Towards Accurate Probabilistic Lexicons for Lexicalized Grammars

Naoki Yoshinaga
Institute of Industrial Science, University of Tokyo
6-1 Komaba 4-chome, Meguro-Ku, Tokyo 153-8505, JAPAN
ynaga at tkl.iis.u-tokyo.ac.jp

Abstract

This paper proposes a method of constructing an accurate probabilistic subcategorization (SCF) lexicon for a lexicalized grammar extracted from a treebank. We employ a latent variable model to smooth co-occurrence probabilities between verbs and SCF types in the extracted lexicalized grammar. We applied our method to a verb SCF lexicon of an HPSG grammar acquired from the Penn Treebank. Experimental results show that probabilistic SCF lexicons obtained by our model achieved a lower test-set perplexity against ones obtained by a naive smoothing model using twice as large training data.

1 Introduction

This paper proposes a smoothing model for probabilistic subcategorization (SCF) lexicons of lexicalized grammars acquired from corpora. Here, *an* SCF *lexicon* consists of pairs of words and *lexical (*SCF*) types* (*e.g*, *tree family*), from which individual lexical entry templates are derived by lexical rules (Jackendoff, 1975; Pollard and Sag, 1994) (*e.g.*, *metarules*: Becker (2000) and Prolo (2002)).[1] Recently, the corpus-oriented approaches have enabled us to acquire wide-coverage lexicalized grammars from large treebanks (Xia, 1999; Chen and Vijay-Shanker, 2000; Chiang, 2000; Hockenmaier and Steedman, 2002; Cahill et al., 2002; Frank et al., 2003; Miyao et al., 2005). However, a great workload is required to develop such large treebanks for languages or domains where a base bracketed corpus (*e.g.*, the Penn Treebank: Marcus et al. (1993)) is not available. When the size of the source treebank is small, we encounter the serious problem of a lack of lexical entries (unseen word-template pairs).

Previous studies investigated unseen word-template pairs in lexicalized grammars acquired from the Penn Treebank (Xia, 1999; Chen and Vijay-Shanker, 2000; Hockenmaier and Steedman, 2002; Miyao et al., 2005); the words can be seen (sw) or unseen (uw), and similarly, the templates can be seen (st) or unseen (ut), so that there are four types of unseen pairs. All the studies reported that unseen (sw, st) pairs caused the major problem in lexical coverage.

This paper focuses on a verb SCF lexicon, and employs a latent variable model (Hofmann, 2001) to smooth co-occurrence probabilities between verbs and SCF types acquired from small-sized corpora. If we can obtain such an accurate probabilistic SCF lexicon, we can construct a wide-coverage SCF lexicon by setting the threshold of the probabilities (Yoshinaga, 2004). Alternatively we can directly use the acquired probabilistic lexicon in supertagging (Chen et al., 2006) and probabilistic parsing (Miyao et al., 2005; Ninomiya et al., 2005).

We applied our method to a verb SCF lexicon of an HPSG grammar acquired from the Penn Treebank (Miyao et al., 2005; Nakanishi et al., 2004). The acquired probabilistic SCF lexicons were more accurate than ones acquired by a naive smoothing model.

[1] In the linguistic literature, the term 'lexical rules' is used to define either syntactic transformations (*e.g.*, wh-movement), diathesis alternations (*e.g.*, dative shift) or both. In this paper, we use the term lexical rules to define syntactic transformations among lexical entry templates that belong to the same lexical type.

2 Related Work

In this section, we first describe previous approaches to the problem of unseen word-template pairs in the lexicalized grammars acquired from treebanks. We then address smoothing methods for SCF lexicons acquired from raw corpora.

2.1 Predicting unseen word-template pairs for lexicalized grammars

The problem of missing lexical entries has been recognized as one of the major problems in lexicalized grammars acquired from treebanks, and a number of researchers attempted to predict unseen lexical entries. In the following, we describe previous methods of predicting unseen (uw, st) and (sw, st) pairs, respectively.[2]

Chiang (2000), Hockenmaier and Steedman (2002) and Miyao et al. (2005) used a simple smoothing method to predict unseen (uw, st) pairs. They regarded infrequent words in the source treebank as unknown words, and assigned the lexical entry templates acquired for these words to unknown words. This treatment of unknown words substantially improved the lexical coverage, probably because infrequent words are likely to take only a few lexical entry templates (*e.g.*, those for transitive verbs).

