Maximum Entropy Model Learning of Subcategorization Preference*

Takehito Utsuro Takashi Miyata Yuji Matsumoto
Graduate School of Information Science,
Nara Institute of Science and Technology
8916-5, Takayama-cho, Ikoma-shi, Nara, 630-01, JAPAN
E-mail: {utsuro,takashi,matsu}@is.aist-nara.ac.jp
URL: http://cactus.aist-nara.ac.jp/staff/utsuro/home-e.html

Abstract

This paper proposes a novel method for learning probabilistic models of subcategorization preference of verbs. Especially, we propose to consider the issues of *case dependencies* and *noun class generalization* in a uniform way. We adopt the maximum entropy model learning method and apply it to the task of model learning of subcategorization preference. Case dependencies and noun class generalization are represented as *features* in the maximum entropy approach. The feature selection facility of the maximum entropy model learning makes it possible to find optimal case dependencies and optimal noun class generalization levels. We describe the results of the experiment on learning probabilistic models of subcategorization preference from the EDR Japanese bracketed corpus. We also evaluated the performance of the selected features and their estimated parameters in the subcategorization preference task.

1 Introduction

In corpus-based NLP, extraction of linguistic knowledge such as lexical/semantic collocation is one of the most important issues and has been intensively studied in recent years. In those research, extracted lexical/semantic collocation is especially useful in terms of ranking parses in syntactic analysis as well as automatic construction of lexicon for NLP.

For example, in the context of syntactic disambiguation, Black (1993) and Magerman (1995) proposed statistical parsing models based-on decision-tree learning techniques, which incorporated not only syntactic but also lexical/semantic information in the decision-trees. As lexical/semantic information, Black (1993) used about 50 semantic categories, while Magerman (1995) used lexical forms of words. Collins (1996) proposed a statistical parser which is based on probabilities of dependencies between head-words in the parse tree. In those works, lexical/semantic collocation are used for ranking parses in syntactic analysis. They put an assumption that syntactic and lexical/semantic features are dependent on each other. In their models, syntactic and lexical/semantic features are combined together, and this causes each parameter to depend on both syntactic and lexical/semantic features.

On the other hand, in the context of automatic lexicon construction, the emphasis is mainly on the extraction of lexical/semantic collocational knowledge of specific words rather than its use in sentence parsing. For example, Haruno (1995) applied an information-theoretic data compression technique to corpus-based case frame learning, and proposed a method of finding case frames of verbs as compressed representation of verb-noun collocational data in corpus. The work concentrated on the extraction of declarative representation of case frames and did not consider their performance in sentence parsing.

*The authors would like to thank Dr. Kentaro Inui and Mr. Kiyoaki Shirai of Tokyo Institute of Technology for valuable information on implementing maximum entropy model learning. This research was partially supported by the Ministry of Education, Science, Sports and Culture, Japan, Grant-in-Aid for Encouragement of Young Scientists, 09780338, 1997.

As in the case of the models of Black (1993), Magerman (1995), and Collins (1996), this paper proposes a method of utilizing lexical/semantic features for the purpose of applying them to ranking parses in syntactic analysis. However, unlike the models of Black (1993), Magerman (1995), and Collins (1996), we put an assumption that syntactic and lexical/semantic features are independent. Then, we focus on extracting lexical/semantic collocational knowledge of verbs which is useful in syntactic analysis.

More specifically, we propose a novel method for learning a probabilistic model of subcategorization preference of verbs. In general, when learning lexical/semantic collocational knowledge of verbs from corpus, it is necessary to consider the following two issues:

1) *Case dependencies*
2) *Noun class generalization*

When considering 1), we have to decide which cases are dependent on each other and which cases are optional and independent of other cases. When considering 2), we have to decide which superordinate class generates each observed leaf class in the verb-noun collocation.

