Generalizing Subcategorization Frames Acquired from Corpora Using Lexicalized Grammars

Naoki Yoshinaga[†]
† University of Tokyo
7-3-1 Hongo, Bunkyo-ku,
Tokyo, 113-0033 Japan
yoshinag@is.s.u-tokyo.ac.jp

Jun'ichi Tsujii[†‡]
‡ CREST, JST
4-1-8, Honcho, Kawaguchi-shi,
Saitama, 332-0012 Japan
tsujii@is.s.u-tokyo.ac.jp

Abstract

This paper presents a method of improving the quality of subcategorization frames (SCFs) acquired from corpora in order to augment a lexicon of a lexicalized grammar. We first estimate a confidence value that a word can have each SCF, and create an SCF confidence-value vector for each word. Since the SCF confidence vectors obtained from the lexicon of the target grammar involve co-occurrence tendency among SCFs for words, we can improve the quality of the acquired SCFs by clustering vectors obtained from the acquired SCF lexicon and the lexicon of the target grammar. We apply our method to SCFs acquired from corpora by using a subset of the SCF lexicon of the XTAG English grammar. A comparison between the resulting SCF lexicon and the rest of the lexicon of the XTAG English grammar reveals that we can achieve higher precision and recall compared to naive frequency cut-off.

1 Introduction

Recently, a variety of methods have been proposed for automatic acquisition of subcategorization frames (SCFs) from corpora (Brent, 1993; Manning, 1993; Briscoe and Carroll, 1997; Sarkar and Zeman, 2000; Korhonen, 2002). Although these research efforts aimed at enhancing lexicon resources, there has been little work on evaluating the impact of acquired SCFs on grammar coverage using large-scale lexicalized grammars with the exception of (Carroll and Fang, 2004).

The problem when we combine acquired SCFs with existing lexicalized grammars is lower quality of the acquired SCFs, since they are acquired in an unsupervised manner, rather than being manually coded. If we attempt to compensate for the lack of recall by being less strict in filtering out less likely SCFs, then we will end up with a larger number of lexical entries. This is fatal for parsing with lexicalized grammars, because empirical parsing efficiency and syntactic ambiguity of lexicalized grammars are known to be proportional to the number of lexical entries used in parsing (Sarkar et al., 2000). We therefore need some method to improve the quality of the acquired SCFs.

Schulte im Walde and Brew (2002) and Korhonen (2003) employed clustering of verb SCF (probability) distributions to induce verb semantic classes. Their studies are based on the assumption that verb SCF distributions are closely related to verb semantic classes. Conversely, if we could induce word classes whose element words have the same set of SCFs, we can eliminate SCFs acquired in error from the corpora and predict plausible SCFs unseen in the corpora. This kind of generalization would be useful to improve the quality of the acquired SCFs.

In this paper, we present a method of generalizing SCFs acquired from corpora in order to augment a lexicon of a lexicalized grammar. For words in the acquired SCF lexicon and the lexicon of the target lexicalized grammar, we first estimate a confidence value that a word can have each SCF. We next perform clustering of SCF confidence-value vectors in order to make use of co-occurrence tendency among SCFs for words in the lexicon of the target lexicalized grammar. Since each centroid value of the obtained clusters indicate whether the words in that class have each SCF, we eliminate implausible SCFs and add unobserved but possible SCFs according to that value. In other words, we can generalize the acquired SCFs by the reliable lexicon of the target lexicalized grammar.

We applied our method to SCFs acquired from mobile phone news groups corpus by a method described in (Carroll and Fang, 2004), in order to generalize the acquired SCFs by using a training portion of the SCF lexicon of the XTAG English grammar (XTAG Research Group, 2001), a large-scale Lexicalized Tree Adjoining Grammar (LTAG) (Schabes et al., 1988). We evaluated the resulting SCF lexicon by comparing it to the rest of

TAG+7: Seventh International Workshop on Tree Adjoining Grammar and Related Formalisms.
May 20-22, 2004, Vancouver, BC, CA.
Pages 104-110.

```
(#S(EPATTERN :TARGET |ftp|
             :SUBCAT (VSUBCAT NONE)
             :CLASSES (22 2985)
             :RELIABILITY 0
             :FREQSCORE 0.01640195
             :FREQCNT 2
             :TLTL (VVD VV0)
             :SLTL ((((|ssh| NN1)))
             :OLT1L NIL
             :OLT2L NIL
             :OLT3L NIL :LRL 0))
```