There are two types of approaches to predict unseen (sw, st) pairs. The first type of approaches (Chen and Vijay-Shanker, 2000; Nakanishi et al., 2004; Chen et al., 2006) exploited an organization of lexical entry templates studied in the linguistic literature; namely, individual lexical entry templates are grouped in terms of higher-level lexical (SCF) types. When a word takes a lexical entry template that belongs to a certain lexical type t, it should take all the other lexical entry templates that belong to t. To identify a set of lexical entry templates that belong to the same lexical type, Chen and Vijay-Shanker (2000) associated the lexical entry templates with tree families in a manually-tailored LTAG (The XTAG Research Group, 1995), Chen et al. (2006) converted the lexical entry templates into linguistically-motivated feature vectors, and Nakanishi et al. (2004) manually defined lexical rules. These methods, however, just translate the problem of unseen word-template pairs into the problem of unseen word-type pairs, and does not predict any unseen word-type pairs. We will hereafter refer to four types of unseen word-type pairs by (sw, sT), (sw, uT), (uw, sT), and (uw, uT) where sT/uT stand for seen/unseen lexical types.

Another type of the approaches has been taken by Hara et al. (2002) and Chen et al. (2006) to predict unseen (sw, st) pairs. Hara et al. (2002) conducted a hard clustering (Forgy, 1965) of words according to their lexical entry templates in order to find classes of words that take the same lexical entry templates. It will be difficult for the hard clustering method to appropriately classify polysemic verbs, which take several lexical types. Chen et al. (2006) performed a clustering of lexical entry templates according to words that take those templates in order to find lexical entry templates that belong to the same tree family. They reported that it was difficult to predict infrequent lexical entry templates by their method. These studies directly encode word-template pairs into vectors for clustering, which will suffer from the data sparseness problem.

In this study, we focus on probabilistic modeling of unseen word-type pairs in the lexicalized grammars, since we can associate lexical entry templates with lexical types by using the aforementioned methods (Chen and Vijay-Shanker, 2000; Nakanishi et al., 2004; Chen et al., 2006). This reduces the number of parameters in the probabilistic models drastically, which will make it easier to estimate an accurate probabilistic model from sparse data.

2.2 Predicting unseen word-SCF pairs for pre-defined SCF types

There are some studies on smoothing SCF lexicons acquired for pre-defined SCF types from raw corpora (Korhonen, 2002; Yoshinaga, 2004). These studies aimed at predicting unseen (sw, sT) pairs for the acquired SCF lexicons. Korhonen (2002) first semi-automatically determined verb semantic classes using Levin's verb classification (Levin, 1993) and WordNet (Fellbaum, 1998), and then employed SCF distributions for represen-

[2] Most of the previous studies attempted to avoid the problem of unseen (sw, ut) and (uw, ut) pairs by modifying the source treebank so as to generalize the resulting grammar; for example, Chen and Vijay-Shanker (2000) used a compact label set instead of one given in the original treebank. Nakanishi et al. (2004) predicted unseen (sw, ut) and (uw, ut) pairs for a given lexicalized grammar by newly creating unseen lexical entry templates using manually defined lexical rules.

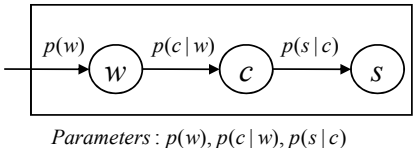

Figure 1: Probabilistic latent semantic analysis of a co-occurrence between words and SCFs

tative verbs in each obtained verb class to calculate accurate back-off estimates for the verbs in that class. Yoshinaga (2004) conducted clustering of verbs according to their SCF confidence vectors, and then used the resulting classes to predict possible SCFs. Both studies successfully predicted unseen word-type pairs for the pre-defined SCF types.

3 PLSA-based Probabilistic SCF Lexicon

This section first applies the probabilistic latent semantic analysis (PLSA: Hofmann (2001)) to co-occurrences between verbs and SCFs, and then describes a PLSA-based smoothing model to estimate the co-occurrence probabilities.

3.1 PLSA to model co-occurrences between verbs and SCF types

We employ the probabilistic latent semantic analysis to model co-occurrences between words and SCF types, where the latent variables are classes whose members have the same SCF distribution. Our modeling is inspired by the studies by Schulte im Walde and Brew (2002) and Korhonen et al. (2003), which demonstrated that a semantic classification of verbs can be obtained by clustering verbs according to their SCF distributions.[3] The PLSA is suitable for this task since it performs a kind of soft clustering, which can naturally handle highly polysemic nature of verbs.