So far, there exist several researches which worked on these two issues in learning collocational knowledge of verbs and also evaluated the results in terms of syntactic disambiguation. Resnik (1993) and Li and Abe (1995) studied how to find an optimal abstraction level of an argument noun in a tree-structured thesaurus. Although they evaluated the obtained abstraction level of the argument noun by its performance in syntactic disambiguation, their works are limited to only one argument. Li and Abe (1996) also studied a method for learning dependencies between case slots and evaluated the discovered dependencies in the syntactic disambiguation task. They first obtained optimal abstraction levels of the argument nouns by the method in Li and Abe (1995), and then tried to discover dependencies between the class-based case slots. They reported that dependencies were discovered only at the slot-level and not at the class-level.

Compared with those previous works, this paper proposes to consider the above two issues in a uniform way. First, we introduce a model of generating a collocation of a verb and argument/adjunct nouns and then view the model as a probabilistic model. As a model learning method, we adopt the maximum entropy model learning method (Della Pietra, Della Pietra, and Lafferty, 1997; Berger, Della Pietra, and Della Pietra, 1996) and apply it to the task of model learning of subcategorization preference. *Case dependencies* and *noun class generalization* are represented as *features* in the maximum entropy approach. In the maximum entropy approach, features are allowed to have overlap and this is quite advantageous when we consider case dependencies and noun class generalization in parameter estimation. The feature selection facility of the maximum entropy model learning method also makes it possible to find optimal set of features, i.e, optimal case dependencies and optimal noun class generalization levels. We introduce several different models according to the difference of case dependencies. We describe the results of the experiment on learning models of subcategorization preference from the EDR Japanese bracketed corpus (EDR, 1995). We also evaluate the performance of the selected features and their estimated parameters in the subcategorization preference task.

2 A Model of Generating a Verb-Noun Collocation from Subcategorization Frame(s)

This section introduces a model of generating a verb-noun collocation from subcategorization frame(s).

2.1 Data Structure

2.1.1 Verb-Noun Collocation

Verb-noun collocation is a data structure for the collocation of a verb and all of its argument/adjunct nouns. A verb-noun collocation e is represented by a feature structure which consists of the verb v and all the pairs of co-occurring case-markers p and thesaurus classes c of case-marked nouns:

$$e = \begin{bmatrix} pred : v \\ p_1 : c_1 \\ \vdots \\ p_k : c_k \end{bmatrix} \quad (1)$$

We assume that a *thesaurus* is a tree-structured type hierarchy in which each node represents a semantic class, and each thesaurus class $c_1, \ldots, c_k$ in a verb-noun collocation is a leaf class. We also introduce $\preceq_c$ as the superordinate-subordinate relation of classes in a thesaurus: $c_1 \preceq_c c_2$ means that c_1 is subordinate to c_2. [1]

2.1.2 Subcategorization Frame

A *subcategorization frame* s is represented by a feature structure which consists of a verb v and the pairs of case-markers p and sense restriction c of case-marked argument/adjunct nouns:

$$s = \begin{bmatrix} pred : v \\ p_1 : c_1 \\ \vdots \\ p_l : c_l \end{bmatrix} \quad (2)$$

Sense restriction $c_1, \ldots, c_l$ of case-marked argument/adjunct nouns are represented by classes at arbitrary levels of the thesaurus. A subcategorization frame s can be divided into two parts: one is the verbal part s_v containing the verb v while the other is the nominal part s_p containing all the pairs of case-markers p and sense restriction c of case-marked nouns.

$$s = s_v \wedge s_p = \left[pred : v \right] \wedge \begin{bmatrix} p_1 : c_1 \\ \vdots \\ p_l : c_l \end{bmatrix} \quad (3)$$

2.1.3 Subsumption Relation

We introduce *subsumption relation* $\preceq_{sf}$ of a *verb-noun collocation* e and a *subcategorization frame* s:

$e \preceq_{sf} s$ iff. for each case-marker p_i in s and its noun class c_{si}, there exists the same case-marker p_i in e and its noun class c_{ei} is subordinate to c_{si}, i.e. $c_{ei} \preceq_c c_{si}$

The subsumption relation $\preceq_{sf}$ is applicable also as a subsumption relation of two subcategorization frames.

