Evaluating Impact of Re-training a Lexical Disambiguation Model on Domain Adaptation of an HPSG Parser

Tadayoshi Hara[1] **Yusuke Miyao**[1] **Jun'ichi Tsujii**[1,2,3]
[1]Department of Computer Science, University of Tokyo
Hongo 7-3-1, Bunkyo-ku, Tokyo 113-0033 Japan
[2]School of Computer Science, University of Manchester
POBox 88, Sackville St, MANCHESTER M60 1QD, UK
[3]NaCTeM(National Center for Text Mining)
Manchester Interdisciplinary Biocentre, University of Manchester
131 Princess St, MANCHESTER M1 7DN, UK
E-mail: {harasan, yusuke, tsujii}@is.s.u-tokyo.ac.jp

Abstract

This paper describes an effective approach to adapting an HPSG parser trained on the Penn Treebank to a biomedical domain. In this approach, we train probabilities of lexical entry assignments to words in a target domain and then incorporate them into the original parser. Experimental results show that this method can obtain higher parsing accuracy than previous work on domain adaptation for parsing the same data. Moreover, the results show that the combination of the proposed method and the existing method achieves parsing accuracy that is as high as that of an HPSG parser retrained from scratch, but with much lower training cost. We also evaluated our method in the Brown corpus to show the portability of our approach in another domain.

1 Introduction

Domain portability is an important aspect of the applicability of NLP tools to practical tasks. Therefore, domain adaptation methods have recently been proposed in several NLP areas, e.g., word sense disambiguation (Chan and Ng, 2006), statistical parsing (Lease and Charniak, 2005; McClosky et al., 2006), and lexicalized-grammar parsing (Johnson and Riezler, 2000; Hara et al., 2005). Their aim was to re-train a probabilistic model for a new domain at low cost, and more or less successfully improved the accuracy for the domain.

In this paper, we propose a method for adapting an HPSG parser (Miyao and Tsujii, 2002; Ninomiya et al., 2006) trained on the WSJ section of the Penn Treebank (Marcus et al., 1994) to a biomedical domain. Our method re-trains a probabilistic model of lexical entry assignments to words in a target domain, and incorporates it into the original parser. The model of lexical entry assignments is a log-linear model re-trained with machine learning features only of word n-grams. Hence, the cost for the re-training is much lower than the cost of training the entire disambiguation model from scratch.

In the experiments, we used an HPSG parser originally trained with the Penn Treebank, and evaluated a disambiguation model re-trained with the GENIA treebank (Kim et al., 2003), which consists of abstracts of biomedical papers. We varied the size of a training corpus, and measured the transition of the parsing accuracy and the cost required for parameter estimation. For comparison, we also examined other possible approaches to adapting the same parser. In addition, we applied our approach to the Brown corpus (Kucera and Francis, 1967) in order to examine portability of our approach.

The experimental results revealed that by simply re-training the probabilistic model of lexical entry assignments we achieve higher parsing accuracy than with a previously proposed adaptation method. In addition, combined with the existing adaptation method, our approach achieves accuracy as high as that obtained by re-training the original parser from scratch, but with much lower training cost. In this paper, we report these experimental results in detail, and discuss how disambiguation models of lexical entry assignments contribute to domain adaptation.

In recent years, it has been shown that lexical in-

Proceedings of the 10th Conference on Parsing Technologies, pages 11–22,
Prague, Czech Republic, June 2007.

formation plays a very important role for high accuracy of lexicalized grammar parsing. Bangalore and Joshi (1999) indicated that, correct disambiguation with supertagging, i.e., assignment of lexical entries before parsing, enabled effective LTAG (Lexicalized Tree-Adjoining Grammar) parsing. Clark and Curran (2004a) showed that supertagging reduced cost for training and execution of a CCG (Combinatory Categorial Grammar) parser while keeping accuracy. Clark and Curran (2006) showed that a CCG parser trained on data derived from lexical category sequences alone was only slightly less accurate than one trained on complete dependency structures. Ninomiya et al. (2006) also succeeded in significantly improving speed and accuracy of HPSG parsing by using supertagging probabilities. These results indicate that the probability of lexical entry assignments is essential for parse disambiguation.

Such usefulness of lexical information has also been shown for domain adaptation methods. Lease and Charniak (2005) showed how existing domain-specific lexical resources on a target domain may be leveraged to augment PTB-training: part-of-speech tags, dictionary collocations, and named-entities. Our findings basically follow the above results. The contribution of this paper is to provide empirical results of the relationships among domain variation, probability of lexical entry assignment, training data size, and training cost. In particular, this paper empirically shows how much in-domain corpus is required for satisfiable performance.

In Section 2, we introduce an HPSG parser and describe an existing method for domain adaptation. In Section 3, we show our methods of re-training a lexical disambiguation model and incorporating it into the original model. In Section 4, we examine our method through experiments on the GENIA treebank. In Section 5, we examine the portability of our method through experiments on the Brown corpus. In Section 6, we showed several recent researches related to domain adaptation.

2 An HPSG Parser

HPSG (Pollard and Sag, 1994) is a syntactic theory based on lexicalized grammar formalism. In HPSG, a small number of grammar rules describe general construction rules, and a large number of lexical entries express word-specific characteristics. The structures of sentences are explained using combinations of grammar rules and lexical entries.

