A log-linear model with an n-gram reference distribution for accurate HPSG parsing

Takashi Ninomiya
Information Technology Center
University of Tokyo
ninomi@r.dl.itc.u-tokyo.ac.jp

Takuya Matsuzaki
Department of Computer Science
University of Tokyo
matuzaki@is.s.u-tokyo.ac.jp

Yusuke Miyao
Department of Computer Science
University of Tokyo
yusuke@is.s.u-tokyo.ac.jp

Jun'ichi Tsujii
Department of Computer Science, University of Tokyo
School of Informatics, University of Manchester
NaCTeM (National Center for Text Mining)
tsujii@is.s.u-tokyo.ac.jp
Hongo 7-3-1, Bunkyo-ku, Tokyo, 113-0033, Japan

Abstract

This paper describes a log-linear model with an n-gram reference distribution for accurate probabilistic HPSG parsing. In the model, the n-gram reference distribution is simply defined as the product of the probabilities of selecting lexical entries, which are provided by the discriminative method with machine learning features of word and POS n-gram as defined in the CCG/HPSG/CDG supertagging. Recently, supertagging becomes well known to drastically improve the parsing accuracy and speed, but supertagging techniques were heuristically introduced, and hence the probabilistic models for parse trees were not well defined. We introduce the supertagging probabilities as a reference distribution for the log-linear model of the probabilistic HPSG. This is the first model which properly incorporates the supertagging probabilities into parse tree's probabilistic model.

1 Introduction

For the last decade, fast, accurate and wide-coverage parsing for real-world text has been pursued in sophisticated grammar formalisms, such as head-driven phrase structure grammar (HPSG) (Pollard and Sag, 1994), combinatory categorial grammar (CCG) (Steedman, 2000) and lexical function grammar (LFG) (Bresnan, 1982). They are preferred because they give precise and in-depth analyses for explaining linguistic phenomena, such as passivization, control verbs and relative clauses. The main difficulty of developing parsers in these formalisms was how to model a well-defined probabilistic model for graph structures such as feature structures. This was overcome by a probabilistic model which provides probabilities of discriminating a correct parse tree among candidates of parse trees in a *log-linear model* or *maximum entropy model* (Berger et al., 1996) with many features for parse trees (Abney, 1997; Johnson et al., 1999; Riezler et al., 2000; Malouf and van Noord, 2004; Kaplan et al., 2004; Miyao and Tsujii, 2005). Following this discriminative approach, techniques for efficiency were investigated for estimation (Geman and Johnson, 2002; Miyao and Tsujii, 2002; Malouf and van Noord, 2004) and parsing (Clark and Curran, 2004b; Clark and Curran, 2004a; Ninomiya et al., 2005).

An interesting approach to the problem of parsing efficiency was using supertagging (Clark and Cur-

Proceedings of the 10th Conference on Parsing Technologies, pages 60–68,
Prague, Czech Republic, June 2007.

ran, 2004b; Clark and Curran, 2004a; Wang, 2003; Wang and Harper, 2004; Nasr and Rambow, 2004; Ninomiya et al., 2006; Foth et al., 2006; Foth and Menzel, 2006), 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. 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 by showing accuracy as high as the state-of-the-art parsers, and Foth et al. (2006) and Foth and Menzel (2006) reported that accuracy was significantly improved by incorporating the supertagging probabilities into manually tuned Weighted CDG. Ninomiya et al. (2006) showed the parsing model using only supertagging probabilities could achieve accuracy as high as the probabilistic model for phrase structures. This means that syntactic structures are almost determined by supertags as is claimed by Bangalore and Joshi (1999). However, supertaggers themselves were heuristically used as an external tagger. They filter out unlikely lexical entries just to help parsing (Clark and Curran, 2004a), or the probabilistic models for phrase structures were trained independently of the supertagger's probabilistic models (Wang and Harper, 2004; Ninomiya et al., 2006). In the case of supertagging of Weighted CDG (Foth et al., 2006), parameters for Weighted CDG are manually tuned, i.e., their model is not a well-defined probabilistic model.