2.2 Generating a Verb-Noun Collocation from Subcategorization Frame(s)

Next, let us consider modeling the generation of a verb-noun collocation from a subcategorization frame. Especially, we describe the basic idea of incorporating *case dependencies* and *noun class generalization* into the model of generating a verb-noun collocation from a subcategorization frame.

Suppose a verb-noun collocation e is given as:

$$e = \begin{bmatrix} pred : v \\ p_1 : c_{e1} \\ \vdots \\ p_k : c_{ek} \end{bmatrix}$$

[1] Although we ignore sense ambiguities of case-marked nouns in the definitions of this section, in the current implementation, we deal with sense ambiguities of case-marked nouns by deciding that a class c is superordinate to an ambiguous leaf class C_l if c is superordinate to at least one of the possible unambiguous classes of C_l.

Then, we consider a subcategorization frame s which can generate e and assume that s subsumes e:

$$e \preceq_{sf} s$$

We denote the generation of the verb-noun collocation e from the subcategorization frame s as:

$$s \longrightarrow e \tag{4}$$

2.2.1 Case Dependencies

When considering a subcategorization frame which can generate a verb-noun collocation e, there are several possibilities of the case dependencies in the subcategorization frame.

For example, consider the following example:

Example 1

Kodomo-ga kouen-de juusu-wo nomu.
child-NOM park-at juice-ACC drink
(A child drinks juice at the park.)

The verb-noun collocation is represented as a feature structure e below:

$$e = \begin{bmatrix} pred : nomu \\ ga : c_c \\ wo : c_j \\ de : c_p \end{bmatrix} \tag{5}$$

In this feature structure e, c_c, c_p, and c_j represent the leaf classes (in the thesaurus) of the nouns *"kodomo(child)"*, *"kouen(park)"*, and *"juusu(juice)"*.

Next, we assume that the concepts *"human"*, *"place"*, and *"beverage"* are superordinate to *"kodomo(child)"*, *"kouen(park)"*, and *"juusu(juice)"*, respectively, and introduce the corresponding classes c_{hum}, c_{plc}, and c_{bev}. Then, the following superordinate-subordinate relations hold:

$$c_c \preceq_c c_{hum}, \quad c_p \preceq_c c_{plc}, \quad c_j \preceq_c c_{bev}$$

Allowing these superordinate classes as sense restriction in subcategorization frames, let us consider several patterns of subcategorization frames each of which can generate the verb-noun collocation e. Those patterns of subcategorization frames vary according to the dependencies of cases within them.

If the three cases *"ga(NOM)"*, *"wo(ACC)"*, and *"de(at)"* are dependent on each other and it is not possible to find any division into several independent subcategorization frames, e can be regarded as generated from a subcategorization frame containing all of the three cases:

$$\begin{bmatrix} pred : nomu \\ ga : c_{hum} \\ wo : c_{bev} \\ de : c_{plc} \end{bmatrix} \longrightarrow e \tag{6}$$

Otherwise, if only the two cases *"ga(NOM)"* and *"wo(ACC)"* are dependent on each other and the *"de(at)"* case is independent of those two cases, e can be regarded as generated from the following two subcategorization frames independently:

$$\begin{bmatrix} pred : nomu \\ ga : c_{hum} \\ wo : c_{bev} \end{bmatrix} \longrightarrow e, \quad \begin{bmatrix} pred : nomu \\ de : c_{plc} \end{bmatrix} \longrightarrow e \tag{7}$$

Otherwise, if all the three cases *"ga(NOM)"*, *"wo(ACC)"*, and *"de(at)"* are independent of each other, e can be regarded as generated from the following three subcategorization frames independently, each of which contains only one case:

$$\begin{bmatrix} pred : nomu \\ ga : c_{hum} \end{bmatrix} \longrightarrow e, \quad \begin{bmatrix} pred : nomu \\ wo : c_{bev} \end{bmatrix} \longrightarrow e, \quad \begin{bmatrix} pred : nomu \\ de : c_{plc} \end{bmatrix} \longrightarrow e \tag{8}$$

2.2.2 Noun Class Generalization

In the similar way, when considering a subcategorization frame which can generate a verb-noun collocation e, there are several possibilities of the noun class generalization levels as the sense restrictions of the case-marked nouns.