We assume that a lexicon of a lexicalized grammar is acquired from a source treebank. Let the conditional probability that a word $w \in W$ appears as a member of a latent class $c \in C$ be $p(c|w)$, and each latent class $c \in C$ takes an SCF $s \in S$ with a conditional probability $p(s|c)$. Here, W and S are a set of words and lexical types seen in the source treebank. When we assume that a word w occurs with a probability $p(w)$, a co-occurrence probability between w and s, $p(w, s)$, is given by:

$$p(w, s) = p(w) \sum_{c \in C} p(c|w)p(s|c).$$

Figure 1 shows our SCF modeling. This generative model has a smoothing effect since the number of free parameters becomes smaller than a simple tabulation model, which directly computes $p(w, s)$ from the observed frequency, by setting the number of the latent variables to a small value.

We then apply a variant of the Expectation Maximization (EM) algorithm (Dempster et al., 1997) called *tempered* EM (Hofmann, 2001) to estimate parameters of this model. In what follows, We first derive the update formulas for the parameters in our model by the EM algorithm, and then explain the tempered EM algorithm.

We assume that the set of parameters θ^t at the t-th iteration is updated to θ^{t+1} at the next iteration, and refer to the individual parameters at the t-th iteration by $p_{\theta^t}(\cdot)$. The update formulas for the individual parameters are derived by constrained optimization of $Q(\theta, \theta^t)$, which defined by

$$Q(\theta, \theta^t) = \sum_{w \in W} \sum_{s \in S} n(w, s) \sum_{c \in C} p_{\theta^t}(c|w, s) \times \log[p_\theta(w) \sum_{c \in C} p_\theta(c|w)p_\theta(s|c)] \quad (1)$$

where

$$p_{\theta^t}(c|w, s) = \frac{p_{\theta^t}(c|w)p_{\theta^t}(s|c)}{\sum_{c \in C} p_{\theta^t}(c|w)p_{\theta^t}(s|c)}$$

and $n(w, s)$ is the observed frequency of a co-occurrence between w and s in the source treebank. Using the Lagrange multiplier method, we obtain the updated parameters $\theta = \theta_{t+1}$ which maximize the Q-function in Equation 1 as follows:

$$p_{\theta_{t+1}}(c|w) = \frac{\sum_{s \in S} n(w, s)p_{\theta^t}(c|w, s)}{n(w)},$$

$$p_{\theta_{t+1}}(s|c) = \frac{\sum_{w \in W} n(w, s)p_{\theta^t}(c|w, s)}{\sum_{w \in W} \sum_{s \in S} n(w, s)p_{\theta^t}(c|w, s)},$$

$$p_{\theta_{t+1}}(w) = \frac{n(w)}{\sum_{w \in W} n(w)}$$

where $n(w)$ is the observed frequency of a word w in the source treebank.

The tempered EM is closely related to deterministic annealing (Rose et al., 1990), and introduces *an inverse computational temperature* β to the EM

[3] Although a comparison between classes obtained by our method with those obtained by their methods must be interesting, we focus on the effect of smoothing in this paper.

algorithm to reduce the sensitivity to local optima and to avoid overfitting. The update formulas for the tempered EM are obtained by replacing p_{θ^t} in the original formulas by the following equation[4]:

$$p_{\theta^t}(c|w,s) = \frac{[p_{\theta^t}(c|w)p_{\theta^t}(s|c)]^\beta}{\sum_{c\in C}[p_{\theta^t}(c|w)p_{\theta^t}(s|c)]^\beta}.$$

We follow Hofmann's approach (Hofmann, 2001) to determine the optimal value of β. We initialize β to 1 and run the EM iterations with *early stopping* (as long as the performance on held-out data improves). We then rescale β by a factor η ($= 0.5$, in the following experiments) and again run the EM iterations with early stopping. We repeat this rescaling until it no longer improves the result.

3.2 Smoothing model for SCF lexicons

We then use the PLSA model described in the previous section to obtain accurate estimates for the co-occurrence probabilities between words and SCFs. In this study, we focus on smoothing co-occurrence probabilities of word-type pairs for seen SCF types, (sw, sT) and (uw, sT). Acquisition of unseen SCF types (and corresponding templates) is beyond the scope of this study.