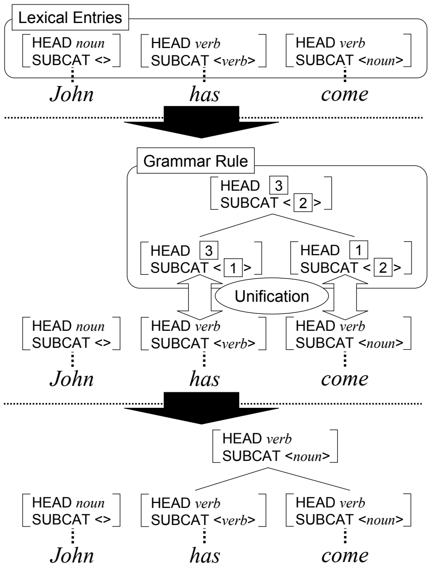

Figure 1: Parsing a sentence “*John has come.*”

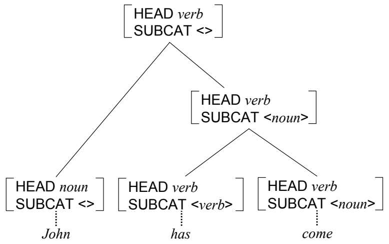

Figure 2: An HPSG parse tree for a sentence “*John has come.*”

Figure 1 shows an example of HPSG parsing of the sentence “*John has come.*” First, as shown at the top of the figure, an HPSG parser assigns a lexical entry to each word in this sentence. Next, a grammar rule is assigned and applied to lexical entries. At the middle of this figure, the grammar rule is applied to the lexical entries for “*has*” and “*come.*” We then obtain the structure represented at the bottom of the figure. After that, the application of grammar rules is done iteratively, and then we can finally obtain the parse tree as is shown in Figure 2. In practice, since two or more parse candidates can be given for one sentence, a disambiguation model gives probabilities to these candidates, and a candidate given the highest probability is then chosen as a correct parse.

The HPSG parser used in this study is Ninomiya et al. (2006), which is based on *Enju* (Miyao and Tsujii, 2005). Lexical entries of Enju were extracted from the Penn Treebank (Marcus et al., 1994), which consists of sentences collected from The Wall Street Journal (Miyao et al., 2004). The disambiguation model of Enju was trained on the same treebank.

The disambiguation model of Enju is based on a feature forest model (Miyao and Tsujii, 2002), which is a log-linear model (Berger et al., 1996) on packed forest structure. The probability, $p_E(t|\mathbf{w})$, of producing the parse result t for a given sentence $\mathbf{w} = \langle w_1, ..., w_u \rangle$ is defined as

$$p_E(t|\mathbf{w}) = \frac{1}{Z_s} \prod_i p_{lex}(l_i|\mathbf{w}, i) \cdot q_{syn}(t|\mathbf{l}),$$

$$Z_s = \sum_{t \in T(\mathbf{w})} \prod_i p_{lex}(l_i|\mathbf{w}, i) \cdot q_{syn}(t|\mathbf{l})$$

where $\mathbf{l} = \langle l_1, ..., l_u \rangle$ is a list of lexical entries assigned to $\mathbf{w}$, $p_{lex}(l_i|\mathbf{w}, i)$ is a probabilistic model giving the probability that lexical entry l_i is assigned to word w_i, $q_{syn}(t|\mathbf{l})$ is an unnormalized log-linear model of tree construction and gives the possibility that parse candidate t is produced from lexical entries $\mathbf{l}$, and $T(\mathbf{w})$ is a set of parse candidates assigned to $\mathbf{w}$. With a treebank of a target domain as training data, model parameters of p_{lex} and q_{syn} are estimated so as to maximize the log-likelihood of the training data.

Probabilistic model p_{lex} is defined as a log-linear model as follows.

$$p_{lex}(l_i|\mathbf{w}, i) = \frac{1}{Z_{w_i}} \exp\left(\sum_j \lambda_j f_j(l_i, \mathbf{w}, i)\right),$$

$$Z_{w_i} = \sum_{l_i \in L(w_i)} \exp\left(\sum_j \lambda_j f_j(l_i, \mathbf{w}, i)\right),$$

where $L(w_i)$ is a set of lexical entries which can be assigned to word w_i. Before training this model, $L(w_i)$ for all w_i are extracted from the training treebank. The feature function $f_j(l_i, \mathbf{w}, i)$ represents the characteristics of l_i, $\mathbf{w}$ and w_i, while corresponding λ_j is its weight. For the feature functions, instead of using unigram features adopted in Miyao and Tsujii (2005), Ninomiya et al. (2006) used "word trigram" and "POS 5-gram" features which are listed in Table 1. With the revised Enju model, they achieved parsing accuracy as high as Miyao and Tsujii (2005), with around four times faster parsing speed.

Table 1: Features for the probabilities of lexical entry selection

surrounding words	$w_{-1}w_0w_1$ (word trigram)
surrounding POS tags	$p_{-2}p_{-1}p_0p_1p_2$ (POS 5-gram)
combinations	$w_{-1}w_0, w_0w_1, p_{-1}w_0, p_0w_0,$ $p_1w_0, p_0p_1p_2p_3, p_{-2}p_{-1}p_0,$ $p_{-1}p_0p_1, p_0p_1p_2, p_{-2}p_{-1},$ $p_{-1}p_0, p_0p_1, p_1p_2$

Johnson and Riezler (2000) suggested the possibility of the method for adapting a stochastic unification-based grammar including HPSG to another domain. They incorporated auxiliary distributions as additional features for an original log-linear model, and then attempted to assign proper weights to the new features. With this approach, they succeeded in decreasing to a degree indistinguishable sentences for a target grammar.

Our previous work proposed a method for adapting an HPSG parser trained on the Penn Treebank to a biomedical domain (Hara et al., 2005). We re-trained a disambiguation model of tree construction, i.e., q_{syn}, for the target domain. In this approach, q_{syn} of the original parser was used as a *reference distribution* (Jelinek, 1998) of another log-linear model, and the new model was trained using a target treebank. Since re-training used only a small treebank of the target domain, the cost was small and parsing accuracy was successfully improved.

3 Re-training of a Disambiguation Model of Lexical Entry Assignments

Our idea of domain adaptation is to train a disambiguation model of lexical entry assignments for the target domain and then incorporate it into the original parser. Since Enju includes the disambiguation model of lexical entry assignments as p_{lex}, we can implement our method in Enju by training another disambiguation model $p'_{lex}(l_i|\mathbf{w}, i)$ of lexical entry assignments for the biomedical domain, and then replacing the original p_{lex} with the newly trained p'_{lex}.