For example, let us again consider Example 1. We assume that the concepts *"animal"* and *"liquid"* are superordinate to *"human"* and *"beverage"*, respectively, and introduce the corresponding classes c_{ani} and c_{liq}. Then, the following superordinate-subordinate relations hold:

$$c_{hum} \preceq_c c_{ani}, \quad c_{bev} \preceq_c c_{liq}$$

If we additionally allow these superordinate classes as sense restriction in subcategorization frames, we can consider several additional patterns of subcategorization frames which can generate the verb-noun collocation e, along with those patterns described in the previous section.

Suppose that only the two cases *"ga(NOM)"* and *"wo(ACC)"* are dependent on each other and the *"de(at)"* case is independent of those two cases as in the formula (7). Since the leaf class c_c (*"child"*) can be generated from either c_{hum} or c_{ani}, and also the leaf class c_j (*"juice"*) can be generated from either c_{bev} or c_{liq}, e can be regarded as generated according to either of the four formulas (the left-side formula of) (7) and (9):

$$\begin{bmatrix} pred : nomu \\ ga : c_{ani} \\ wo : c_{bev} \end{bmatrix} \longrightarrow e\,, \quad \begin{bmatrix} pred : nomu \\ ga : c_{hum} \\ wo : c_{liq} \end{bmatrix} \longrightarrow e\,, \quad \begin{bmatrix} pred : nomu \\ ga : c_{ani} \\ wo : c_{liq} \end{bmatrix} \longrightarrow e \qquad (9)$$

2.3 Case Dependencies and the Design of the Generation Models

As we described in the previous section, there are several possibilities of the case dependencies in a verb-noun collocation, and this results in the differences of the subcategorization frames which can generate the given verb-noun collocation. According to the different assumptions on the case dependencies, we can design several different models of generating a verb-noun collocation from subcategorization frame(s).

2.3.1 Partial-Frame Model

First, we put no assumption on the case dependencies in the given verb-noun collocation e, and assume that any subcategorization frame s which subsumes e can generate e.

$$e \preceq_{sf} s$$

With this requirement, the subcategorization frame s does not have to have all the cases in e, but has to have only some part of the cases in e. We call the model satisfying this requirement *the partial-frame model.* All the examples of the formulas (6) and (9) satisfy this requirement and can be regarded as examples of the partial-frame model.

2.3.2 One-Frame Model

Next, in addition to the requirement that s subsumes e, we put another assumption that all the cases in the given verb-noun collocation e are dependent on each other and that a subcategorization frame s which can generate e should have exactly the same cases as e has:

$$e = \begin{bmatrix} pred : v \\ p_1 : c_1 \\ \vdots \\ p_k : c_k \end{bmatrix}, \quad s = \begin{bmatrix} pred : v \\ p_1 : c'_1 \\ \vdots \\ p_k : c'_k \end{bmatrix} \qquad (10)$$

We call the model satisfying this requirement as *the one-frame model.* For example, supposing that the verb-noun collocation e in the equation (5) is given, the example in the formula (6) satisfies this requirement.

2.3.3 Independent-Case Model

In addition to the requirement that s subsumes e, we can also put an assumption that all the cases in the given verb-noun collocation e are independent of each other and that a subcategorization frame s which has only one case of e can generate e:

$$s = \begin{bmatrix} pred : v \\ p_i : c'_i \end{bmatrix} \quad (1 \leq i \leq k)$$

We call the model satisfying this requirement as *the independent-case model*. For example, supposing that the verb-noun collocation e in the equation (5) is given, the examples in the formula (8) satisfy this requirement.

2.3.4 Independent-Frame Model

As can be seen in the definitions of the above three models, the basic idea of defining the model of generating a verb-noun collocation from subcategorization frame(s) lies in identifying the dependencies of the cases in the given verb-noun collocation and expressing the dependencies within a subcategorization frame. Here, we briefly show a method of statistically identifying the dependencies of the cases in verb-noun collocations from corpus.[2] Then, by incorporating the identified case dependencies into the generation model, we introduce a model of generating a verb-noun collocation from a tuple of *independent* partial subcategorization frames. We call this model as *the independent-frame model*.