In what follows, we first mention a smoothing model for co-occurrence probabilities of (uw, sT) pairs, and then describe a smoothing model for co-occurrence probabilities of (sw, sT) pairs.

3.2.1 Estimation of word-type co-occurrence probabilities for unknown words

Following the previous studies (Chiang, 2000; Hockenmaier and Steedman, 2002; Miyao et al., 2005) described in Section 2.1, we calculate a co-occurrence probability between an unseen word w' and a seen SCF type s as follows:

$$p^m_{unseen}(s|w') = \mu_1 p^m_{\text{MLE}}(s) + \mu_2 p_{\text{MLE}}(s) \quad (2)$$

where

$$p^m_{\text{MLE}}(s) = \frac{\sum_{w\in\{w|n(w)\le m\}} n(w,s)}{\sum_{w\in\{w|n(w)\le m\}}\sum_{s\in S} n(w,s)},$$
$$p_{\text{MLE}}(s) = \frac{\sum_{w\in W} n(w,s)}{\sum_{w\in W}\sum_{s\in S} n(w,s)}, \quad (3)$$

and μ_i is *a weight* of each probabilistic model, which satisfies the constraint $\sum_{i=1}^2 \mu_i = 1$. We estimate μ_i by the EM algorithm using held-out data.

[4]The interested readers are referred to the cited literature (Hofmann, 2001) to see the technical details.

In short, we regard infrequent words that appear less than or equal to m in the source treebank as unknown words, and use the observed frequency of SCFs for these words to calculate the co-occurrence probabilities. We assume $p^0_{unseen}(s|w') = p_{\text{MLE}}(s)$.

3.2.2 Estimation of word-type co-occurrence probabilities for known words

To estimate a co-occurrence probability between a seen word w and a seen SCF s, we interpolate the following three models. The first model provides the maximum likelihood estimation (MLE) of the co-occurrence probability, which is computed by:

$$p_{\text{MLE}}(s|w) = \frac{n(w,s)}{\sum_{s\in S} n(w,s)}.$$

The second model provides a smoothed probability based on the PLSA model, which is calculated by:

$$p^n_{\text{PLSA}}(s|w) = \sum_{c\in C} p(c|w)p(s|c)$$

where $p(c|w)$ and $p(s|c)$ are probabilities estimated under the PLSA model and n is the number of the latent classes. We should note that the above two models are computed using all the word-type pairs observed in the source treebank (including the word-type pairs for the infrequent words used in Equation 2).

The last model provides $p_{\text{MLE}}(s)$ in Equation 3, which is the maximum likelihood estimation of $p(s)$. We combine these three models by linear interpolation:

$$p^n_{seen}(s|w) = \lambda_1 p_{\text{MLE}}(s|w) + \lambda_2 p^n_{\text{PLSA}}(s|w) + \lambda_3 p_{\text{MLE}}(s)$$

where $\sum_{i=1}^3 \lambda_i = 1$.

In summary, when we regard words that appear less than or equal to m as unknown words, we obtain a co-occurrence probability of a word w and an SCF type s as follows:[5]

$$p^{m,n}(s|w) = \begin{cases} p^n_{seen}(s|w) & (n(w) > m) \\ p^m_{unseen}(s|w) & (n(w) \le m) \end{cases} \quad (4)$$

[5]We can use $p^n_{seen}(s|w)$ to estimate the co-occurrence probabilities for the infrequent words (*e.g.*, $0 < n(w) \le m$). However, preliminary experiments showed that it slightly deteriorates the accuracy of the resulting probabilistic lexicons.

Table 1: Specification of SCFs for HPSG acquired from WSJ Sections 02-21 and their subsets

	SOURCE TREEBANK										
	02	**02-03**	**02-05**	**02-07**	**02-09**	**02-11**	**02-13**	**02-15**	**02-17**	**02-19**	**02-21**
# SCF types	78	93	135	151	164	175	197	209	215	235	253
# verbs	1,020	1,294	1,936	2,254	2,476	2,704	2,940	3,134	3,334	3,462	3,586
Ave. # SCFs/verb	1.46	1.53	1.61	1.68	1.69	1.72	1.75	1.78	1.80	1.82	1.85