Figure 1: An acquired SCF for a verb "ftp"

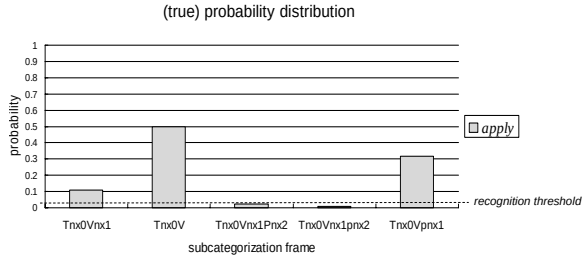

Figure 2: Probability distributions of SCFs for *apply*

the lexicon of the XTAG English grammar, and then compared the results with those obtained by naive frequency cut-off.

2 Background

2.1 Acquisition of SCFs for Lexicalized Grammars

We start by acquiring SCFs for a lexicalized grammar from corpora by the method described in (Carroll and Fang, 2004).

In their study, they first acquire fine-grained SCFs by the method proposed by (Briscoe and Carroll, 1997; Korhonen, 2002). Figure 1 shows an example of one acquired SCF entry for a verb "*ftp*." Each acquired SCF entry has several fields about the observed SCF. We explain here only its portion related to this study. The TARGET field is a word stem (|ftp| in Figure 1), the first number in the CLASSES field indicates an SCF ID (22 in Figure 1), and FREQCNT shows how often words derivable from the word stem had the SCF identified by the SCF ID (2 times in Figure 1) in the training corpus. The obtained SCFs comprise the total 163 types of relatively fine-grained SCFs, which are originally based on the SCFs in the ANLT (Boguraev and Briscoe, 1987) and COMLEX (Grishman et al., 1994) dictionaries. In this example, the SCF ID 22 corresponds to an SCF of intransitive verb.

They then obtain SCFs for the target lexicalized grammar (the LINGO English Resource Grammar (Flickinger, 2000) in their study) by using a handcrafted translation map from these 163 types to one of the types of SCFs in the target grammar. They report that they could achieve a coverage improvement of 4.5% (52.7% to 57.2%) with a parsing time double (9.78 sec. to 21.78 sec.).

This approach is easily extensible to any lexicalized grammars, if the grammars have an organized architecture of lexicon, which derive possible lexical entries from each SCF the grammar defines. Existing lexicalized grammars usually are equipped with this kind of organization, *e.g.*, lexical types in LINGO ERG and tree families in the XTAG English grammar.

2.2 Clustering of Verb SCF Distributions

There are some related work on clustering of SCF probability distributions (Schulte im Walde and Brew, 2002; Korhonen et al., 2003). These studies aim at obtaining verb semantic classes, which closely related to syntactic behavior of argument selection.

Schulte im Walde and Brew (2002) employed clustering of verb SCF distributions to induce verb semantic classes. They first represent a verb SCF distribution by an n-dimensional vector for each verb. Each element in the SCF distribution represents a probability that a verb appears with the corresponding SCF. They then perform k-Means clustering (Forgy, 1965) of these vectors in order to obtain verb semantic classes.

Korhonen et al. (2003) also conducted clustering of verb SCF distributions using a different clustering method including the nearest neighbors clustering and the Information Bottleneck clustering (Tishby et al., 1999). They investigated the effect of polysemic verbs on clustering.

Although these studies demonstrated that there is a certain classification of verbs by clustering of verb SCF distributions, they do not focus on the improvement of the quality of the SCF lexicon. In this paper, we focus on the problem to identify whether a word can have each SCF and try to obtain word classes whose element words have the same set of SCFs.

3 Method

The basic idea of our method is first to obtain word classes whose element words have the same set of SCFs, using not only acquired SCFs but also existing SCFs in the target grammar. We then eliminate implausible acquired SCFs and add plausible unseen SCFs according to the set of SCFs represented by the centroids of the resulting clusters.