Partial Subcategorization Frame

Suppose a verb-noun collocation e is given as in the formula (10) and a subcategorization frame s satisfies the requirement of the one-frame model in section 2.3.2, i.e., as in the formula (10), s has exactly the same case-markers as e has, and s subsumes e.

Then, we define a *partial subcategorization frame* s_i of s as a subcategorization frame which has the same verb v as s as well as some of the case-markers of s and their semantic classes. Then, we can find a division of s into a tuple $\langle s_1, \ldots, s_n \rangle$ of partial subcategorization frames of s, where any pair s_i and $s_{i'}$ $(i \neq i')$ do not have common case-markers and the unification $s_1 \wedge \cdots \wedge s_n$ of all the partial subcategorization frames equals to s:

$$s = s_1 \wedge \cdots \wedge s_n, \quad s_i = \begin{bmatrix} pred : v \\ \vdots \\ p_{ij} : c_{ij} \\ \vdots \end{bmatrix}, \quad \begin{array}{l} \forall j \forall j' \; p_{ij} \neq p_{i'j'} \\ (i, i' = 1, \ldots, n, \;\; i \neq i') \end{array} \tag{11}$$

Independence of Partial Subcategorization Frames

The conditional joint probability $p(s_1, \ldots, s_n \mid v)$ is estimated by summing up $p(e \mid v)$ where e is subsumed by all of $s_1, \ldots, s_n$ $(e \preceq_{sf} s_1, \ldots, s_n)$:

$$p(s_1, \ldots, s_n \mid v) \approx \sum_{e \preceq_{sf} s_1, \ldots, s_n} p(e \mid v) \tag{12}$$

Then, we introduce a parameter α $(0 \leq \alpha \leq 1)$ for relaxing the constraint of independence. Partial subcategorization frames $s_1, \ldots, s_n$ are judged as *independent* if, for every subset $s_{i_1}, \ldots, s_{i_j}$ of j of these partial subcategorization frames $(j = 2, \ldots, n)$, the following inequalities hold:

$$\alpha \leq \frac{p(s_{i_1}, \ldots, s_{i_j} \mid v)}{p(s_{i_1} \mid v) \cdots p(s_{i_j} \mid v)} \leq \frac{1}{\alpha} \tag{13}$$

This definition of independence judgment means that the condition on independence judgment becomes weaker as α decreases, while it becomes more strict as α increases.

[2] Details of the method of statistically identifying the dependencies of the cases in verb-noun collocations are given in Utsuro and Matsumoto (1997).

Generation from Independent Partial Subcategorization Frames
Now, we denote the generation of e from a tuple $\langle s_1, \ldots, s_n \rangle$ of independent partial subcategorization frames of s as below:

$$\langle s_1, \ldots, s_n \rangle \longrightarrow e \tag{14}$$

Example
For example, suppose that a verb-noun collocation e is given as in the formula (5) in section 2.2.1. If the three cases in e are dependent on each other as in the generation of e in the formula (6), the generation of e is denoted as below in the case of the independent-frame model:

$$\left\langle \begin{bmatrix} pred : nomu \\ ga : c_{hum} \\ wo : c_{bev} \\ de : c_{plc} \end{bmatrix} \right\rangle \longrightarrow e \tag{15}$$

Otherwise, if only the two cases *"ga(NOM)"* and *"wo(ACC)"* are dependent on each other and the *"de(at)"* case is independent of those two cases as in the generation of e in the formula (7), the generation of e is denoted as below:

$$\left\langle \begin{bmatrix} pred : nomu \\ ga : c_{hum} \\ wo : c_{bev} \end{bmatrix}, \begin{bmatrix} pred : nomu \\ de : c_{plc} \end{bmatrix} \right\rangle \longrightarrow e \tag{16}$$

3 Maximum Entropy Modeling

This section gives a formal description of maximum entropy modeling (Della Pietra, Della Pietra, and Lafferty, 1997; Berger, Della Pietra, and Della Pietra, 1996).