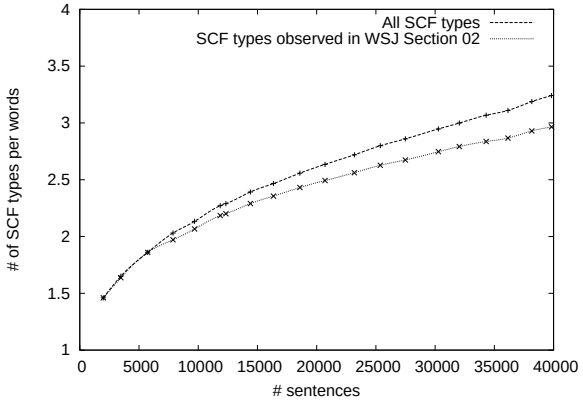

Figure 2: The average number of SCF types assigned to words in WSJ Section 02

In the following section, we compare the above smoothing model with a naive smoothing model, which estimates the co-occurrence probabilities only from $p^m_{\text{MLE}}(s|w)$ and $p_{\text{MLE}}(s)$ as follows:

$$p'^m(s|w) = \begin{cases} \lambda'_1 p_{\text{MLE}}(s|w) + \lambda'_2 p_{\text{MLE}}(s) & \\ \qquad\qquad (n(w) > m) & \\ p^m_{unseen}(s) \quad (n(w) \leq m) & \end{cases} \quad (5)$$

where $\sum_{i=1}^{2} \lambda'_i = 1$.

4 Experiments

We investigate the effect of our smoothing model on SCFs acquired for HPSG grammars.

4.1 Data and Settings

We start by extracting word-SCF pairs from Sections 02-21 of the Wall Street Journal (WSJ) portion of the Penn Treebank and their subset sections by use of the existing methods (Miyao et al., 2005; Nakanishi et al., 2004).

Table 1 shows the details of the acquired SCFs. The average number of SCF types acquired for each verb increases rather mildly with the size of the source treebank. However, when we focus on verbs that appeared in Section 2, the average number of SCF types for these verbs increases more rapidly (Figure 2). This is because most of frequent verbs appeared in Section 2, and such verbs took the larger number of SCF types than other infrequent verbs. Figure 2 also confirms that most of the 'frequent' SCF types were seen in a small portion of the treebank (WSJ Section 2). Thus, predicting unseen word-type pairs for seen SCF types will have more impact on the grammar coverage.

We then applied our smoothing model to the acquired SCF lexicons. We constructed five PLSA models $p^n_{\text{PLSA}}(s|w)$ for each acquired set of word-SCF pairs by ranging the number of latent variables n from 40 to 640, and then obtained the linear-interpolated models (Equations 4 and 5) with $m = 0, 1, 2$. The PLSA models and the weights of the linear interpolation are estimated by using WSJ Section 22 as held-out data. To estimate the PLSA models, we ran the tempered EM algorithm 100 times, and chose the model that obtained the largest likelihood on the held-out data, because the estimation of the PLSA models is likely to suffer from local optima due to the large number of free parameters. To estimate the weight μ_i of the models for unknown words $p^m_{unseen}(s)$ in Equation 2, we used word-type pairs (in the held-out data) for the infrequent words and words that did not appear in the source treebank, ($w \in \{w|n(w) \leq m\}$).

To evaluate the accuracy of the estimated co-occurrence probabilities, we employ *the test-set perplexity*, PP, which is defined by:

$$PP = 2^{-\frac{1}{N}\sum_{w\in W_t}\sum_{s\in S_t} n_t(w,s)\log p(w,s)}$$

where W_t and S_t are a set of words and lexical types seen in the test data, $N = \sum_{w\in W_t} n_t(w)$, and $n_t(w)$ and $n_t(w,s)$ are the observed frequency of a word w and a co-occurrence between w and s in the test data, respectively. This measure indicates the complexity of the task that determines an SCF type for a given verb $w \in W_t$ with a model $p(w,s)$.

Table 2: Test-set perplexity of $p(s|w)$ against the test SCFs acquired from WSJ Section 24 for the SCF types that are observed in WSJ Section 2