3.1 Representation of Confidence Values for SCFs

We represent *an SCF confidence-value vector* of each word w_i with a vector v_i, an object for clustering. Each element v_{ij} in v_i represents the confidence value of SCF

s_j for w_i, which expresses how reliable a word w_i has SCF s_j. We should note that the confidence value is not the probability that a word w_i appears with SCF s_j but a probability of existence of SCF s_j for the word w_i. In this study, we assume that a word w_i can have each SCF s_j with a certain (non-zero) probability $\theta_{ij}(= p(s_{ij}|w_i) > 0$ where $\sum_j \theta_{ij} = 1)$, but only SCFs whose probabilities exceed a certain threshold are recognized as SCFs for the word in the lexicon. We hereafter call this threshold *recognition threshold*. Figure 2 exemplifies a probability distribution of SCFs for *apply*. In this context, we can regard a confidence value of each SCF as the possibility that a probability of a SCF exceeds the recognition threshold.

One intuitive way to estimate a confidence value is to assume an observed probability, *i.e.*, relative frequency, is equal to a probability θ_{ij} of SCF s_j for a word w_i ($\theta_{ij} = freq_{ij}/\sum_j freq_{ij}$ where $freq_{ij}$ is a frequency count that a word w_i have the SCF s_j in corpora[1]). We simply assign 1 to a confidence value $conf_{ij}$ when the relative frequency of s_j for a word w_i exceeds the recognition threshold, and otherwise assign 0 to a confidence value of $conf_{ij}$. However, an observed probability is totally unreliable for infrequent words. For example, when we use a confidence value derived from a relative frequency as above, we cannot distinguish cases where a word w_1 appears once with a SCF s_j and a word w_2 appears 100 times, always with the SCF s_j, which are both the relative frequency 1. Moreover, even when we would like to encode confidence values of reliable SCFs in the target lexicalized grammar, it is also problematic to distinguish the confidence value of those SCFs with confidence values of acquired SCFs.

The other promising way to estimate a true probability θ_{ij} is to regard it as a stochastic variable in the context of Bayesian statistics (Gelman et al., 1995). In this context, *a posteriori* distribution of the probability θ_{ij} of a SCF s_j for a word w_i is given by:

$$p(\theta_{ij}|D) = \frac{P(\theta_{ij})P(D|\theta_{ij})}{P(D)} = \frac{P(\theta_{ij})P(D|\theta_{ij})}{\int_0^1 P(\theta_{ij})P(D|\theta_{ij})d\theta_{ij}}, \quad (1)$$

where $P(\theta_{ij})$ is *a priori* distribution, and D is the data we have observed. Since every occurrence of SCFs in the data D is independent with each other, the data D can be regarded as Bernoulli trials in this case. When we observe the data D that a word w_i appears n times and has SCF s_j $x(\leq n)$ times, its conditional distribution is therefore represented by binominal distribution:

$$P(D|\theta_{ij}) = \binom{n}{x} \theta_{ij}^x (1-\theta_{ij})^{(n-x)}. \quad (2)$$

To calculate this *a posteriori* distribution, we need to define the *a priori* distribution $P(\theta_{ij})$. The question is which probability distribution of θ_{ij} can appropriately reflect prior knowledge. In other words, it should encode knowledge we use to estimate SCFs for an unknown word w_i. We simply determine it from distributions of probability values of s_j for known words. We use distributions of observed probability values of s_j for all words acquired from the corpus by using a method described in (Tsuruoka and Chikayama, 2001). In their study, they assume *a priori* distribution as the *beta* distribution defined as:

$$p(\theta_{ij}|\alpha,\beta) = \frac{\theta_{ij}^{\alpha-1}(1-\theta_{ij})^{\beta-1}}{B(\alpha,\beta)}, \quad (3)$$

where $B(\alpha,\beta) = \int_0^1 \theta_{ij}^{\alpha-1}(1-\theta_{ij})^{\beta-1}d\theta_{ij}$. The value of α and β is determined by moment estimation.[2] By substituting Equations 2 and 3 into Equation 1, we finally obtain the *a posteriori* distribution $p(\theta_{ij}|D)$ as:

$$p(\theta_{ij}|\alpha,\beta,D) = \frac{\frac{\theta_{ij}^{\alpha-1}(1-\theta_{ij})^{\beta-1}}{B(\alpha,\beta)}\binom{n}{x}\theta_{ij}^x(1-\theta_{ij})^{(n-x)}}{\int_0^1 P(\theta_{ij})P(D|\theta_{ij})d\theta_{ij}} = c\cdot\theta_{ij}^{x+\alpha-1}(1-\theta_{ij})^{n-x+\beta-1} \quad (4)$$

where $c = \binom{n}{x}/(B(\alpha,\beta)\int_0^1 P(\theta_{ij})P(D|\theta_{ij})d\theta_{ij})$.