We propose a log-linear model for probabilistic HPSG parsing in which the supertagging probabilities are introduced as a reference distribution for the probabilistic HPSG. The reference distribution is simply defined as the product of the probabilities of selecting lexical entries, which are provided by the discriminative method with machine learning features of word and part-of-speech (POS) n-gram as defined in the CCG/HPSG/CDG supertagging. This is the first model which properly incorporates the supertagging probabilities into parse tree's probabilistic model. We compared our model with the probabilistic model for phrase structures (Miyao and Tsujii, 2005). This model uses word and POS unigram for its reference distribution, i.e., the probabilities of unigram supertagging. Our model can be regarded as an extension of a unigram reference distribution to an n-gram reference distribution with features that are used in supertagging. We also compared with a probabilistic model in (Ninomiya et al., 2006). The probabilities of their model are defined as the product of probabilities of supertagging and probabilities of the probabilistic model for phrase structures, but their model was trained independently of supertagging probabilities, i.e., the supertagging probabilities are not used for reference distributions.

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

$L = \{l = \langle w, F \rangle | w \in \mathcal{W}, F \in \mathcal{F}\}$ is a set of lexical entries, and

R is a set of schemata; i.e., $r \in R$ is a partial function: $\mathcal{F} \times \mathcal{F} \to \mathcal{F}$.

Given a sentence, an HPSG computes a set of phrasal signs, i.e., feature structures, as a result of

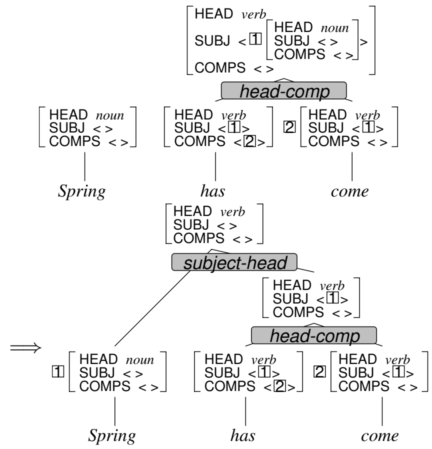

Figure 1: HPSG parsing.

parsing. Note that HPSG is one of the lexicalized grammar formalisms, in which lexical entries determine the dominant syntactic structures.

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

(Probabilistic HPSG)

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

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

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* (Jelinek, 1998; Johnson and Riezler, 2000) of the probabilistic HPSG model; i.e., the computation of parse trees given low probabilities by the model is omitted in the estimation stage (Miyao and Tsujii, 2005), or a probabilistic model can be augmented by several distributions estimated from the larger and simpler corpus (Johnson and Riezler, 2000). In (Miyao and Tsujii, 2005), $p_0(T|\mathbf{w})$ is defined as the product of probabilities of selecting lexical entries with word and POS unigram features:

(Miyao and Tsujii (2005)'s model)

$$p_{uniref}(T|\mathbf{w}) = p_0(T|\mathbf{w}) \frac{1}{Z_{\mathbf{w}}} \exp\left(\sum_u \lambda_u f_u(T)\right)$$

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

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

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 other two types of probabilistic models using a supertagger (Ninomiya et al., 2006). 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 probabilistic models explained above. 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:

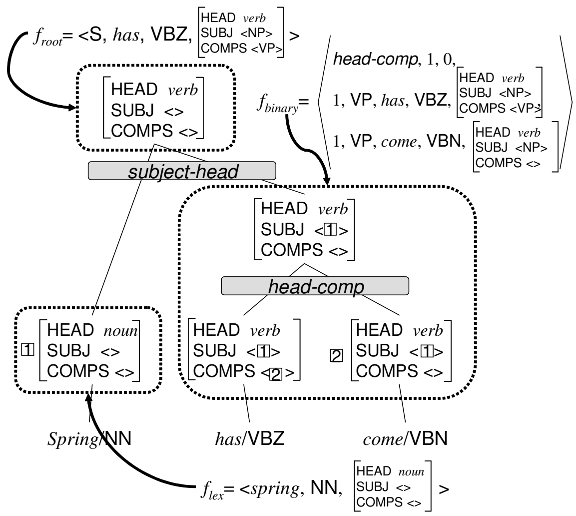

Figure 2: Example of features.