In this paper, for p'_{lex}, we train a disambiguation model $p_{lex-mix}(l_i|\mathbf{w}, i)$ of lexical entry assignments. $p_{lex-mix}$ is a maximum entropy model and the feature functions for it is the same as p_{lex} as given in Table 1. With these feature functions, we train $p_{lex-mix}$ on the treebanks both of the original and biomedical domains.

In the experiments, we examine the contribution of our method to parsing accuracy. In addition, we implement several other possible methods for comparison of the performances.

baseline: use the original model of Enju

GENIA only: execute the same method of training the disambiguation model of Enju, using only the GENIA treebank

Mixture: execute the same method of training the disambiguation model of Enju, using both of the Penn Treebank and the GENIA treebank (a kind of smoothing method)

HMT05: execute the method proposed in our previous work (Hara et al., 2005)

Our method: replace p_{lex} in the original model with $p_{lex-mix}$, while leaving q_{syn} as it is

Our method (GENIA): replace p_{lex} in the original model with $p_{lex-genia}$, which is a probabilistic model of lexical entry assignments trained only with the GENIA treebank, while leaving q_{syn} as it is

Our method + GENIA: replace p_{lex} in the original model with $p_{lex-mix}$ and q_{syn} with $q_{syn-genia}$, which is a disambiguation model of tree construction trained with the GENIA treebank

Our method + HMT05: replace p_{lex} in the original model with $p_{lex-mix}$ and q_{syn} with the model re-trained with our previous method (Hara et al., 2005) (the combination of our method and the "HMT05" method)

baseline (lex): use only p_{lex} as a disambiguation model

GENIA only (lex): use only $p_{lex-genia}$ as a disambiguation model, which is a probabilistic model of lexical entry assignments trained only with the GENIA treebank

Mixture (lex): use only $p_{lex-mix}$ as a disambiguation model

The "baseline" method does no adaptation to the biomedical domain, and therefore gives lower parsing accuracy for the domain than for the original domain. This method is regarded as the baseline of the experiments. The "GENIA only" method relies solely on the treebank for the biomedical domain, and therefore it cannot work well with the small treebank. The "Mixture" method is a kind of smoothing method using all available training data at the same time, and therefore the method can give the highest accuracy of the three, which would be regarded as the ideal accuracy with the naive methods. However, training this model is expected to be very costly.

The "baseline (lex)," "GENIA only (lex)," and "Mixture (lex)" approaches rely solely on models of lexical entry assignments, and show lower accuracy than those that contain both of models of lexical entry assignments and tree constructions. These approaches can be utilized as indicators of importance of combining the two types of models.

Our previous work (Hara et al., 2005) showed that the model trained with the "HMT05" method can give higher accuracy than the "baseline" method, even with the small amount of the treebanks in the biomedical domain. The model also takes much less cost to train than with the "Mixture" method. However, they reported that the method could not give as high accuracy as the "Mixture" method.

4 Experiments with the GENIA Corpus

4.1 Experimental Settings

We implemented the models shown in Section 3, and then evaluated the performance of them. The original parser, Enju, was developed on Section 02-21 of the Penn Treebank (39,832 sentences) (Miyao and Tsujii, 2005; Ninomiya et al., 2006). For training those models, we used the GENIA treebank (Kim et al., 2003), which consisted of 1,200 abstracts (10,848 sentences) extracted from MEDLINE. We divided it into three sets of 900, 150, and 150 abstracts (8,127, 1,361, and 1,360 sentences), and these sets were used respectively as training, development, and final evaluation data. The method of Gaussian MAP estimation (Chen and Rosenfeld, 1999) was used for smoothing. The meta parameter σ of the Gaussian distribution was determined so as to maximize the accuracy on the development set.

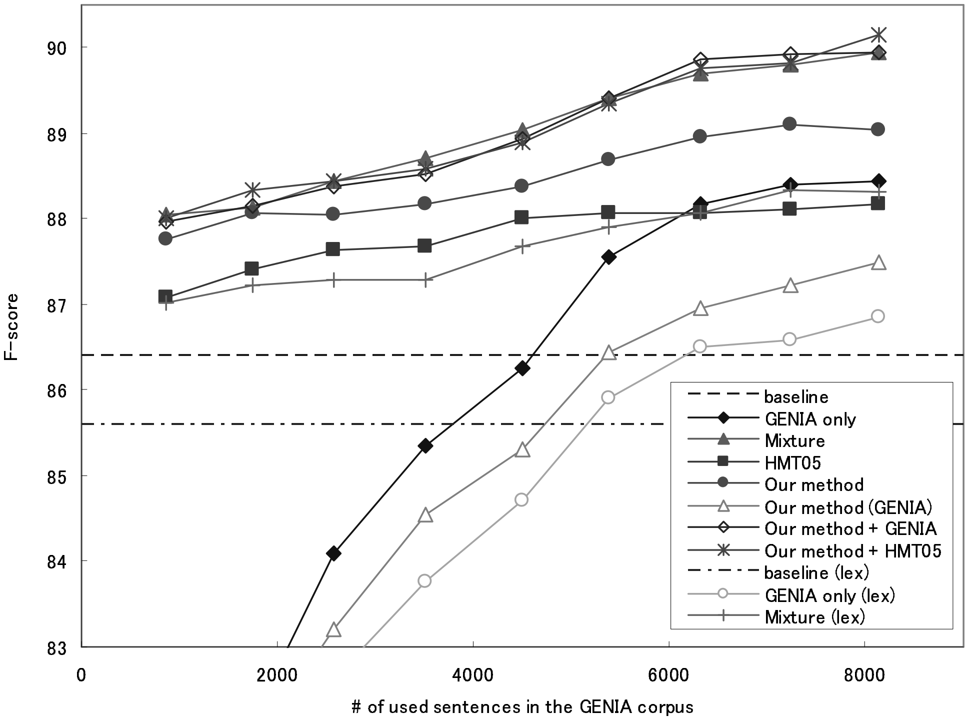

Figure 3: Corpus size vs. accuracy for various methods

In the following experiments, we measured the accuracy of predicate-argument dependencies on the evaluation set. The measure is labeled precision/recall (LP/LR), which is the same measure as previous work (Clark and Curran, 2004b; Miyao and Tsujii, 2005) that evaluated the accuracy of lexicalized grammars on the Penn Treebank.

