r, d, c, hl \rangle$, $\langle r, d, c, sy \rangle$, $\langle r, c, sp, hp, hl \rangle$, $\langle r, c, sp, hp \rangle$, $\langle r, c, sp, hl \rangle$, $\langle r, c, sp, sy \rangle$

combinations of feature templates for f_{unary}

$\langle r, hw, hp, hl \rangle$, $\langle r, hw, hp \rangle$, $\langle r, hw, hl \rangle$, $\langle r, sy, hw \rangle$, $\langle r, hp, hl \rangle$, $\langle r, hp \rangle$, $\langle r, hl \rangle$, $\langle r, sy \rangle$

combinations of feature templates for f_{root}

$\langle hw, hp, hl \rangle$, $\langle hw, hp \rangle$, $\langle hw, hl \rangle$, $\langle sy, hw \rangle$, $\langle hp, hl \rangle$, $\langle hp \rangle$, $\langle hl \rangle$, $\langle sy \rangle$

combinations of feature templates for f_{lex}

$\langle w_i, p_i, l_i \rangle$, $\langle p_i, l_i \rangle$

r	name of the applied schema
d	distance between the head words of the daughters
c	whether a comma exists between daughters and/or inside daughter phrases
sp	number of words dominated by the phrase
sy	symbol of the phrasal category
hw	surface form of the head word
hp	part-of-speech of the head word
hl	lexical entry assigned to the head word
w_i	i-th word
p_i	part-of-speech for w_i
l_i	lexical entry for w_i

Table 1: Features.

3 Extremely lexicalized probabilistic models

In the experiments, we tested parsing with the previous model for the probabilistic HPSG explained in Section 2 and other three types of probabilistic models defined with the probabilities of lexical entry selection. The first one is the simplest probabilistic model, which is defined with only the probabilities of lexical entry selection. It is defined simply as the product of the probabilities of selecting all lexical entries in the sentence; i.e., the model does not use the probabilities of phrase structures like the previous models.

Given a set of lexical entries, L, a sentence, $\mathbf{w} = \langle w_1, \ldots, w_n \rangle$, and the probabilistic model of lexical entry selection, $p(l_i \in L | \mathbf{w}, i)$, the first model is formally defined as follows:

(Model 1)

$$p_{model1}(T|\mathbf{w}) = \prod_{i=1}^{n} p(l_i|\mathbf{w}, i),$$

where l_i is a lexical entry assigned to word w_i in T and $p(l_i|\mathbf{w}, i)$ is the probability of selecting lexical entry l_i for w_i.

The second model is defined as the product of the probabilities of selecting all lexical entries in the sentence and the root node probability of the parse tree. That is, the second model is also defined without the probabilities on phrase structures:

(Model 2)

$$p_{model2}(T|\mathbf{w}) = \frac{1}{Z_{model2}} p_{model1}(T|\mathbf{w}) \exp\left(\sum_{\substack{u \\ (f_u \in f_{root})}} \lambda_u f_u(T) \right)$$

$$Z_{model2} = \sum_{T'} p_{model1}(T'|\mathbf{w}) \exp\left(\sum_{\substack{u \\ (f_u \in f_{root})}} \lambda_u f_u(T') \right),$$

where Z_{model2} is the sum over the set of all possible parse trees for the sentence.

The third model is a hybrid of model 1 and the previous model. The probabilities of the lexical entries in the previous model are replaced with the probabilities of lexical entry selection:

(Model 3)

$$p_{model3}(T|\mathbf{w}) = \frac{1}{Z_{model3}} p_{model1}(T|\mathbf{w}) \exp\left(\sum_u \lambda_u f_u(T) \right)$$

$$Z_{model3} = \sum_{T'} p_{model1}(T'|\mathbf{w}) \exp\left(\sum_u \lambda_u f_u(T') \right).$$

In this study, the same model parameters used in the previous model were used for phrase structures.