			SOURCE TREEBANK										
MODEL	m	n	**02**	**02-03**	**02-05**	**02-07**	**02-09**	**02-11**	**02-13**	**02-15**	**02-17**	**02-19**	**02-21**
unknown			10.809	10.779	10.769	10.750	10.747	10.754	10.759	10.751	10.746	10.748	10.739
naive	0		4.030	3.730	3.414	3.303	3.273	3.224	3.172	3.137	3.132	3.124	3.116
PLSA	0	40	**3.786**	3.532	3.253	3.192	3.157	3.118	3.056	3.039	3.048	3.026	3.025
	0	80	3.809	3.540	3.239	3.167	3.132	3.098	3.055	3.034	3.033	3.024	3.019
	0	160	3.843	3.500	3.241	3.153	3.126	3.081	3.051	3.038	3.023	3.023	3.027
	0	320	3.813	**3.498**	3.244	**3.139**	3.127	3.078	**3.037**	3.023	3.025	**3.008**	3.021
	0	640	3.804	3.524	**3.215**	3.142	**3.118**	**3.060**	3.039	**3.016**	**3.011**	3.015	**3.009**
naive	1		3.865	3.616	3.371	3.256	3.225	3.194	3.144	3.104	3.094	3.087	3.071
PLSA	1	40	**3.651**	3.432	3.217	3.147	3.110	3.090	3.031	3.006	3.010	2.990	2.982
	1	80	3.675	3.443	3.202	3.131	3.083	3.067	3.030	3.005	2.996	2.988	2.974
	1	160	3.704	**3.402**	3.210	3.106	3.078	3.058	3.025	3.006	2.993	2.988	2.983
	1	320	3.676	3.405	3.205	3.099	3.082	3.050	3.015	2.995	2.988	**2.975**	2.977
	1	640	3.671	3.425	**3.178**	**3.097**	**3.071**	**3.035**	**3.013**	**2.989**	**2.979**	2.979	**2.967**
naive	2		3.846	3.629	3.384	3.294	3.230	3.205	3.156	3.115	3.104	3.088	3.074
PLSA	2	40	**3.650**	3.460	3.232	3.185	3.125	3.102	3.040	3.014	3.017	2.991	2.985
	2	80	3.675	3.463	3.219	3.171	3.098	3.080	3.038	3.013	3.004	2.989	2.978
	2	160	3.694	**3.432**	3.225	3.147	3.089	3.071	3.033	3.014	3.001	2.989	2.986
	2	320	3.685	3.437	3.218	3.139	3.096	3.062	3.022	3.002	2.997	**2.976**	2.980
	2	640	3.676	3.449	**3.197**	**3.139**	**3.083**	**3.049**	**3.019**	**2.996**	**2.987**	2.980	**2.970**

4.2 Results

Table 2 shows the test-set perplexities against word-SCF pairs acquired from WSJ Section 24. In this result, we excluded SCF types unseen in WSJ Section 2 from the test set to compare models using different source treebanks. In Table 2, *unknown* refers to a model that uses only the observed frequency of SCFs, $p_{\text{MLE}}(s)$, as shown in Equation 3. This model indicates the difficulty of this task. The models *naive* and PLSA refer to the interpolated models with and without the PLSA model which are defined in Equations 4 and 5, respectively. The treatment of unknown words reduced the test-set perplexity (cf. the models with $m = 0$ vs. their counterparts with $m = 1, 2$), and the PLSA-based models further reduced the test-set perplexity compared to the *naive* models, when they were estimated using the same size of corpora. It is also noteworthy that we can achieve a lower test-set perplexity by making the number of latent classes of the PLSA model larger. The optimal number of the latent classes would be between 320 and 640. The probabilistic SCF lexicons obtained with our PLSA-based models achieved a lower test-set perplexity against ones obtained with *naive* models with twice as much training data (cf. *naive* ($m = 1$) estimated with WSJ Section 02-21 vs. PLSA ($(m, n) = (1, 640)$) estimated with WSJ Section 02-11), and even improved the accuracy of the probabilistic SCF lexicon when we use the large source treebank (cf. *naive* and PLSA estimated with WSJ Section 02-21).

Table 3 shows a test-set perplexity against word-SCF pairs acquired from WSJ Section 24, when the test-set perplexity is calculated on all the SCF types observed in the source treebank. In this setting, only models in the same column can be fairly compared. For all the subsets of the treebank, our PLSA-based model achieved a lower test-set perplexity than the naive smoothing model.

5 Conclusion

We have presented a PLSA-based smoothing model for co-occurrence probabilities between verbs and SCFs to construct an accurate probabilistic SCF lexicon for a lexicalized grammar acquired from a small-sized corpus. We applied our smoothing model to SCFs for an HPSG grammar acquired from the Penn Treebank. The proposed smoothing model provided an accurate probabilistic SCF lexicon with a lower test-set perplexity against the one obtained with the naive interpolation model.