The features for the examined approaches were all the same as the original disambiguation model. In our previous work, the features for "HMT05" were tuned to some extent. We evened out the features in order to compare various approaches under the same condition. The lexical entries for training each model were extracted from the treebank used for training the model of lexical entry assignments.

We compared the performances of the given models from various angles, by focusing mainly on the accuracy against the cost. For each of the models, we measured the accuracy transition according to the size of the GENIA treebank for training and according to the training time. We changed the size of the GENIA treebank for training: 100, 200, 300, 400, 500, 600, 700, 800, and 900 abstracts. Figure 3 and 4 show the F-score transition according to the size of the training set and the training time among the given models respectively. Table 2 and Table 3 show the parsing performance and the training cost obtained when using 900 abstracts of the GENIA treebank. Note that Figure 4 does not include the results of the "Mixture" method because only the method took too much training cost as shown in Table 3. It should also be noted that training time in Figure 4 includes time required for both training and development tests. In Table 2, accuracies with models other than "baseline" showed the significant differences from "baseline" according to stratified shuffling test (Cohen, 1995) with p-value < 0.05.

In the rest of this section we analyze these experimental results by focusing mainly on the contribution of re-training lexical entry assignment models. We first observe the results with the naive or existing approaches. On the basis of these results, we evaluate the impact of our method. We then explore the combination of our method with other methods, and analyze the errors for our future research.

4.2 Exploring Naive or Existing Approaches

Without adaptation, Enju gave the parsing accuracy of 86.39 in F-score, which was 3.42 point lower than

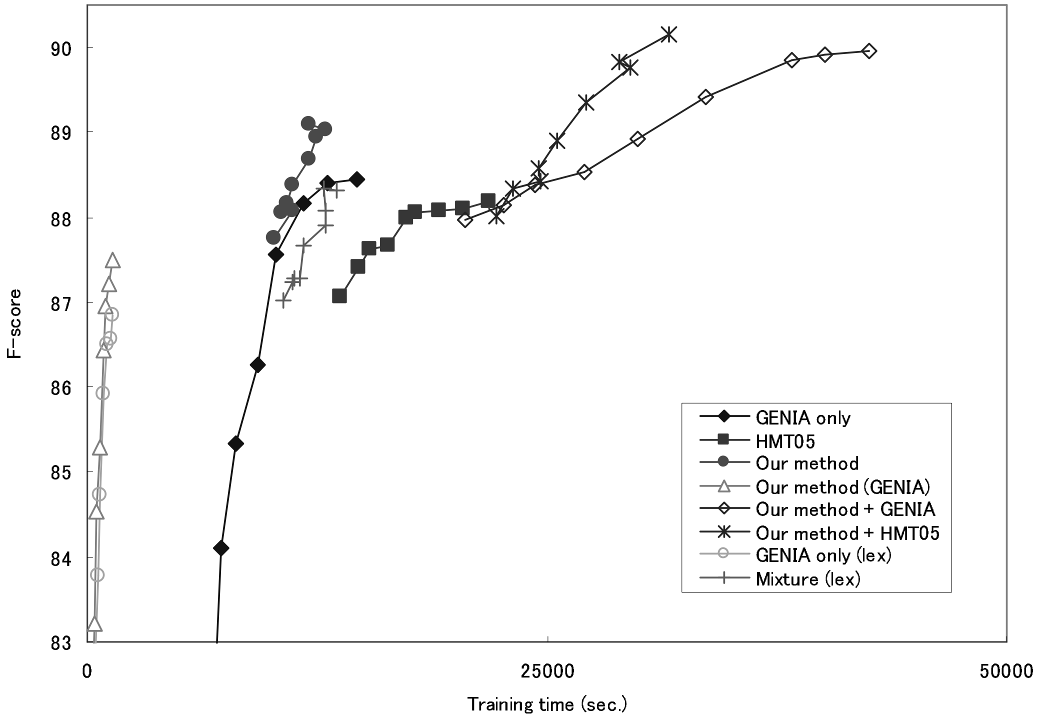

Figure 4: Training time vs. accuracy for various methods

that Enju gave for the original domain, the Penn Treebank. This is the baseline of the experiments.

Figure 3 shows that, for less than about 4,500 training sentences, the “GENIA only” method could not obtain as high parsing accuracy as the “baseline” method. This result would indicate that the training data would not be sufficient for re-training the whole disambiguation model from scratch. However, if we prepared more than about 4,500 sentences, the method could give higher accuracy than “baseline” with low training cost (see Figure 4). On the other hand, the “Mixture” method could obtain the highest level of the parsing accuracy for any size of the GENIA treebank. However, Table 3 shows that this method required too much training cost. It would be a major barrier for further challenges for improvement with various additional parameters.

The “HMT05” method could give higher accuracy than the “baseline” method for any size of the training sentences although the accuracy was lower than the “Mixture” method. The method could also be carried out in much smaller training time and lower cost than the “Mixture” method. These points would be the benefits of the “HMT05” method. On the other hand, when we compared the “HMT05” method with the “GENIA only” method, for the larger size of the training corpus, the “HMT05” method was defeated by the “GENIA only” method in parsing accuracy and training cost.