3.1 The Maximum Entropy Principle

We consider a random process that produces an *output* value y, a member of a finite set $\mathcal{Y}$. In generating y, the process may be influenced by some *contextual information* x, a member of a finite set $\mathcal{X}$. Our task is to construct a stochastic model that accurately represents the behavior of the random process. Such a model is a method of estimating the conditional probability that, given a context x, the process will output y. We denote by $p(y \mid x)$ the probability that the model assigns to y in context x. We also denote by $\mathcal{P}$ the set of all conditional probability distributions. Thus a model $p(y \mid x)$ is an element of $\mathcal{P}$.

To study the process, we observe the behavior of the random process by collecting a large number of samples of the *event* (x, y). We can summarize the training sample in terms of its *empirical probability distribution* $\tilde{p}$, defined by:

$$\tilde{p}(x, y) \equiv \frac{freq(x, y)}{\sum_{x,y} freq(x, y)} \tag{17}$$

where $freq(x, y)$ is the number of times that the pair (x, y) occurs in the sample.

Next, in order to express certain features of the whole event (x, y), a binary-valued indicator function is introduced and called a *feature function*. Usually, we suppose that there exists a large collection $\mathcal{F}$ of candidate features, and include in the model only a subset $\mathcal{S}$ of the full set of candidate features $\mathcal{F}$. We call $\mathcal{S}$ the set of *active* features. The choice of $\mathcal{S}$ must capture as much information about the random process as possible, yet only include features whose expected values can be reliably estimated. In this section and the next section, we assume that the set $\mathcal{S}$ of active features can be found in some way. How to find $\mathcal{S}$ will be described in section 3.3.

Now, we assume that $\mathcal{S}$ contains n feature functions. For each feature $f_i(\in \mathcal{S})$, the sets V_{xi} and V_{yi} will be given for indicating the sets of the values of x and y for that feature. According to those sets, each feature function f_i will be defined as follows:

$$f_i(x,y) = \begin{cases} 1 & \text{if } x \in V_{xi} \text{ and } y \in V_{yi} \\ 0 & \text{otherwise} \end{cases} \tag{18}$$

When we discover a feature that we feel is useful, we can acknowledge its importance by requiring that our model accord with the feature's empirical distribution. In maximum entropy modeling approach, this is done by constraining that the expected value of each f_i with respect to the model $p(y \mid x)$ (left-hand side) be the same as that of f_i in the training sample (right-hand side):

$$\sum_{x,y} \tilde{p}(x)p(y \mid x)f_i(x,y) = \sum_{x,y} \tilde{p}(x,y)f_i(x,y) \quad \text{for } \forall f_i \in \mathcal{S} \tag{19}$$

This requirement is called a *constraint equation*. This requirement means that we would like p to lie in the subset of $\mathcal{P}$.

Then, among the possible models p, the philosophy of the maximum entropy modeling approach is that we should select the most uniform distribution. A mathematical measure of the uniformity of a conditional distribution $p(y \mid x)$ is provided by the conditional entropy:

$$H(p) \equiv -\sum_{x,y} \tilde{p}(x)p(y \mid x) \log p(y \mid x) \tag{20}$$

Now, we present the principle of maximum entropy:

Maximum Entropy Principle

> To select a model from a set of allowed probability distributions, choose the model p_* with maximum entropy $H(p)$:
>
> $$p_* = \arg\max_p H(p) \tag{21}$$

3.2 Parameter Estimation

It can be shown that there always exists a unique model p_* with maximum entropy in any constrained set. According to Della Pietra, Della Pietra, and Lafferty (1997) and Berger, Della Pietra, and Della Pietra (1996), the solution can be found as the following $p_\lambda(y \mid x)$ of the form of the exponential family:

$$p_\lambda(y \mid x) = \frac{\exp\left(\sum_i \lambda_i f_i(x,y)\right)}{\sum_y \exp\left(\sum_i \lambda_i f_i(x,y)\right)} \tag{22}$$

where a *parameter* λ_i is introduced for each feature f_i.