(Ninomiya et al. (2006)'s 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 probabilities of lexical entry selection, $p(l_i|\mathbf{w}, i)$, are defined as follows:

(Probabilistic model of lexical entry selection)

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

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

where Z_w is the sum over all possible lexical entries for the word w_i.

The second model is a hybrid model of supertagging and the probabilistic HPSG. The probabilities are given as the product of Ninomiya et al. (2006)'s model 1 and the probabilistic HPSG.

(Ninomiya et al. (2006)'s model 3)

$$p_{model3}(T|\mathbf{w}) = p_{model1}(T|\mathbf{w}) p_{hpsg}(T|\mathbf{w})$$

In the experiments, we compared our model with Miyao and Tsujii (2005)'s model and Ninomiya et al. (2006)'s model 1 and 3. The features used in our model and their model are combinations of the feature templates listed in Table 1 and Table 2. 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. f_{lex} is a feature template set for calculating the unigram reference distribution and is used in Miyao and Tsujii (2005)'s model. f_{sptag} is a feature template set for calculating the probabilities of selecting lexical entries in Ninomiya et al. (2006)'s model 1 and 3. The feature templates in f_{sptag} are word trigrams and POS 5-grams. An example of features applied to the parse tree for the sentence "*Spring has come*" is shown in Figure 2.

$$
\begin{aligned}
f_{binary} &= \left\langle \begin{array}{l} r, d, c, \\ sp_l, sy_l, hw_l, hp_l, hl_l, \\ sp_r, sy_r, hw_r, hp_r, hl_r \end{array} \right\rangle \\
f_{unary} &= \langle r, sy, hw, hp, hl \rangle \\
f_{root} &= \langle sy, hw, hp, hl \rangle \\
f_{lex} &= \langle w_i, p_i, l_i \rangle \\
f_{sptag} &= \left\langle \begin{array}{l} w_{i-1}, w_i, w_{i+1}, \\ p_{i-2}, p_{i-1}, p_i, p_{i+1}, p_{i+2} \end{array} \right\rangle
\end{aligned}
$$

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: Feature templates.

3 Probabilistic HPSG with an n-gram reference distribution

In this section, we propose a probabilistic model with an n-gram reference distribution for probabilistic HPSG parsing. This is an extension of Miyao and Tsujii (2005)'s model by replacing the unigram reference distribution with an n-gram reference distribution. Our model is formally defined as follows:

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

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

Table 2: Combinations of feature templates.

(Probabilistic HPSG with an n-gram reference distribution)

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

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

In our model, Ninomiya et al. (2006)'s model 1 is used as a reference distribution. The probabilistic model of lexical entry selection and its feature templates are the same as defined in Ninomiya et al. (2006)'s model 1.

The formula of our model is the same as Ninomiya et al. (2006)'s model 3. But, their model is not a probabilistic model with a reference distribution. Both our model and their model consist of the probabilities for lexical entries (= $p_{model1}(T|\mathbf{w})$) and the probabilities for phrase structures (= the rest of each formula). The only difference between our model and their model is the way of how to train model parameters for phrase structures. In both our model and their model, the parameters for lexical entries (= the parameters of $p_{model1}(T|\mathbf{w})$) are first estimated from the word and POS sequences independently of the parameters for phrase structures. That is, the estimated parameters for lexical entries are the same in both models, and hence the probabilities of $p_{model1}(T|\mathbf{w})$ of both models are the same. Note that the parameters for lexical entries will never be updated after this estimation stage; i.e., the parameters for lexical entries are not estimated in the same time with the parameters for phrase structures. The difference of our model and their model is the estimation of parameters for phrase structures. In our model, given the probabilities for lexical entries, the parameters for phrase structures are estimated so as to maximize the entire probabilistic model (= the product of the probabilities for lexical entries and the probabilities for phrase structures) in the training corpus. In their model, the parameters for phrase structures are trained without using the probabilities for lexical entries, i.e., the parameters for phrase structures are estimated so as to maximize the probabilities for phrase structures only. That is, the parameters for lexical entries and the parameters for phrase structures are trained independently in their model.