The probabilities of lexical entry selection, $p(l_i|\mathbf{w}, i)$, are defined as follows:

(Probabilistic Model of Lexical Entry Selection)

$$p(l_i|\mathbf{w}, i) = \frac{1}{Z_w} \exp\left(\sum_u \lambda_u f_u(l_i, \mathbf{w}, i) \right)$$

$$f_{exlex} = \left\langle \begin{array}{l} w_{i-1}, w_i, w_{i+1}, \\ p_{i-2}, p_{i-1}, p_i, p_{i+1}, p_{i+2} \end{array} \right\rangle$$

combinations of feature templates
$\langle w_{i-1} \rangle, \langle w_i \rangle, \langle w_{i+1} \rangle,$
$\langle p_{i-2} \rangle, \langle p_{i-1} \rangle, \langle p_i \rangle, \langle p_{i+1} \rangle, \langle p_{i+2} \rangle, \langle p_{i+3} \rangle,$
$\langle w_{i-1}, w_i \rangle, \langle w_i, w_{i+1} \rangle,$
$\langle p_{i-1}, w_i \rangle, \langle p_i, w_i \rangle, \langle p_{i+1}, w_i \rangle,$
$\langle p_i, p_{i+1}, p_{i+2}, p_{i+3} \rangle, \langle p_{i-2}, p_{i-1}, p_i \rangle,$
$\langle p_{i-1}, p_i, p_{i+1} \rangle, \langle p_i, p_{i+1}, p_{i+2} \rangle$
$\langle p_{i-2}, p_{i-1} \rangle, \langle p_{i-1}, p_i \rangle, \langle p_i, p_{i+1} \rangle, \langle p_{i+1}, p_{i+2} \rangle$

Table 2: Features for the probabilities of lexical entry selection.

procedure Parsing($\langle w_1, \ldots, w_n \rangle$, $\langle L, R \rangle$, α, β, κ, δ, θ)
  for $i = 1$ to n
    foreach $F' \in \{F | \langle w_i, F \rangle \in L\}$
      $p = \sum_u \lambda_u f_u(F')$
      $\pi[i-1, i] \leftarrow \pi[i-1, i] \cup \{F'\}$
      if $(p > \rho[i-1, i, F'])$ **then**
        $\rho[i-1, i, F'] \leftarrow p$
    LocalThresholding($i-1$, i,α, β)
  for $d = 1$ to n
    for $i = 0$ to $n - d$
      $j = i + d$
      for $k = i + 1$ to $j - 1$
        foreach $F_s \in \phi[i, k],\ F_t \in \phi[k, j],\ r \in R$
          if $F = r(F_s, F_t)$ **has succeeded**
            $p = \rho[i, k, F_s] + \rho[k, j, F_t] + \sum_u \lambda_u f_u(F)$
            $\pi[i, j] \leftarrow \pi[i, j] \cup \{F\}$
            if $(p > \rho[i, j, F])$ **then**
              $\rho[i, j, F] \leftarrow p$
      LocalThresholding(i, j,κ, δ)
    GlobalThresholding(i, n, θ)

procedure IterativeParsing($\mathbf{w}$, G, α_0, β_0, κ_0, δ_0, θ_0, $\Delta\alpha$, $\Delta\beta$, $\Delta\kappa$, $\Delta\delta$, $\Delta\theta$, α_{last}, β_{last}, κ_{last}, δ_{last}, θ_{last})
  $\alpha \leftarrow \alpha_0$; $\beta \leftarrow \beta_0$; $\kappa \leftarrow \kappa_0$; $\delta \leftarrow \delta_0$; $\theta \leftarrow \theta_0$;
  loop while $\alpha \leq \alpha_{\text{last}}$ and $\beta \leq \beta_{\text{last}}$ and $\kappa \leq \kappa_{\text{last}}$ and $\delta \leq \delta_{\text{last}}$ and $\theta \leq \theta_{\text{last}}$
    call Parsing($\mathbf{w}$, G, α, β, κ, δ, θ)
    if $\pi[1, n] \neq \emptyset$ **then exit**
    $\alpha \leftarrow \alpha + \Delta\alpha$; $\beta \leftarrow \beta + \Delta\beta$;
    $\kappa \leftarrow \kappa + \Delta\kappa$; $\delta \leftarrow \delta + \Delta\delta$; $\theta \leftarrow \theta + \Delta\theta$;

Figure 3: Pseudo-code of iterative parsing for HPSG.

$$Z_w = \sum_{l'} \exp\left(\sum_u \lambda_u f_u(l', \mathbf{w}, i) \right),$$

where Z_w is the sum over all possible lexical entries for the word w_i. The feature templates used in our model are listed in Table 2 and are word trigrams and POS 5-grams.