When we determine the value of the recognition threshold as t, we can calculate a confidence value $conf_{ij}$ that a word w_i can have s_j by integrating the *a posteriori* distribution $p(\theta_{ij}|D)$ from the threshold t to 1:

$$conf_{ij} = \int_t^1 c\cdot\theta_{ij}^{x+\alpha-1}(1-\theta_{ij})^{n-x+\beta-1}d\theta_{ij} \quad (5)$$

By using this confidence value, we can express an SCF confidence-value vector v_i for a word w_i in the acquired SCF lexicon ($v_{ij} = conf_{ij}$).[3]

In order to combine SCF confidence-value vectors for words acquired from corpora and those for words in the

[1] We used values of `FREQCNT` to obtain frequency counts of SCFs.

[2] The expectation value and variance of the beta distribution are made equal to those of the observed probability values.

[3] By using the fact that $\int_0^1 P(\theta_{ij}|\alpha,\beta) = 1$, we can calculate $conf_{ij}$ as follows.

$$conf_{ij} = \frac{\int_t^1 c\cdot\theta_{ij}^{x+\alpha-1}(1-\theta_{ij})^{n-x+\beta-1}d\theta_{ij}}{\int_0^1 c\cdot\theta_{ij}^{x+\alpha-1}(1-\theta_{ij})^{n-x+\beta-1}d\theta_{ij}} = \frac{\int_t^1 \theta_{ij}^{x+\alpha-1}(1-\theta_{ij})^{n-x+\beta-1}d\theta_{ij}}{\int_0^1 \theta_{ij}^{x+\alpha-1}(1-\theta_{ij})^{n-x+\beta-1}d\theta_{ij}} \quad (6)$$

```
Input:  a set of SCF confidence-value
          vectors V = {v_1, v_2, ..., v_n} ⊆ R^m
        a distance function d : R^m × Z^m → R
        a function to compute a centroid
          μ : {v_j1, v_j2, ..., v_jl} → R^m
Output: a set of clusters C_j

while cluster members are not stable do
  foreach cluster C_j
    C_j = {v_i | ∀c_l, d(v_i, c_j) ≤ d(v_i, c_l)}
  end foreach
  foreach clusters C_j
    c_j = μ(C_j)
  end foreach
end while

return C_j
```

Figure 3: Clustering algorithm for SCF confidence-value distributions

lexicon of the target grammar, we also represent SCF confidence-value vectors for the words in the target grammars. In this paper, we express SCF confidence-value vectors v'_i for words in the SCF lexicon of the target grammar by:

$$v'_{ij} = \begin{cases} 1-\varepsilon & w_i \textit{ has } s_j \textit{ in the lexicon} \\ \varepsilon & \textit{otherwise} \end{cases} \quad (7)$$

where ε expresses an unreliability of the lexicon. In this study, we simply set it to the machine epsilon. In other words, we trust the lexicon as much as possible.

3.2 Clustering Algorithm for SCF Confidence-Value Distributions

We next present a k-Means-like clustering algorithm for SCF confidence-value vectors, as shown in Figure 3. Given an initial assignment of data objects to k clusters, our algorithm computes a representative value of each cluster called *centroids*. Our algorithm then iteratively updates clusters by assigning each object to its closest centroid and recomputing centroids until cluster members become stable.

Although our algorithm is roughly based on the k-Means algorithm, it is different in an important respect. We define the elements of the centroid values of the obtained clusters as a discrete value of 0 or 1 because we want to obtain clusters which include words that have the exactly same set of SCFs. We then derive a distance function d to calculate the distance from a data object v_i to each centroid c_m. Since the distance function is used to determine the closest cluster for v_i, we define the function d to output the probability that v_i has the SCF set expressed by centroid c_m as follows:

$$d(v_i, c_m) = \prod_{c_{mj}=1} v_{ij} \cdot \prod_{c_{mj}=0} (1-v_{ij}). \quad (8)$$

By using this function, we can determine the closest cluster as $\underset{C_m}{\operatorname{argmax}}\, d(v_i, c_m)$.