In future research, we plan to evaluate the acquired probabilistic SCF lexicon in terms of its contribution to the performance of supertagging (Chen et al., 2006) and probabilistic parsing (Miyao et al., 2005; Ninomiya et al., 2005). We will apply our smoothing model to SCFs for

Table 3: Test-set perplexity of $p(s|w)$ against the test SCFs acquired from WSJ Section 24

			SOURCE TREEBANK										
MODEL	m	n	**02**	**02-03**	**02-05**	**02-07**	**02-09**	**02-11**	**02-13**	**02-15**	**02-17**	**02-19**	**02-21**
unknown			10.809	10.837	11.134	11.162	11.214	11.213	11.349	11.355	11.344	11.338	11.354
naive	0		4.030	3.753	3.524	3.425	3.419	3.362	3.364	3.323	3.297	3.282	3.275
PLSA	0	40	**3.786**	3.552	3.348	3.299	3.280	3.236	3.213	3.197	3.193	3.165	3.169
	0	80	3.809	3.564	3.334	3.268	3.253	3.214	3.209	3.190	3.176	3.163	3.162
	0	160	3.843	3.520	3.337	3.254	3.250	3.197	3.207	3.194	3.168	3.162	3.172
	0	320	3.813	**3.520**	3.342	**3.241**	3.247	3.193	**3.193**	3.180	3.171	**3.148**	3.166
	0	640	3.804	3.543	**3.309**	3.244	**3.244**	**3.173**	3.195	**3.166**	**3.153**	3.156	**3.153**
naive	1		3.865	3.638	3.480	3.377	3.369	3.331	3.334	3.289	3.257	3.244	3.228
PLSA	1	40	**3.651**	3.452	3.311	3.253	3.232	3.207	3.188	3.163	3.154	3.127	3.124
	1	80	3.675	3.466	3.296	3.232	3.203	3.182	3.184	3.160	3.137	3.125	3.116
	1	160	3.704	**3.422**	3.305	3.206	3.201	3.174	3.179	3.160	3.136	3.126	3.126
	1	320	3.676	3.427	3.303	3.200	3.201	3.165	3.170	3.151	3.133	**3.114**	3.120
	1	640	3.671	3.444	**3.272**	**3.199**	**3.196**	**3.149**	**3.168**	**3.138**	**3.119**	3.118	**3.109**
naive	2		3.846	3.651	3.493	3.416	3.375	3.343	3.347	3.300	3.268	3.245	3.231
PLSA	2	40	**3.650**	3.480	3.326	3.293	3.247	3.221	3.197	3.172	3.162	3.128	3.127
	2	80	3.675	3.487	3.314	3.274	3.220	3.197	3.193	3.168	3.146	3.127	3.119
	2	160	3.694	**3.452**	3.320	3.248	3.213	3.187	3.188	3.168	3.145	3.127	3.129
	2	320	3.685	3.459	3.316	**3.242**	3.216	3.177	3.178	3.159	3.142	**3.115**	3.123
	2	640	3.676	3.468	**3.292**	3.242	**3.209**	**3.163**	**3.175**	**3.146**	**3.127**	3.119	**3.113**

LTAGs and other lexicalized grammars acquired from treebank, by using lexical rules (Prolo, 2002) to reduce lexical entries into lexical types. We will also investigate the correspondence between the verb classes obtained by our method and the semantic verb classes suggested by Levin (1993) and Korhonen and Briscoe (2004).

References

Becker, Tilman. 2000. Patterns in metarules for TAG. In Abeillé, Anne and Owen Rambow, editors, *Tree Adjoining Grammars: formalisms, linguistic analysis and processing*, pages 331–342. CSLI Publications.

Cahill, Aoife, Mairead McCarthy, Josef van Genabith, and Andy Way. 2002. Parsing with PCFGs and automatic f-structure annotation. In *Proceedings of the seventh International Lexical-Functional Grammar Conference*, pages 76–95, Athens, Greece.

Chen, John and Krishnamurti Vijay-Shanker. 2000. Automated extraction of TAGs from the Penn Treebank. In *Proceedings of the Sixth International Workshop on Parsing Technologies (IWPT 2000)*, pages 65–76, Trento, Italy.

Chen, John, Srinivas Bangalore, and Krishnamurti Vijay-Shanker. 2006. Automated extraction of Tree-Adjoining Grammars from treebanks. *Natural Language Engineering*, 12(3):251–299, September.

Chiang, David. 2000. Statistical parsing with an automatically-extracted Tree Adjoining Grammar. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL 2000)*, pages 456–463, Hong Kong, China.