4.3 Impact of Re-training a Lexical Disambiguation Model

When we focused on our method, it could constantly give higher accuracy than the “baseline” and the “HMT05” methods. These results would indicate that, for an individual method, re-training a model of lexical entry assignments might be more critical to domain adaptation than re-training that of tree construction. In addition, for the small treebank, our method could give as high accuracy as the “Mixture” method with much lower training cost. Our method would be a very satisfiable approach when applied with a small treebank. It should be noted that the re-trained lexical model could not solely give the accuracy as high as our method (see “Mixture (lex)” in Figure 3). The combination of a re-trained lexical model and a tree construction model would have given such a high performance.

When we compared the training time for our

Table 2: Parsing accuracy and time for various methods

	For GENIA Corpus				For Penn Treebank			
	LP	LR	F-score	Time	LP	LR	F-score	Time
baseline	86.71	86.08	86.39	476 sec.	89.99	89.63	89.81	675 sec.
GENIA only	88.99	87.91	88.45	242 sec.	72.07	45.78	55.99	2,441 sec.
Mixture	90.01	89.87	89.94	355 sec.	89.93	89.60	89.77	767 sec.
HMT05	88.47	87.89	88.18	510 sec.	88.92	88.61	88.76	778 sec.
Our method	89.11	88.97	89.04	327 sec.	89.96	89.63	89.79	713 sec.
Our method (GENIA)	86.06	85.15	85.60	542 sec.	70.18	44.88	54.75	3,290 sec.
Our method + GENIA	90.02	89.88	89.95	320 sec.	88.11	87.77	87.94	718 sec.
Our method + HMT05	90.23	90.08	90.15	377 sec.	89.31	88.98	89.14	859 sec.
baseline (lex)	85.93	85.27	85.60	377 sec.	87.52	87.13	87.33	553 sec.
GENIA only (lex)	87.42	86.28	86.85	197 sec.	71.49	45.41	55.54	1,928 sec.
Mixture (lex)	88.43	88.18	88.31	258 sec.	87.49	87.12	87.30	585 sec.

Table 3: Training cost of various methods

	Training time	Memory used
baseline	0 sec.	0.00 GByte
GENIA only	14,695 sec.	1.10 GByte
Mixture	238,576 sec.	5.05 GByte
HMT05	21,833 sec.	1.10 GByte
Our method	12,957 sec.	4.27 GByte
Our method (GENIA)	1,419 sec.	0.94 GByte
Our method + GENIA	42,475 sec.	4.27 GByte
Our method + HMT05	31,637 sec.	4.27 GByte
baseline (lex)	0 sec.	0.00 GByte
GENIA only (lex)	1,434 sec.	1.10 GByte
Mixture (lex)	13,595 sec.	4.27 GByte

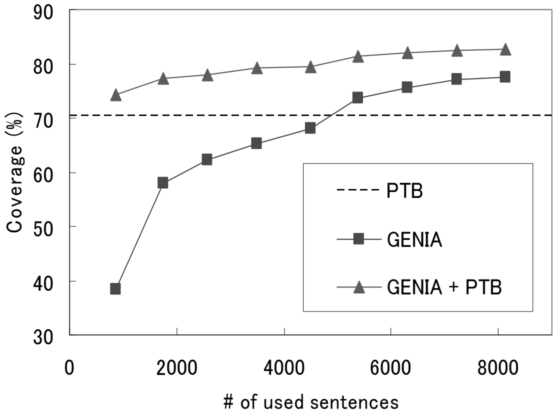

Figure 5: Corpus size vs. coverage of each training set for the GENIA corpus

Table 4: Coverage of each training set

Training set	% of covered sentences	
	for GENIA	for PTB
GENIA treebank	77.54 %	25.66 %
PTB treebank	70.45 %	84.12 %
GENIA treebank + PTB treebank	82.74 %	84.86 %

method with the one for the "HMT05" method, our method required less time than the "HMT05" method. This would be because our method required only the re-training of the very simple model, that is, a probabilistic model of lexical entry assignments.

It should be noted that our method would not work only with in-domain treebank. The "Our method (GENIA)" and the "GENIA only (lex)" methods could hardly give as high parsing accuracy as the "baseline" method. Although, for the larger size of the GENIA treebank, the methods could obtain a little higher accuracy than the "baseline" method, the benefit was very little. These results would indicate that only the treebank in the target domain would be insufficient for adaptation. Figure 5 shows the coverage of each training corpus for the GENIA treebank, which would also support the above observation. It shows that the GENIA treebank could not solely cover so much sentences in the GENIA corpus as the combination of the Penn Treebank and the GENIA treebank.

4.4 Effectiveness of Combining Lexical and Syntactic Disambiguation Models

When we compared the "Our method + HMT05" and "Our method + GENIA" methods with the "Mixture" method, the former two models could give as the high parsing accuracies as the latter one for any size of the training corpus. In particular, for the maximum size, the "Our method + HMT05" models could give a little higher parsing accuracy than the "Mixture" method. This difference was

Table 5: Errors in various methods

	Total errors	=	Common errors with baseline	+	Specific errors
GENIA only	2,889	=	1,906 (65.97%)	+	983 (34.03%)
Mixture	2,653	=	2,177 (82.06%)	+	476 (17.94%)
HMT05	3,063	=	2,470 (80.64%)	+	593 (19.36%)
Our method	2,891	=	2,405 (83.19%)	+	486 (16.81%)
Our method (GENIA)	3,153	=	2,070 (65.65%)	+	1,083 (34.35%)
Our method + GENIA	2,650	=	2,056 (77.58%)	+	594 (22.42%)
Our method + HMT05	2,597	=	1,943 (74.82%)	+	654 (25.18%)
baseline	3,542				
	Total errors	=	Common errors with baseline (lex)	+	Specific errors
GENIA only (lex)	3,320	=	2,509 (75.57%)	+	811 (24.43%)
Mixture (lex)	3,100	=	2,769 (89.32%)	+	331 (10.68%)
baseline (lex)	3,757				