Miyao and Tsujii (2005)'s model also uses a reference distribution, but with word and POS unigram features, as is explained in the previous section. The only difference between our model and Miyao and Tsujii (2005)'s model is that our model uses sequences of word and POS tags as n-gram features for selecting lexical entries in the same way as supertagging does.

4 Experiments

We evaluated the speed and accuracy of parsing 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 prob-

[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.

	No. of tested sentences	Total No. of sentences	Avg. length of tested sentences
Section 23	2,299 (100.00%)	2,299	22.2
Section 24	1,245 (99.84%)	1,247	23.0

Table 3: Statistics of the Penn Treebank.

	Section 23 (Gold POSs)						
	LP (%)	LR (%)	LF (%)	UP (%)	UR (%)	UF (%)	Avg. time (ms)
Miyao and Tsujii (2005)'s model	87.26	86.50	86.88	90.73	89.93	90.33	604
Ninomiya et al. (2006)'s model 1	87.23	86.47	86.85	90.05	89.27	89.66	129
Ninomiya et al. (2006)'s model 3	89.48	88.58	89.02	92.33	91.40	91.86	152
our model 1	89.78	89.28	89.53	92.58	92.07	92.32	234
our model 2	90.03	89.60	89.82	92.82	92.37	92.60	1379
	Section 23 (POS tagger)						
	LP (%)	LR (%)	LF (%)	UP (%)	UR (%)	UF (%)	Avg. time (ms)
Miyao and Tsujii (2005)'s model	84.96	84.25	84.60	89.55	88.80	89.17	674
Ninomiya et al. (2006)'s model 1	85.00	84.01	84.50	88.85	87.82	88.33	154
Ninomiya et al. (2006)'s model 3	87.35	86.29	86.82	91.24	90.13	90.68	183
Matsuzaki et al. (2007)'s model	86.93	86.47	86.70	-	-	-	30
our model 1	87.28	87.05	87.17	91.62	91.38	91.50	260
our model 2	87.56	87.46	87.51	91.88	91.77	91.82	1821

Table 4: Experimental results for Section 23.

abilistic 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 quick check (Malouf et al., 2000).

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[2]. 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 ≤ 100 words in Section 23. The performance of each model was analyzed using the sentences in Section 24 of ≤ 100 words. Table 3 details the numbers and average lengths of the tested sentences of ≤ 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). LF and UF in the figure are labeled F-score and unlabeled F-score. F-score is the harmonic mean of precision and recall. We evaluated our model in two settings. One is implemented with a narrow beam width ('our model 1' in the figure), and the other is implemented with a wider beam width ('our model 2' in the figure)[3]. 'our model

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.

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

[3] The beam thresholding parameters for 'our model 1' are $\alpha_0 = 10, \Delta\alpha = 5, \alpha_{\text{last}} = 30, \beta_0 = 5.0, \Delta\beta = 2.5, \beta_{\text{last}} = 15.0, \delta_0 = 10, \Delta\delta = 5, \delta_{\text{last}} = 30$, $\kappa_0 = 5.0, \Delta\kappa = 2.5, \kappa_{\text{last}} = 15.0$, $\theta_0 = 6.0, \Delta\theta = 3.5$, and $\theta_{\text{last}} = 20.0$.