Dempster, Arthur P., Nan M. Laird, and Donald B. Rubin. 1997. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, January.

Fellbaum, Christiane, editor. 1998. *WordNet: An Electronic Lexical Database*. The MIT Press.

Forgy, Edward W. 1965. Cluster analysis of multivariate data: Efficiency vs. interpretability of classifications. *Biometrics*, 21(3):768–780, June.

Frank, Anette, Louisa Sadler, Josef van Genabith, and Andy Way. 2003. From treebank resources to LFG f-structures: Automatic f-structure annotation of treebank trees and CFGs extracted from treebanks. In Abeillé, Anne, editor, *Treebanks: Building and Using Syntactically Annotated Corpora*, pages 367–389. Kluwer Academic Publishers.

Hara, Tadayoshi, Yusuke Miyao, and Jun'ichi Tsujii. 2002. Clustering for obtaining syntactic classes of words from automatically extracted LTAG grammars. In *Proceedings of the sixth International Workshop on Tree Adjoining Grammars and Related Frameworks (TAG+6)*, pages 227–233, Venice, Italy.

Hockenmaier, Julia and Mark Steedman. 2002. Acquiring compact lexicalized grammars from a cleaner treebank. In *Proceedings of the third International Conference on Language Resources and Evaluation (LREC 2002)*, pages 1974–1981, Las Palmas, Spain.

Hofmann, Thomas. 2001. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, 42(1):177–196, January.

Jackendoff, Ray S. 1975. Morphological and semantic regularities in the lexicon. *Language*, 51(3):639–671, September.

Korhonen, Anna and Edward J. Briscoe. 2004. Extended lexical-semantic classification of English verbs. In *Proceedings of the Computational Lexical Semantics Workshop at Human Language Technology Conference and the fifth Annual Meeting of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL 2004)*, pages 38–45, Boston, MA, USA.

Korhonen, Anna, Yuval Krymolowski, and Zvika Marx. 2003. Clustering polysemic subcategorization frame distributions semantically. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL 2003)*, pages 64–71, Sapporo, Japan.

Korhonen, Anna. 2002. *Subcategorization Acquisition*. Ph.D. thesis, Computer Laboratory, University of Cambridge, Cambridge, UK.

Levin, Beth. 1993. *English Verb Classes and Alternations: A Preliminary Investigation*. University of Chicago Press.

Marcus, Mitchell, Beatrice Santorini, and Mary A. Marcinkiewicz. 1993. Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics*, 19(2):313–330, June.

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

Nakanishi, Hiroko, Yusuke Miyao, and Jun'ichi Tsujii. 2004. Using inverse lexical rules to acquire a wide-coverage lexicalized grammar. In *Proceedings of the Workshop on Beyond Shallow Analyses at the first International Joint Conference on Natural Language Processing (ijc-NLP 2004)*, Hainan Island, China.

Ninomiya, Takashi, Yoshimasa Tsuruoka, Yusuke Miyao, and Jun'ichi Tsujii. 2005. Efficacy of beam thresholding, unification filtering and hybrid parsing in probabilistic HPSG parsing. In *Proceedings of the ninth International Workshop on Parsing Technologies (IWPT 2005)*, pages 103–114, Vancouver, BC, Canada.

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

Prolo, Carlos A. 2002. Generating the XTAG English grammar using metarules. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING 2002)*, pages 814–820, Taipei, Taiwan.

Rose, Kenneth, Eitan Gurewwitz, and Geoffrey Fox. 1990. A deterministic annealing approach to clustering. *Pattern Recognition Letters*, 11(9):589–594, September.

Schulte im Walde, Sabine and Chris Brew. 2002. Inducing German semantic verb classes from purely syntactic subcategorisation information. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL 2003)*, pages 223–230, Sapporo, Japan.

The XTAG Research Group. 1995. A Lexicalized Tree Adjoining Grammar for English. Technical Report IRCS-95-03, IRCS, University of Pennsylvania.

Xia, Fei. 1999. Extracting Tree Adjoining Grammars from bracketed corpora. In *Proceedings of the fifth Natural Language Processing Pacific Rim Symposium (NLPRS 1999)*, pages 398–403, Beijing, China.

Yoshinaga, Naoki. 2004. Improving the accuracy of subcategorizations acquired from corpora. In *the Companion Volume to the Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL 2004)*, pages 43–48, Barcelona, Spain.