4 Experiments

4.1 Implementation

We implemented the iterative parsing algorithm (Ninomiya et al., 2005) for the probabilistic HPSG models. It first starts parsing with a narrow beam. If the parsing fails, then the beam is widened, and parsing continues until the parser outputs results or the beam width reaches some limit. Though the probabilities of lexical entry selection are introduced, the algorithm for the presented probabilistic models is almost the same as the original iterative parsing algorithm.

The pseudo-code of the algorithm is shown in Figure 3. In the figure, the $\pi[i, j]$ represents the set of partial parse results that cover words $w_{i+1}, \ldots, w_j$, and $\rho[i, j, F]$ stores the maximum figure-of-merit (FOM) of partial parse result F at cell (i, j). The probability of lexical entry F is computed as $\sum_u \lambda_u f_u(F)$ for the previous model, as shown in the figure. The probability of a lexical entry for models 1, 2, and 3 is computed as the probability of lexical entry selection, $p(F|\mathbf{w}, i)$. The FOM of a newly created partial parse, F, is computed by summing the values of ρ of the daughters and an additional FOM of F if the model is the previous model or model 3. The FOM for models 1 and 2 is computed by only summing the values of ρ of the daughters; i.e., weights $\exp(\lambda_u)$ in the figure are assigned zero. The terms κ and δ are the thresholds of the number of phrasal signs in the chart cell and the beam width for signs in the chart cell. The terms α and β are the thresholds of the number and the beam width of lexical entries, and θ is the beam width for global thresholding (Goodman, 1997).

4.2 Evaluation

We evaluated the speed and accuracy of parsing with extremely lexicalized models by using Enju 2.1, the HPSG grammar for English (Miyao et al., 2005; Miyao and Tsujii, 2005). The lexicon of the grammar was extracted from Sections 02-21 of the Penn Treebank (Marcus et al., 1994) (39,832 sentences). The grammar consisted of 3,797 lexical entries for 10,536 words[1]. The probabilistic models were trained using the same portion of the treebank. We used beam thresholding, global thresholding (Goodman, 1997), preserved iterative parsing (Ninomiya et al., 2005) and other tech-

[1] An HPSG treebank is automatically generated from the Penn Treebank. Those lexical entries were generated by applying lexical rules to observed lexical entries in the HPSG treebank (Nakanishi et al., 2004). The lexicon, however, included many lexical entries that do not appear in the HPSG treebank. The HPSG treebank is used for training the probabilistic model for lexical entry selection, and hence, those lexical entries that do not appear in the treebank are rarely selected by the probabilistic model. The 'effective' tag set size, therefore, is around 1,361, the number of lexical entries without those never-seen lexical entries.

	No. of tested sentences		Total No. of	Avg. length of tested sentences	
	$\leq$ 40 words	$\leq$ 100 words	sentences	$\leq$ 40 words	$\leq$ 100 words
Section 23	2,162 (94.04%)	2,299 (100.00%)	2,299	20.7	22.2
Section 24	1,157 (92.78%)	1,245 (99.84%)	1,247	21.2	23.0

Table 3: Statistics of the Penn Treebank.