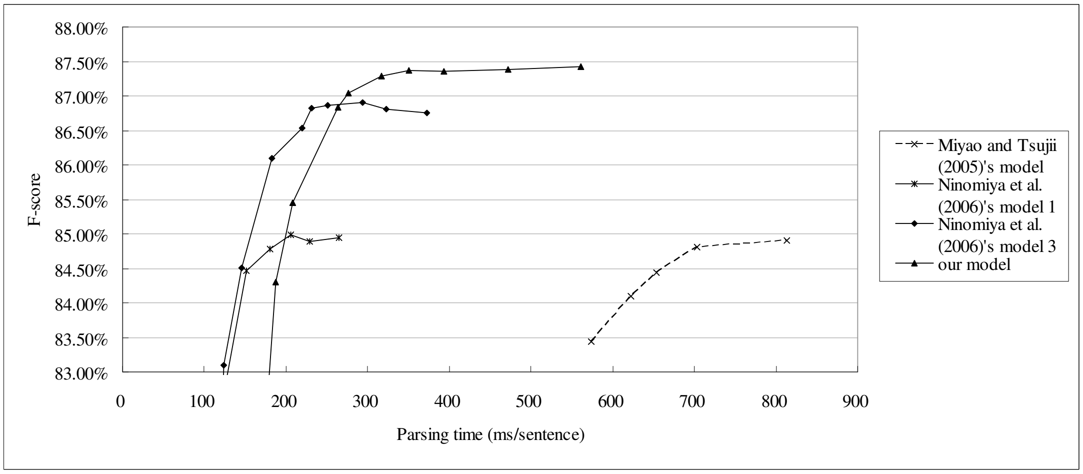

Figure 3: F-score versus average parsing time for sentences in Section 24 of $\leq$ 100 words.

1' was introduced to measure the performance with balanced F-score and speed, which we think appropriate for practical use. 'our model 2' was introduced to measure how high the precision and recall could reach by sacrificing speed. Our models increased the parsing accuracy. 'our model 1' was around 2.6 times faster and had around 2.65 points higher F-score than Miyao and Tsujii (2005)'s model. 'our model 2' was around 2.3 times slower but had around 2.9 points higher F-score than Miyao and Tsujii (2005)'s model. We must admit that the difference between our models and Ninomiya et al. (2006)'s model 3 was not as great as the difference from Miyao and Tsujii (2005)'s model, but 'our model 1' achieved 0.56 points higher F-score, and 'our model 2' achieved 0.8 points higher F-score. When the automatic POS tagger was introduced, F-score dropped by around 2.4 points for all models.

We also compared our model with Matsuzaki et al. (2007)'s model. Matsuzaki et al. (2007) proposed a technique for efficient HPSG parsing with supertagging and CFG filtering. Their results with the same grammar and servers are also listed in the lower half of Table 4. They achieved drastic improvement in efficiency. Their parser ran around 6 times faster than Ninomiya et al. (2006)'s model 3, 9 times faster than 'our model 1' and 60 times faster than 'our model 2.' Instead, our models achieved better accuracy. 'our model 1' had around 0.5 higher F-score, and 'our model 2' had around 0.8 points higher F-score. Their efficiency is mainly due to elimination of ungrammatical lexical entries by the CFG filtering. They first parse a sentence with a CFG grammar compiled from an HPSG grammar, and then eliminate lexical entries that are not in the parsed CFG trees. Obviously, this technique can also be applied to the HPSG parsing of our models. We think that efficiency of HPSG parsing with our models will be drastically improved by applying this technique.

The average parsing time and labeled F-score curves of each probabilistic model for the sentences in Section 24 of $\leq$ 100 words are graphed in Figure 3. The graph clearly shows the difference of our model and other models. As seen in the graph, our model achieved higher F-score than other model when beam threshold was widen. This implies that other models were probably difficult to reach the F-score of 'our model 1' and 'our model 2' for Section 23 even if we changed the beam thresholding parameters. However, F-score of our model dropped eas-

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). The terms with suffixes 0 are the initial values. The parser iterates parsing until it succeeds to generate a parse tree. The parameters increase for each iteration by the terms prefixed by Δ, and parsing finishes when the parameters reach the terms with suffixes last. Details of the parameters are written in (Ninomiya et al., 2005). The beam thresholding parameters for 'our model 2' are $\alpha_0 = 18, \Delta\alpha = 6, \alpha_{\text{last}} = 42, \beta_0 = 9.0, \Delta\beta = 3.0, \beta_{\text{last}} = 21.0, \delta_0 = 18, \Delta\delta = 6, \delta_{\text{last}} = 42, \kappa_0 = 9.0, \Delta\kappa = 3.0, \kappa_{\text{last}} = 21.0$. In 'our model 2', the global thresholding was not used.