After every assignment, we determine a next centroid c_m of each cluster C_m as follows:

$$c_{mj} = \begin{cases} 1 & \textit{when } \prod_{v_i \in C_m} v_{ij} > \prod_{v_i \in C_m} (1-v_{ij}) \\ 0 & \textit{otherwise} \end{cases} \quad (9)$$

We then address the way to determine the number of clusters and initial assignments of objects. In this paper, we assume that the most of the possible set of SCFs for words are included in the target lexicalized grammar, and make use of the existing sets of SCFs for the words in the lexicon of the target grammar to determine the possible set of SCFs for words out of the lexicon. We first extract SCF confidence-value vectors from the lexicon of the target grammar by regarding $\varepsilon = 0$ in Equation 7. By eliminating duplications from them, we obtain SCF centroid-value vectors c_m. We then initialize the number of clusters k to the number of c_m and use them as initial centroids.[4]

We finally update the acquired SCFs using each element's value in the centroid of each cluster and the confidence value of SCFs in this order. We first eliminate SCF s_j for w_i in a cluster m when the value c_{mj} of the centroid c_m is 0, and add SCF s_j for w_i in a cluster m when the value c_{mj} of the centroid c_m is 1. This is because c_{mj} represents whether the words in that class can have SCF s_j. We then eliminate implausible SCFs s_j for w_i from the resulting SCFs according to its corresponding confidence value $conf_{ij}$. We call this elimination *centroid cut-off*. In the following experiments, we compare this cut-off with naive *frequency cut-off*, which uses only relative frequencies to eliminate SCFs and *confidence cut-off*, which uses only confidence values to eliminate SCFs. Note that frequency cut-off and confidence cut-off use only corpus-based statistics to eliminate SCFs.

4 Experiments

We applied our method to an SCF lexicon acquired from 135,902 sentences of the mobile phone news group archived by Google.com, which is the same data used in (Carroll and Fang, 2004). The number of the resulting SCFs is 14,783 for 3,864 word stems. We then translated them to an SCF lexicon for the XTAG English grammar (XTAG Research Group, 2001) by using a translation map manually defined by Ted Briscoe. It defines a mapping from 23 out of 163 possible SCF types into 13 out of 57 XTAG SCFs called *tree families* listed in Table 1. The number of resulting SCFs for the XTAG English grammar was 6,742 for 2,860 word stems.

[4] When a lexicon of the grammar is not comprehensive or less accurate, we should determine the number of clusters using other algorithms (Bischof et al., 1999; Hamerly, 2003).

Table 1: Tree families of the XTAG English grammar mapped from 23 out of 163 SCF types

Tree family	Explanation
Tnx0Ax1	Adjective small clause
Tnx0Vnx1	Transitive
Tnx0Vs1	Sentential complement
Tnx0Vnx2nx1	Ditransitive
Tnx0Vnx1Pnx2	Multiple anchor ditransitive with PP
Tnx0Vnx1pnx2	Ditransitive with PP
Tnx0Vplnx1	Transitive verb Particle
Tnx0Vpl	Intransitive verb Particle
Tnx0Vnx1s2	Sentential complement with NP
Tnx0Vpnx1	Intransitive with PP
Ts0Vnx1	Transitive sentential subject
Tnx0Vax1	Intransitive with adjective
Tnx0Vplnx2nx1	Ditransitive verb Particle

In order to evaluate our method, we split the SCF lexicon of the XTAG English grammar into the training portion and the test portion. The training portion includes 9,427 SCFs for 8,399 words, while the test portion includes 433 SCFs for 280 words The test portion is selected from the SCF lexicon for words that are observed in the acquired SCF lexicon. We extract SCF confidence-value vectors from the training portion and combine them with the SCF confidence-value vectors obtained from the acquired SCFs. The number of the resulting data objects is 8,679.[5] We also make use of the SCF confidence-value vectors obtained from the training SCF lexicon as an initial centroid by regarding ε as 0. The total number of them was 35.[6] We then performed clustering of these 8,679 data objects into 35 clusters.

We finally evaluate precision and recall of the resulting SCFs by comparing them with the test SCF lexicon of the XTAG English grammar.