	Section 23 ($\leq$ 40 + Gold POSs)					Section 23 ($\leq$ 100 + Gold POSs)				
	LP (%)	LR (%)	UP (%)	UR (%)	Avg. time (ms)	LP (%)	LR (%)	UP (%)	UR (%)	Avg. time (ms)
previous model	87.65	86.97	91.13	90.42	468	87.26	86.50	90.73	89.93	604
model 1	87.54	86.85	90.38	89.66	111	87.23	86.47	90.05	89.27	129
model 2	87.71	87.02	90.51	89.80	109	87.38	86.62	90.17	89.39	130
model 3	89.79	88.97	92.66	91.81	132	89.48	88.58	92.33	91.40	152
	Section 23 ($\leq$ 40 + POS tagger)					Section 23 ($\leq$ 100 + POS tagger)				
	LP (%)	LR (%)	UP (%)	UR (%)	Avg. time (ms)	LP (%)	LR (%)	UP (%)	UR (%)	Avg. time (ms)
previous model	85.33	84.83	89.93	89.41	509	84.96	84.25	89.55	88.80	674
model 1	85.26	84.31	89.17	88.18	133	85.00	84.01	88.85	87.82	154
model 2	85.37	84.42	89.25	88.26	134	85.08	84.09	88.91	87.88	155
model 3	87.66	86.53	91.61	90.43	155	87.35	86.29	91.24	90.13	183

Table 4: Experimental results for Section 23.

niques for deep parsing[2]. The parameters for beam searching were determined manually by trial and error using Section 22: $\alpha_0 = 4, \Delta\alpha = 4, \alpha_{\text{last}} = 20, \beta_0 = 1.0, \Delta\beta = 2.5, \beta_{\text{last}} = 11.0, \delta_0 = 12, \Delta\delta = 4, \delta_{\text{last}} = 28$, $\kappa_0 = 6.0, \Delta\kappa = 2.25, \kappa_{\text{last}} = 15.0$, $\theta_0 = 8.0, \Delta\theta = 3.0$, and $\theta_{\text{last}} = 20.0$. With these thresholding parameters, the parser iterated at most five times for each sentence.

We measured the accuracy of the predicate-argument relations output of the parser. A predicate-argument relation is defined as a tuple $\langle \sigma, w_h, a, w_a \rangle$, where σ is the predicate type (e.g., adjective, intransitive verb), w_h is the head word of the predicate, a is the argument label (MODARG, ARG1, ..., ARG4), and w_a is the head word of the argument. Labeled precision (LP)/labeled recall (LR) is the ratio of tuples correctly identified by the parser[3]. Unlabeled precision (UP)/unlabeled recall (UR) is the ratio of tuples without the predicate type and the argument label. This evaluation scheme was the same as used in previous evaluations of lexicalized grammars (Hockenmaier, 2003; Clark and Curran, 2004b; Miyao and Tsujii, 2005). The experiments were conducted on an AMD Opteron server with a 2.4-GHz CPU. Section 22 of the Treebank was used as the development set, and the performance was evaluated using sentences of $\leq$ 40 and 100 words in Section 23. The performance of each parsing technique was analyzed using the sentences in Section 24 of $\leq$ 100 words. Table 3 details the numbers and average lengths of the tested sentences of $\leq$ 40 and 100 words in Sections 23 and 24, and the total numbers of sentences in Sections 23 and 24.

The parsing performance for Section 23 is shown in Table 4. The upper half of the table shows the performance using the correct POSs in the Penn Treebank, and the lower half shows the performance using the POSs given by a POS tagger (Tsuruoka and Tsujii, 2005). The left and right sides of the table show the performances for the sentences of $\leq$ 40 and $\leq$ 100 words. Our models significantly increased not only the parsing speed but also the parsing accuracy. Model 3 was around three to four times faster and had around two points higher precision and recall than the previous model. Surprisingly, model 1, which used only lexical information, was very fast and as accurate as the previous model. Model 2 also improved the accuracy slightly without information of phrase structures. When the automatic POS tagger was introduced, both precision and recall dropped by around 2 points, but the tendency towards improved speed and accuracy was again ob-

[2]Deep parsing techniques include quick check (Malouf et al., 2000) and large constituent inhibition (Kaplan et al., 2004) as described by Ninomiya et al. (2005), but hybrid parsing with a CFG chunk parser was not used. This is because we did not observe a significant improvement for the development set by the hybrid parsing and observed only a small improvement in the parsing speed by around 10 ms.

[3]When parsing fails, precision and recall are evaluated, although nothing is output by the parser; i.e., recall decreases greatly.