Table 6: Types of disambiguation errors

Error cause	# of errors		
	Common	Only for	
		Baseline	Adapted
Attachment ambiguity			
prepositional phrase	12	12	6
relative clause	0	1	0
adjective	4	2	2
adverb	1	3	1
verb phrase	10	3	1
subordinate clause	0	2	0
Argument/modifier distinction			
to-infinitive	0	0	7
Lexical ambiguity			
preposition/modifier	0	3	0
verb subcategorization frame	5	0	6
participle/adjective	0	2	0
Test set errors			
Errors of treebank	2	0	0
Other types of error causes			
Comma	10	8	4
Noun phrase identification	21	5	8
Coordination/insertion	6	3	5
Zero-pronoun resolution	8	1	0
Others	1	1	2

shown to be significant according to stratified shuffling test with p-value $<$ 0.10, which might suggest the beneficial impact of the “Our method + HMT05” method. In addition, Figure 4 and Table 3 show that training the “Our method + HMT05” or “Our method + GENIA” model required much less time and PC memory than training the “Mixture” model. According to the above observation, we would be able to say that the “Our method + HMT05” method might be the most ideal among the given methods.

The “Our method + HMT05” and “Our method + GENIA” methods showed the different performances in the point that the former could obtain high parsing accuracy with less training time than the latter. This would come from the fact that the latter method trained $q_{syn-genia}$ solely with lexical entries in the GENIA treebank, while the former one trained q_{syn} with rich lexical entries borrowed from $q_{lex-mix}$. Rich lexical entries would decrease unknown lexical entries, and therefore would improve the effectiveness of making the feature forest model. On the other hand, the difference in lexical entries would not seem to affect so much on the contribution of tree construction model to the parsing accuracy. In our experiments, the parameters for a tree construction model such as feature functions were not adjusted thoroughly, which might possibly blur the benefits of the rich lexical entries.

4.5 Error Analysis

Table 5 shows the comparison of the number of errors for various models with that for the original model in parsing the GENIA corpus. For each of the methods, the table gives the numbers of common errors with the original Enju model and the ones specific to that method. If possible, we would like our methods to decrease the errors in the original Enju model while not increasing new errors. The table shows that our method gave the least percentage of newly added errors among the approaches except for the methods utilizing only lexical entry assignments models. On the other hand, the “Our method + HMT05” approach gave over 25 % of newly added errors, although we considered above that the approach gave the best performance.

In order to explore this phenomenon, we observed the errors for the "Our method + HMT05" and the baseline models, and then classified them into several types. Table 6 shows manual classification of causes of errors for the two models in 50 sentences. In the classification, one error often propagated and resulted in multiple errors of predicate argument dependencies. The numbers in the table include such double counting. It would be desirable that the errors in the rightmost column were less than the ones in the middle column, which means that the "Our method + HMT05" approach did not produce more errors specific to the approach than the baseline.

With the "Our method + HMT05" approach, errors for "attachment ambiguity" decreased as a whole. Errors for "comma" and lexical ambiguities of "preposition/modifier" and "participle/adjective" also decreased. For these attributes, the approach could learn in the training phase lexical properties of continuous words with the lexical entry assignment model, and syntactic relations of separated words with the tree construction model. On the other hand, the errors for "to-infinitive argument/modifier distinction" and "verb subcategorization frame ambiguity" considerably increased. These two types of errors have close relation to each other because the failure to recognize verb subcategorization frames tends to cause the failure to recognize the syntactic role of the to-infinitives. We must research further on these errors in our future work.

When we focused on "noun phrase identification," most of the errors did not differ between the two models. In the biomedical domain, there would be many technical terms which could not be correctly identified solely with the disambiguation model, which would possibly result in such many untouched errors. In order to properly cope with these terms, we might have to introduce some kinds of dictionaries or named entity recognition methods.

5 Experiments with the Brown Corpus

5.1 Brown Corpus

We applied our methods to the Brown corpus (Kucera and Francis, 1967) and examined the portability of our method. The Brown corpus consists of 15 domains, and the Penn Treebank gives bracketed version of the corpus for the 8 domains containing 19,395 sentences (Table 7).

Table 7: Domains in the Brown corpus

label	domain	sentences
CF	popular lore	2,420
CG	belles lettres	2,546
CK	general fiction	3,172
CL	mystery and detective fiction	2,745
CM	science fiction	615
CN	adventure and western fiction	3,521
CP	romance and love story	3,089
CR	humor	812
All	total of all the above domains	19,395

For the target of adaptation, we utilized the domain containing all of these 8 domains as a total fiction domain (labelled "All") as well as the individual ones. As in the experiments with the GENIA Treebank, we divided sentences for each domain into three parts, 80% for training, 10% for develepment test, and 10% for final test. For the "All" domain, we merged all training sets, all development test sets, and all final test sets for the 8 domains respectively.

Table 8 and 9 show the parsing accuracy and training time for each domain with the various methods shown in Section 3. The methods are fundamentally the same as in the experiments with the GENIA corpus except that the target corpus is replaced with the Brown corpus. In order to avoid confusion, we replaced "GENIA" in the names of the methods with "Brown." Each of the bold numbers in Table 8 means that it was the best accuracy given for the target domain. It should be noted that the "CM" and "CR" domain contains very small treebank, and therefore we must consider that the results with these domains would not be so useful.