We first compare confidence cut-off with frequency cut-off to investigate effects of Bayesian estimation. Figure 4 shows precision and recall of the resulting SCF sets using confidence cut-off and frequency cut-off. We measured precision and recall of the SCF sets obtained using confidence cut-off whose recognition threshold $t = 0.01$ (confidence cut-off 0.01), 0.03 (confidence cut-off 0.03), and 0.05 (confidence cut-off 0.05) by varying threshold for the confidence value from 0 to 1. We also measured those for the SCF sets obtained using frequency cut-off by varying threshold for the relative frequency from 0 to 1. The graph apparently indicates that the confidence cut-offs outperformed the frequency cut-off. When we compare confidence cut-offs with different recognition thresholds, we can improve precision using higher recognition threshold while we can improve recall using lower recognition threshold. This result is quite consistent with our expectations.

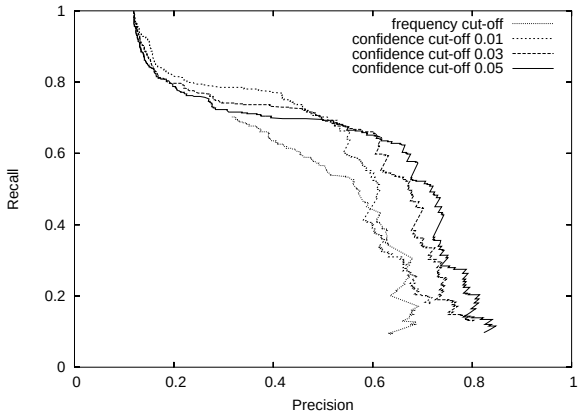

Figure 4: Precision and recall of the resulting SCFs using confidence cut-off and frequency cut-off

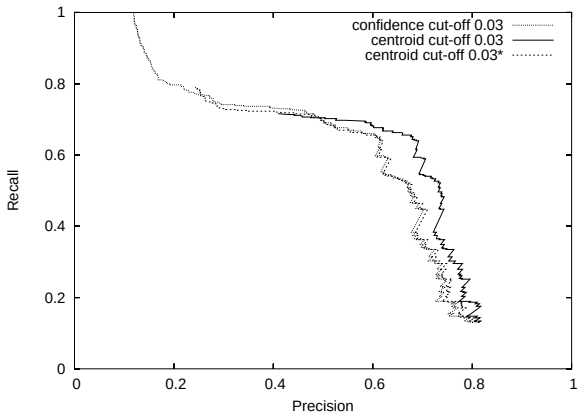

Figure 5: Precision and recall of the resulting SCFs using confidence cut-off and frequency cut-off

We then compare centroid cut-off with confidence cut-off to observe effects of clustering using information in the lexicon of the XTAG English grammar. Figure 5 shows precision and recall of the resulting SCF sets using centroid cut-off and confidence cut-off with the recognition threshold $t = 0.03$ by varying the threshold for the confidence value. In order to show the effects of information of the training SCF lexicon, centroid cut-off 0.03* is SCFs obtained by clustering of SCF confidence-value vectors in the acquired SCFs only with random initialization. The graph apparently shows that clustering is meaningful only when we make use of the reliable SCF confidence-value vectors obtained from the manually tai-

[5] We used the SCF confidence-value vectors for words which are included in the XTAG English grammar. When both the training SCF lexicon and the acquired SCF lexicon have the same words, we simply used an SCF confidence-value vector obtained from the acquired SCF lexicon.

[6] We used the SCF confidence-value vectors that appear with more than two words.