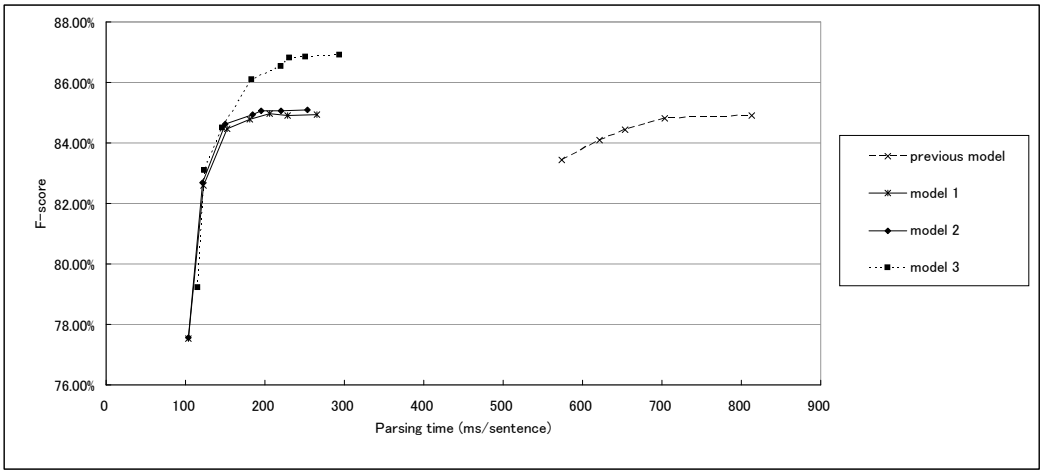

Figure 4: F-score versus average parsing time for sentences in Section 24 of ≤ 100 words.

served.

The unlabeled precisions and recalls of the previous model and models 1, 2, and 3 were significantly different as measured using stratified shuffling tests (Cohen, 1995) with p-values < 0.05. The labeled precisions and recalls were significantly different among models 1, 2, and 3 and between the previous model and model 3, but were not significantly different between the previous model and model 1 and between the previous model and model 2.

The average parsing time and labeled F-score curves of each probabilistic model for the sentences in Section 24 of ≤ 100 words are graphed in Figure 4. The superiority of our models is clearly observed in the figure. Model 3 performed significantly better than the previous model. Models 1 and 2 were significantly faster with almost the same accuracy as the previous model.

5 Discussion

5.1 Supertagging

Our probabilistic model of lexical entry selection can be used as an independent classifier for selecting lexical entries, which is called the supertagger (Bangalore and Joshi, 1999; Clark and Curran, 2004b). The CCG supertagger uses a maximum entropy classifier and is similar to our model.

We evaluated the performance of our probabilistic model as a supertagger. The accuracy of the resulting supertagger on our development set (Section 22) is given in Table 5 and Table 6. The test sentences were automatically POS-tagged. Results of other supertaggers for automatically extracted lexicalized grammars are listed in Table 5. Table 6 gives the average number of supertags assigned to a word, the per-word accuracy, and the sentence accuracy for several values of γ, which is a parameter to determine how many lexical entries are assigned.

	test data	accuracy (%)
HPSG supertagger (this paper)	22	87.51
CCG supertagger (Curran and Clark, 2003)	00/23	91.70 / 91.45
LTAG supertagger (Shen and Joshi, 2003)	22/23	86.01 / 86.27

Table 5: Accuracy of single-tag supertaggers. The numbers under "test data" are the PTB section numbers of the test data.

γ	tags/word	word acc. (%)	sentence acc. (%)
1e-1	1.30	92.64	34.98
1e-2	2.11	95.08	46.11
1e-3	4.66	96.22	51.95
1e-4	10.72	96.83	55.66
1e-5	19.93	96.95	56.20

Table 6: Accuracy of multi-supertagging.

When compared with other supertag sets of automatically extracted lexicalized grammars, the (effective) size of our supertag set, 1,361 lexical entries, is between the CCG supertag set (398 categories) used by Curran and Clark (2003) and the LTAG supertag set (2920 elementary trees) used by Shen and Joshi (2003). The relative order based on the sizes of the tag sets exactly matches the order based on the accuracies of corresponding supertaggers.