5.2 Evaluation of Portability of Our Method

When we focus on the "ALL" domain, the approaches other than the baseline succeeded to give higher parsing accuracy than the baseline. This would show that these approaches were effective not only for the GENIA corpus but also for the Brown corpus. The "Mixture" method gave the highest accuracy which was 3.41 point higher than the baseline. The "Our method + HMT05" approach also gave the accuracy as high as the "Mixture" method. In addition, as is the case with the GENIA corpus, the approach could be trained with much less time than the "Mixture" method. Not only for these two

Table 8: Parsing accuracy for the Brown corpus

	F-score								
	ALL	**CF**	**CG**	**CK**	**CL**	**CM**	**CN**	**CP**	**CR**
baseline	83.09	85.75	85.38	81.12	77.53	85.30	82.84	85.18	76.63
Brown only	84.84	77.65	78.92	75.72	70.56	50.02	78.38	79.10	50.34
Mixture	**86.50**	86.59	**85.94**	82.49	**78.66**	84.82	84.28	86.85	76.45
HMT05	83.79	85.80	84.98	81.48	76.91	85.25	83.50	85.66	77.15
Our method	86.14	86.73	85.74	82.77	77.95	85.40	84.23	**86.90**	76.71
Our method (GENIA)	84.71	78.49	79.63	75.43	70.86	50.24	78.49	79.69	51.82
Our method + GENIA	86.00	86.12	85.41	**83.22**	77.10	83.39	84.21	85.77	76.91
Our method + HMT05	86.44	**86.76**	85.85	82.90	77.70	**85.61**	**84.43**	86.87	77.48
baseline (lex)	82.19	84.69	83.85	80.25	76.32	83.42	81.29	84.13	77.33
Brown only (lex)	83.92	77.12	77.81	75.06	70.35	49.95	77.06	78.84	50.63
Mixture (lex)	85.29	85.47	84.18	81.88	77.22	83.98	82.67	85.65	**77.58**

Table 9: Consumed time for various methods for the Brown corpus

	Consumed time for training (sec.)								
	ALL	**CF**	**CG**	**CK**	**CL**	**CM**	**CN**	**CP**	**CR**
baseline	0	0	0	0	0	0	0	0	0
Brown only	42,614	4,115	3,763	2,478	2,162	925	2,362	2,695	1,226
Mixture	383,557	190,449	159,490	156,299	210,357	131,335	170,108	224,045	184,251
HMT05	30,933	6,003	4,830	4,186	5,010	1,681	4,411	5,069	1,588
Our method	15,912	11,053	10,988	11,151	10,782	10,158	11,075	10,594	10,284
Our method (Brown)	3,273	312	373	310	249	46	321	317	86
Our method + Brown	130,434	24,633	21,848	20,171	19,184	11,995	19,164	20,922	13,461
Our method + HMT05	54,355	17,722	16,627	15,229	14,914	12,226	15,760	16,175	11,724
baseline (lex)	0	0	0	0	0	0	0	0	0
Brown only (lex)	3,001	317	373	308	251	47	321	317	86
Mixture (lex)	21,148	11,128	11,251	11,094	10,728	10,466	11,151	10,897	10,537

methods, the experimental results for the “All” domain showed the tendency similar to the GENIA corpus as a whole, except for the less improvement with the “HMT05” method.

When we focus on the individual domains, our method could successfully obtain higher parsing accuracy than the baseline for all the domains. Moreover, for the “CP” domain, our method could give the highest parsing accuracy among the methods. These results would support the portability of retraining the model for lexical entry assignment. The “Our method + HMT05” approach, which gave the highest performance for the GENIA corpus, also gave accuracy improvement for the all domains while it did not give so much impact for the “CL” domain. The “Mixture” approach, which utilized the same lexical entry assignment model, could obtain 0.94 point higher parsing accuracy than the “Our method + HMT05” approach. Table 10, which shows the lexical coverage with each domains, does not seem to indicate the noteworthy difference in lexical entry coverage between the “CL” and the other domains. As mentioned in the error analysis in Section 4, the model of tree construction might affect the performance in some way. In our future work, we must clarify the mechanism of this result and would like to further improve the performance.

6 Related Work

For recent years, domain adaptation has been studied extensively. This section explores how our research is relevant to the previous works.

Our previous work (Hara et al., 2005) and this research mainly focused on how to draw as much benefit from a smaller amount of in-domain annotated data as possible. Titov and Henderson (2006) also took this type of approach. They first trained a probabilistic model on original and target treebanks and used it to define a kernel over parse trees. This kernel was used in a large margin classifier trained on a small set of data only from the target domain, and the classifier was then used for reranking the top

Table 10: Coverage of each training set for the Brown corpus

Training set	% of covered sentences for the target corpus								
	ALL	**CF**	**CG**	**CK**	**CL**	**CM**	**CN**	**CP**	**CR**
Target treebank	74.99 %	49.13 %	50.00 %	47.97 %	49.08 %	29.66 %	53.51 %	64.01 %	8.57%
PTB treebank	70.02 %	72.09 %	68.93 %	66.42 %	68.87 %	78.62 %	70.00 %	77.59 %	47.14 %
Target + PTB	79.77 %	74.71 %	71.47 %	71.59 %	70.45 %	80.00 %	72.70 %	80.39 %	52.86 %

parses on the target domain.

On the other hand, several studies have explored how to draw useful information from unlabelled in-domain data. Roark and Bacchiani (2003) adapted a lexicalized PCFG by using maximum *a posteriori* (MAP) estimation for handling unlabelled adaptation data. In the field of classifications, Blitzer et al. (2006) utilized unlabelled corpora to extract features of structural correspondences, and then adapted a POS-tagger to a biomedical domain. Steedman et al. (2003) utilized a co-training parser for adaptation and showed that co-training is effective even across domains. McClosky et al. (2006) adapted a re-ranking parser to a target domain by self-training the parser with unlabelled data in the target domain. Clegg and Shepherd (2005) combined several existing parsers with voting schemes or parse selection, and then succeeded to gain the improvement of performance for a biomedical domain. Although unsupervised methods can exploit large in-domain data, the above studies could not obtain the accuracy as high as that for an original domain, even with the sufficient size of the unlabelled corpora. On the other hand, we showed that our approach could achieve this goal with about 6,500 labelled sentences. However, when 6,500 labelled can not be prepared, it might be worth while to explore the potentiality of combining the above unsupervised and our supervised methods.