SCF	# SCFs	frequency cut-off				confidence cut-off 0.03				centroid cut-off 0.03			
		Precision		Recall		Precision		Recall		Precision		Recall	
Tnx0Ax1	12(1)	na	(0/0)	0.000	(0/12)	na	(0/0)	0.000	(0/12)	na	(0/0)	0.000	(0/12)
Tnx0Vnx1	267(222)	0.959	(212/221)	0.794	(212/267)	0.958	(253/264)	0.948	(253/267)	0.956	(260/272)	0.974	(260/267)
Tnx0Vs1	38(29)	0.357	(10/28)	0.263	(10/38)	0.381	(8/21)	0.211	(8/38)	0.323	(10/31)	0.263	(10/38)
Tnx0Vnx2nx1	21(16)	0.105	(6/57)	0.286	(6/21)	0.185	(10/54)	0.476	(10/21)	0.122	(9/74)	0.429	(9/21)
Tnx0Vnx1Pnx2	8(4)	0.200	(3/15)	0.375	(3/8)	0.200	(2/10)	0.250	(2/8)	0.250	(2/8)	0.250	(2/8)
Tnx0Vnx1pnx2	5(1)	0.024	(1/41)	0.200	(1/5)	0.029	(1/34)	0.200	(1/5)	na	(0/0)	0.000	(0/5)
Tnx0Vplnx1	40(23)	0.538	(7/13)	0.175	(7/40)	0.667	(6/9)	0.150	(6/40)	0.778	(7/9)	0.175	(7/40)
Tnx0Vpl	20(0)	na	(0/0)	0.000	(0/20)	na	(0/0)	0.000	(0/20)	na	(0/0)	0.000	(0/20)
Tnx0Vnx1s2	11(6)	0.083	(1/12)	0.091	(1/11)	0.200	(1/5)	0.091	(1/11)	0.200	(1/5)	0.091	(1/11)
Ts0Vnx1	8(1)	0.000	(0/2)	0.000	(0/8)	na	(0/0)	0.000	(0/8)	na	(0/0)	0.000	(0/8)
Tnx0Vax1	2(1)	0.000	(0/9)	0.000	(0/2)	0.000	(0/3)	0.000	(0/2)	0.000	(0/1)	0.000	(0/2)
Tnx0Vplnx2nx1	1(0)	0.000	(0/2)	0.000	(0/1)	na	(0/0)	0.000	(0/1)	na	(0/0)	0.000	(0/1)

Table 2: Precision and recall for 400 SCFs obtained from freqency cut-off, confidence cut-off 0.03, and centroid cut-off 0.03

lored lexicon. The centroid cut-off using the lexicon boosted precision and recall compared to the confidence cut-off and the centroid cut-off without the lexicon.

We finally investigate precision and recall of the resulting SCFs for every SCF type in order to evaluate effects of our method on each SCF. Table 2 shows precision and recall of the SCFs by using frequency cut-off (the threshold for the relative frequency 0.092), confidence cut-off 0.03 (the threshold for the confidence value 0.953), centroid cut-off 0.03 (the threshold for the confidence value 0.889)[7] by using thresholds for the relative frequency and the confidence value that preserve exactly 400 SCFs. The numbers in curly brackets in # of SCFs colum show the number of SCFs in the test SCF lexicon that are acquired from the training corpus. The left and right numbers in curly brackets in the precision columns show the number of correct SCFs against all SCFs in the resulting SCF lexicon while those in the recall columns show the number of correct SCFs against all SCFs in the test SCF lexicon. We can observe a tendency that the confidence cut-off and the centroid cut-off preserve more transitive (Tnx0Vnx1) SCF. This is because some SCFs of Tnx0Vnx1 in the test SCF lexicon are not observed in the training corpus but are predicted by *a priori* distribution for SCF Tnx0Vnx1. Also, the centroid cut-off tends to reduce implausible SCFs of Tnx0Vnx1Pnx2 and Tnx0Vax1. Since the threshold for the confidence value of the centroid cut-off 0.03 (0.889) is smaller than that of the confidence cut-off 0.03 (0.953), the clustering could eliminate implausible SCFs without reducing recall.

In short, one reason why the centroid cut-off outperforms the confidence cut-off (or the frequency cut-off) is due to the way how the centroid cut-off eliminate SCFs not existed in the lexicon. When we eliminate SCFs with lower relative frequency under the assumption that those SCFs tend to be wrongly acquired SCFs, it must also eliminate correct SCFs with low relative frequencies. By using co-occurrence tendency among SCFs as another criteria to judge the implausibility of the SCFs, we can eliminate more wrongly acquired SCFs because they tend to violate the co-occurrence tendency. Another reason why the centroid cut-off and the confidence cut-off outperform the the frequency cut-off is due to the way how those cut-offs add new unseen SCFs. We can add plausible SCFs from those SCFs which is reliable according to their *a priori* distribution. Furthermore, since the centroid cut-off makes use of the co-occurrence tendency among SCFs, it adds only SCFs which are plausible in terms of corpus-based statistics (confidence value) under the restriction provided by the co-occurrence tendency among SCFs in the lexicon of the target grammar.