5.2 Efficacy of extremely lexicalized models

The implemented parsers of models 1 and 2 were around four times faster than the previous model without a loss of accuracy. However, what surprised us is not the speed of the models, but the fact that they were as accurate as the previous model, though they do not use any phrase-structure-based probabilities. We think that the correct parse is more likely to be selected if the correct lexical entries are assigned high probabilities because lexical entries include specific information about subcategorization frames and syntactic alternation, such as wh-movement and passivization, that likely determines the dominant structures of parse trees. Another possible reason for the accuracy is the constraints placed by unification-based grammars. That is, incorrect parse trees were suppressed by the constraints.

The best performer in terms of speed and accuracy was model 3. The increased speed was, of course, possible for the same reasons as the speeds of models 1 and 2. An unexpected but very impressive result was the significant improvement of accuracy by two points in precision and recall, which is hard to attain by tweaking parameters or hacking features. This may be because the phrase structure information and lexical information complementarily improved the model. The lexical information includes more specific information about the syntactic alternation, and the phrase structure information includes information about the syntactic structures, such as the distances of head words or the sizes of phrases.

Nasr and Rambow (2004) showed that the accuracy of LTAG parsing reached about 97%, assuming that the correct supertags were given. We exemplified the dominance of lexical information in real syntactic parsing, i.e., syntactic parsing without gold-supertags, by showing that the probabilities of lexical entry selection dominantly contributed to syntactic parsing.

The CCG supertagging demonstrated fast and accurate parsing for the probabilistic CCG (Clark and Curran, 2004a). They used the supertagger for eliminating candidates of lexical entries, and the probabilities of parse trees were calculated using the phrase-structure-based model without the probabilities of lexical entry selection. Our study is essentially different from theirs in that the probabilities of lexical entry selection have been demonstrated to dominantly contribute to the disambiguation of phrase structures.

We have not yet investigated whether our results can be reproduced with other lexicalized grammars. Our results might hold only for HPSG because HPSG has strict feature constraints and has lexical entries with rich syntactic information such as wh-movement.

6 Conclusion

We developed an extremely lexicalized probabilistic model for fast and accurate HPSG parsing. The model is very simple. The probabilities of parse trees are defined with only the probabilities of selecting lexical entries, which are trained by the discriminative methods in the log-linear model with features of word trigrams and POS 5-grams as defined in the CCG supertagging. Experiments revealed that the model achieved impressive accuracy as high as that of the previous model for the probabilistic HPSG and that the implemented parser runs around four times faster. This indicates that accurate and fast parsing is possible using rather simple mechanisms. In addition, we provided another probabilistic model, in which the probabilities for the leaf nodes in a parse tree are given by the probabilities of supertagging, and the probabilities for the intermediate nodes are given by the previous phrase-structure-based model. The experiments demonstrated not only speeds significantly increased by three to four times but also impressive improvement in parsing accuracy by around two points in precision and recall.

We hope that this research provides a novel approach to deterministic parsing in which only lexical selection and little phrasal information without packed representations dominates the parsing strategy.

References

Steven P. Abney. 1997. Stochastic attribute-value grammars. *Computational Linguistics*, 23(4):597–618.

Srinivas Bangalore and Aravind Joshi. 1999. Supertagging: An approach to almost parsing. *Computational Linguistics*, 25(2):237–265.

Adam Berger, Stephen Della Pietra, and Vincent Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71.

Eugene Charniak and Mark Johnson. 2005. Coarse-to-fine n-best parsing and maxent discriminative reranking. In *Proc. of ACL'05*, pages 173–180.

Stephen Clark and James R. Curran. 2004a. The importance of supertagging for wide-coverage CCG parsing. In *Proc. of COLING-04*.

Stephen Clark and James R. Curran. 2004b. Parsing the WSJ using CCG and log-linear models. In *Proc. of ACL'04*, pages 104–111.

Paul R. Cohen. 1995. *Empirical Methods for Artificial Intelligence*. The MIT Press.

Michael Collins. 1999. *Head-Driven Statistical Models for Natural Language Parsing*. Ph.D. thesis, Univ. of Pennsylvania.

James R. Curran and Stephen Clark. 2003. Investigating GIS and smoothing for maximum entropy taggers. In *Proc. of EACL'03*, pages 91–98.