ily when we narrow down the beam threshold, compared to other models. We think that this is mainly due to its bad implementation of parser interface. The n-gram reference distribution is incorporated into the kernel of the parser, but the n-gram features and a maximum entropy estimator are defined in other modules; n-gram features are defined in a grammar module, and a maximum entropy estimator for the n-gram reference distribution is implemented with a general-purpose maximum entropy estimator module. Consequently, strings that represent the n-gram information are very frequently changed into feature structures and vice versa when they go in and out of the kernel of the parser. On the other hand, Ninomiya et al. (2006)'s model 3 uses the supertagger as an external module. Once the parser acquires the supertagger's outputs, the n-gram information never goes in and out of the kernel. This advantage of Ninomiya et al. (2006)'s model can apparently be implemented in our model, but this requires many parts of rewriting of the implemented parser. We estimate that the overhead of the interface is around from 50 to 80 ms/sentence. We think that re-implementation of the parser will improve the parsing speed as estimated. In Figure 3, the line of our model crosses the line of Ninomiya et al. (2006)'s model. If the estimation is correct, our model will be faster and more accurate so that the lines in the figure do not cross. Speed-up in our model is left as a future work.

5 Conclusion

We proposed a probabilistic model in which supertagging is consistently integrated into the probabilistic model for HPSG. In the model, the n-gram reference distribution is simply defined as the product of the probabilities of selecting lexical entries with machine learning features of word and POS n-gram as defined in the CCG/HPSG/CDG supertagging. We conducted experiments on the Penn Treebank with a wide-coverage HPSG parser. In the experiments, we compared our model with the probabilistic HPSG with a unigram reference distribution (Miyao and Tsujii, 2005) and the probabilistic HPSG with supertagging (Ninomiya et al., 2006). Though our model was not as fast as Ninomiya et al. (2006)'s models, it achieved the highest accuracy among them. Our model had around 2.65 points higher F-score than Miyao and Tsujii (2005)'s model and around 0.56 points higher F-score than the Ninomiya et al. (2006)'s model 3. When we sacrifice parsing speed, our model achieved around 2.9 points higher F-score than Miyao and Tsujii (2005)'s model and around 0.8 points higher F-score than Ninomiya et al. (2006)'s model 3. Our model achieved higher F-score because parameters for phrase structures in our model are trained with the supertagging probabilities, which are not in other models.

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.

Joan Bresnan. 1982. *The Mental Representation of Grammatical Relations*. MIT Press, Cambridge, MA.

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.

Killian Foth and Wolfgang Menzel. 2006. Hybrid parsing: Using probabilistic models as predictors for a symbolic parser. In *Proc. of COLING-ACL 2006*.

Killian Foth, Tomas By, and Wolfgang Menzel. 2006. Guiding a constraint dependency parser with supertags. In *Proc. of COLING-ACL 2006*.

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.

F. Jelinek. 1998. *Statistical Methods for Speech Recognition*. The MIT Press.

Mark Johnson and Stefan Riezler. 2000. Exploiting auxiliary distributions in stochastic unification-based grammars. In *Proc. of NAACL-2000*, pages 154–161.

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*.

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.

Takuya Matsuzaki, Yusuke Miyao, and Jun'ichi Tsujii. 2007. Efficient HPSG parsing with supertagging and CFG-filtering. In *Proc. of IJCAI 2007*, pages 1671–1676.

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.

Takashi Ninomiya, Takuya Matsuzaki, Yoshimasa Tsuruoka, Yusuke Miyao, and Jun'ichi Tsujii. 2006. Extremely lexicalized models for accurate and fast HPSG parsing. In *Proc. of EMNLP 2006*, pages 155–163.

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.

Mark Steedman. 2000. *The Syntactic Process*. The MIT Press.

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.