When we focuses on biomedical domains, there have also been various works which coped with domain adaptation. Biomedical sentences contain many technical terms which cannot be easily recognized without expert knowledge, and this damages performances of NLP tools directly. In order to solve this problem, two types of approaches have been suggested. The first approach is to utilize existing domain-specific lexical resources. Lease and Charniak (2005) utilized POS tags, dictionary collocations, and named entities for parser adaptation, and then succeeded to achieve accuracy improvement. The second approach is to expand lexical entries for a target domain. Szolovits (2003) extended a lexical dictionary for a target domain by predicting lexical information for words. They transplanted lexical *indiscernibility* of words in an original domain into a target domain. Pyysalo et al. (2004) showed the experimental results that this approach improved the performance of a parser for Link Grammar. Since our re-trained model of lexical entry assignments was shown to be unable to cope with this problem properly (shown in Section 4), the combination of the above approaches with our approach would be expected to bring further improvement.

7 Conclusions

This paper presented an effective approach to adapting an HPSG parser trained on the Penn Treebank to a biomedical domain. We trained a probabilistic model of lexical entry assignments in a target domain and then incorporated it into the original parser. The experimental results showed that this approach obtains higher parsing accuracy than the existing approach of adapting the structural model alone. Moreover, the results showed that, the combination of our method and the existing approach could achieve parsing accuracy that is as high as that obtained by re-training an HPSG parser for the target domain from scratch, but with much lower training cost. With this model, the parsing accuracy for the target domain improved by 3.84 f-score points, using a domain-specific treebank of 8,127 sentences. Experiments showed that 6,500 sentences are sufficient for achieving as high parsing accuracy as the baseline for the original domain.

In addition, we applied our method to the Brown corpus in order to evaluate the portability of our method. Experimental results showed that the parsing accuracy for the target domain improved by 3.35 f-score points. On the other hand, when we focused on some individual domains, that combination approach could not give the desirable results.

In future work, we would like to explore further performance improvement of our approach. For the first step, domain-specific features such as named entities could be much help for solving unsuccessful recognition of technical terms.

Acknowledgment

This research was partially supported by Grant-in-Aid for Specially Promoted Research 18002007.

References

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

A. L. Berger, S. A. Della Pietra, and V. J. Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71.

J. Blitzer, R. McDonald, and F. Pereira. 2006. Domain adaptation with structural correspondence learning. In *Proc. EMNLP 2006*.

Y. S. Chan and H. T. Ng. 2006. Estimating class priors in domain adaptation for word sense disambiguation. In *Proc. 21st COLING and 44th ACL*.

S. Chen and R. Rosenfeld. 1999. A gaussian prior for smoothing maximum entropy models. Technical Report CMUCS-99-108, Carnegie Mellon University.

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

S. Clark and J. R. Curran. 2004b. Parsing the WSJ using CCG and log-linear models. In *Proc. 42nd ACL*.

S. Clark and J. R. Curran. 2006. Partial training for a lexicalized-grammar parser. In *Proc. NAACL-06*.

A. B. Clegg and A. Shepherd. 2005. Evaluating and integrating treebank parsers on a biomedical corpus. In *Proc. the ACL Workshop on Software*.

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

T. Hara, Y. Miyao, and J. Tsujii. 2005. Adapting a probabilistic disambiguation model of an HPSG parser to a new domain. In *Proc. IJCNLP 2005*.

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

M. Johnson and S. Riezler. 2000. Exploiting auxiliary distributions in stochastic unification-based grammars. In *Proc. 1st NAACL*.

J. D. Kim, T. Ohta, Y. Teteisi, and J. Tsujii. 2003. GENIA corpus - a semantically annotated corpus for biotextmining. *Bioinformatics*, 19(suppl. 1):i180–i182.

H. Kucera and W. N. Francis. 1967. *Computational Analysis of Present-Day American English*. Brown University Press, Providence, RI.

M. Lease and E. Charniak. 2005. Parsing biomedical literature. In *In Proc. IJCNLP 2005*.

M. Marcus, G. Kim, M. A. Marcinkiewicz, R. MacIntyre, A. Bies, M. Ferguson, K. Katz, and B. Schasberger. 1994. The Penn Treebank: Annotating predicate argument structure. In *ARPA HLT Workshop*.

D. McClosky, E. Charniak, and M. Johnson. 2006. Reranking and self-training for parser adaptation. In *Proc. 21st COLING and 44th ACL*.

Y. Miyao and J. Tsujii. 2002. Maximum entropy estimation for feature forests. In *Proc. HLT 2002*.

Y. Miyao and J. Tsujii. 2005. Probabilistic disambiguation models for wide-coverage HPSG parsing. In *Proc. ACL 2005*.

Y. Miyao, T. Ninomiya, and J. Tsujii. 2004. Corpus-oriented grammar development for acquiring a Head-driven Phrase Structure Grammar from the Penn Treebank. In *Proc. IJCNLP-04*.

T. Ninomiya, T. Matsuzaki, Y. Tsuruoka, Y. Miyao, and J. Tsujii. 2006. Extremely lexicalized models for accurate and fast HPSG parsing. In *Proc. EMNLP 2006*.

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

S. Pyysalo, F. Ginter, T. Pahikkala, J. Koivula, J. Boberg, J. Jrvinen, and T. Salakoski. 2004. Analysis of Link Grammar on biomedical dependency corpus targeted at protein-protein interactions. In *Proc. BioNLP/NLPBA 2004*.

B. Roark and M. Bacchiani. 2003. Supervised and unsupervised PCFG adaptation to novel domains. In *Proc. HLT-NAACL 2003*.

M. Steedman, M. Osborne, A. Sarkar, S. Clark, R. Hwa, J. Hockenmaier, P. Ruhen, S. Baker, and J. Crim. 2003. Bootstrapping statistical parsers from small datasets. In *Proc. European ACL (EACL)*.

P. Szolovits. 2003. Adding a medical lexicon to an English parser. In *AMIA Annu Symp Proc*.

Ivan Titov and James Henderson. 2006. Porting statistical parsers with data-defined kernels. In *Proc. CoNLL-2006*.