Stuart Geman and Mark Johnson. 2002. Dynamic programming for parsing and estimation of stochastic unification-based grammars. In *Proc. of ACL'02*, pages 279–286.

Joshua Goodman. 1997. Global thresholding and multiple pass parsing. In *Proc. of EMNLP-1997*, pages 11–25.

Julia Hockenmaier. 2003. Parsing with generative models of predicate-argument structure. In *Proc. of ACL'03*, pages 359–366.

Mark Johnson, Stuart Geman, Stephen Canon, Zhiyi Chi, and Stefan Riezler. 1999. Estimators for stochastic "unification-based" grammars. In *Proc. of ACL '99*, pages 535–541.

R. M. Kaplan, S. Riezler, T. H. King, J. T. Maxwell III, and A. Vasserman. 2004. Speed and accuracy in shallow and deep stochastic parsing. In *Proc. of HLT/NAACL'04*.

Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *Proc. of ACL'03*, pages 423–430.

Robert Malouf and Gertjan van Noord. 2004. Wide coverage parsing with stochastic attribute value grammars. In *Proc. of IJCNLP-04 Workshop "Beyond Shallow Analyses"*.

Robert Malouf, John Carroll, and Ann Copestake. 2000. Efficient feature structure operations without compilation. *Journal of Natural Language Engineering*, 6(1):29–46.

Robert Malouf. 2002. A comparison of algorithms for maximum entropy parameter estimation. In *Proc. of CoNLL-2002*, pages 49–55.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1994. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Yusuke Miyao and Jun'ichi Tsujii. 2002. Maximum entropy estimation for feature forests. In *Proc. of HLT 2002*, pages 292–297.

Yusuke Miyao and Jun'ichi Tsujii. 2005. Probabilistic disambiguation models for wide-coverage HPSG parsing. In *Proc. of ACL'05*, pages 83–90.

Yusuke Miyao, Takashi Ninomiya, and Jun'ichi Tsujii, 2005. *Keh-Yih Su, Jun'ichi Tsujii, Jong-Hyeok Lee and Oi Yee Kwong (Eds.), Natural Language Processing - IJCNLP 2004 LNAI 3248*, chapter Corpus-oriented Grammar Development for Acquiring a Head-driven Phrase Structure Grammar from the Penn Treebank, pages 684–693. Springer-Verlag.

Hiroko Nakanishi, Yusuke Miyao, and Jun'ichi Tsujii. 2004. An empirical investigation of the effect of lexical rules on parsing with a treebank grammar. In *Proc. of TLT'04*, pages 103–114.

Alexis Nasr and Owen Rambow. 2004. Supertagging and full parsing. In *Proc. of the 7th International Workshop on Tree Adjoining Grammar and Related Formalisms (TAG+7)*.

Takashi Ninomiya, Yoshimasa Tsuruoka, Yusuke Miyao, and Jun'ichi Tsujii. 2005. Efficacy of beam thresholding, unification filtering and hybrid parsing in probabilistic hpsg parsing. In *Proc. of IWPT 2005*, pages 103–114.

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

Stefan Riezler, Detlef Prescher, Jonas Kuhn, and Mark Johnson. 2000. Lexicalized stochastic modeling of constraint-based grammars using log-linear measures and EM training. In *Proc. of ACL'00*, pages 480–487.

Libin Shen and Aravind K. Joshi. 2003. A SNoW based supertagger with application to NP chunking. In *Proc. of ACL'03*, pages 505–512.

Yoshimasa Tsuruoka and Jun'ichi Tsujii. 2005. Bidirectional inference with the easiest-first strategy for tagging sequence data. In *Proc. of HLT/EMNLP 2005*, pages 467–474.

Wen Wang and Mary P. Harper. 2004. A statistical constraint dependency grammar (CDG) parser. In *Proc. of ACL'04 Incremental Parsing workshop: Bringing Engineering and Cognition Together*, pages 42–49.

Wen Wang. 2003. *Statistical Parsing and Language Modeling based on Constraint Dependency Grammar*. Ph.D. thesis, Purdue University.