[7] Since no word takes SCF Tnx0Vpnx1 in the test SCF lexicon, we omit it here.

5 Concluding Remarks and Future Work

In this paper, we presented a novel way to improve the quality of SCFs acquired from corpora in order to augment a lexicalized grammar with them. By applying our method to the acquired SCF lexicon using the XTAG English grammar, we showed that our method improved both precision and recall of the resulting SCFs compared to the naive frequency-based cut-off.

In future work, we are going to investigate the parsing performance of the XTAG English grammar augmented with SCFs obtained by our method. We will apply our method to lexicalized grammars with relatively smaller lexicon, *e.g.*, the LINGO English Resource Grammar (Flickinger, 2000).

Acknowledgment

The authors wish to thank Yoshimasa Tsuruoka and Takuya Matsuzaki for their advice on probabilistic modeling of the set of SCFs, and thank Alex Fang for his help in using SCFs acquired from the corpus. The authors are also indebted to Yusuke Miyao, John Carroll and the three anonymous reviewers for their valuable comments on this paper. The first author was supported in part by JSPS Research Fellowships for Young Scientists.

References

Horst Bischof, Ales Leonardis, and Alexander Selb. 1999. MDL principle for robust vector quantization. *Pattern Analysis and Applications*, 2(1):59–72.

Branimir Boguraev and Ted Briscoe. 1987. Large lexicons for natural language processing: utilising the grammar coding system of LDOCE. *Computational Linguistics*, 13(4):203–218.

Michael R. Brent. 1993. From grammar to lexicon. *Computational Linguistics*, 19(2):243–262.

Ted Briscoe and Jhon Carroll. 1997. Automatic extraction of subcategorization from corpora. In *Proc. of the fifth ANLP*, pages 356–363.

Jhon Carroll and Alex C. Fang. 2004. The automatic acquisition of verb subcategorizations and their impact on the performance of an HPSG parser. In *Proc. of the first ijc-NLP*, pages 107–114.

Dan Flickinger. 2000. On building a more efficient grammar by exploiting types. *Natural Language Engineering*, 6(1):15–28.

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

Andrew Gelman, John B. Carlin, Hal S. Stern, and Donald B. Rubin, editors. 1995. *Bayesian Data Analysis*. Chapman and Hall.

Ralph Grishman, Catherine Macleod, and Adam Meyers. 1994. Comlex syntax: Building a computational lexicon. In *Proc. of the 15th COLING*, pages 268–272.

Greg Hamerly. 2003. *Learning structure and concepts in data through data clustering*. Ph.D. thesis, University of Californila, San Diego.

Anna Korhonen, Yuval Krymolowski, and Zvika Marx. 2003. Clustering polysemic subcategorization frame distributions semantically. In *Proc. of the 41st ACL*, pages 64–71.

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

Christopher D. Manning. 1993. Automatic acquisition of a large subcategorization dictionary from corpora. In *Proc. of the 31st ACL*, pages 235–242.

Anoop Sarkar and Daniel Zeman. 2000. Automatic extraction of subcategorization frames for Czech. In *Proc. of 18th COLING*, pages 691–697.

Anoop Sarkar, Fei Xia, and Aravind K. Joshi. 2000. Some experiments on indicators of parsing complexity for lexicalized grammars. In *Proc. of the 18th COLING workshop*, pages 37–42.

Yves Schabes, Anne Abeillé, and Aravind K. Joshi. 1988. Parsing strategies with 'lexicalized' grammars: application to Tree Adjoining Grammars. In *Proc. of the 12th COLING*, pages 578–583.

Sabine Schulte im Walde and Chris Brew. 2002. Inducing German semantic verb classes from purely syntactic subcategorisation information. In *Proc. of the 41st ACL*, pages 223–230.

Naftali Tishby, Fernand C. Pereira, and William Bialek. 1999. The information bottleneck method. In *Proc. of the 37th ACL*, pages 368–377.

Yoshimasa Tsuruoka and Takashi Chikayama. 2001. Estimating reliability of contextual evidences in decision-list classifiers under bayesian learning. In *Proc. of the sixth NLPRS*, pages 701–707.

XTAG Research Group. 2001. A lexicalized Tree Adjoining Grammar for English. Technical Report IRCS-01-03, IRCS, University of Pennsylvania.