Della Pietra, Della Pietra, and Lafferty (1997) and Berger, Della Pietra, and Della Pietra (1996) also presented an optimization method of estimating the parameter values λ^*_i that maximize the entropy, which is called *Improved Iterative Scaling* (IIS) algorithm.

3.3 Feature Selection

Given the full set $\mathcal{F}$ of candidate features, this section outlines how to select an appropriate subset $\mathcal{S}$ of active features. The feature selection process is an incremental procedure that builds up $\mathcal{S}$ by successively adding features. At each step, we select the candidate feature which, when adjoined to the set of active features $\mathcal{S}$, produces the greatest increase in log-likelihood of the training sample:[3]

$$L_{\tilde{p}}(p) \equiv \sum_{x,y} \tilde{p}(x,y) \log p(y \mid x) \tag{23}$$

[3]It is shown in Della Pietra, Della Pietra, and Lafferty (1997) and Berger, Della Pietra, and Della Pietra (1996) that the model p_* with maximum entropy $H(p)$ is the model in the parametric family $p_\lambda(y \mid x)$ of the formula (22) that maximizes the likelihood of the training sample $\tilde{p}$.

4 Maximum Entropy Model Learning of Subcategorization Preference

This section describes how to apply the maximum entropy modeling approach to the task of model learning of subcategorization preference.

4.1 Events

In our task of model learning of subcategorization preference, each *event* (x, y) in the training sample is a verb-noun collocation e, which is defined as in the formula (1). As well as a subcategorization frame, a verb-noun collocation e can be divided into two parts: one is the verbal part e_v containing the verb v while the other is the nominal part e_p containing all the pairs of case-markers p and thesaurus leaf classes c of case-marked nouns:

$$e = e_v \wedge e_p = \left[\; pred : v \; \right] \wedge \begin{bmatrix} p_1 : c_1 \\ \vdots \\ p_k : c_k \end{bmatrix}$$

Then, we define the *context* x of an event (x, y) as the verb v and the *output* y as the nominal part e_p of e, and each event in the training sample is denoted as (v, e_p):

$$x \equiv v, \quad y \equiv e_p$$

4.2 Features

Each (partial) subcategorization frame is represented as a *feature* in the maximum entropy modeling approach. In the case of the partial-frame/one-frame/independent-case models in the sections 2.3.1 $\sim$ 2.3.3, a binary-valued feature function $f_s(v, e_p)$ is defined for each subcategorization frame s. In the case of the independent-frame model in section 2.3.4, a binary-valued feature function $f_{s_i}(v, e_p)$ is defined for each partial subcategorization frames s_i in the tuple of the formula (14). Each feature function f has its own parameter λ, which is also the parameter of the corresponding (partial) subcategorization frame. According to the possible variations of case dependencies and noun class generalization, we consider every possible patterns of subcategorization frames which can generate a verb-noun collocation, and then construct the full set $\mathcal{F}$ of candidate features.

In the following, we give formal definitions of the features in each of the partial-frame/one-frame/independent-case/independent-frame models which we introduced in section 2.3.

4.2.1 Partial-Frame Model

Each feature function corresponds to a subcategorization frame s. For each subcategorization frame s, a binary-valued feature function $f_s(v, e_p)$ is defined to be true if and only if the given verb-noun collocation e is subsumed by s:

$$f_s(v, e_p) = \begin{cases} 1 & \text{if } e{=}([pred : v] \wedge e_p) \preceq_{sf} s \\ 0 & \text{otherwise} \end{cases}$$

4.2.2 One-Frame Model

Each feature function corresponds to a subcategorization frame s which has exactly the same cases as the given verb-noun collocation e has. For each subcategorization frame s, a binary-valued feature function $f_s(v, e_p)$ is defined to be true if and only if the given verb-noun collocation e has exactly the same cases as s has and is also subsumed by s:

$$e = \begin{bmatrix} pred : v \\ p_1 : c_1 \\ \vdots \\ p_k : c_k \end{bmatrix}, \quad s = \begin{bmatrix} pred : v \\ p_1 : c'_1 \\ \vdots \\ p_k : c'_k \end{bmatrix}, \quad f_s(v, e_p) = \begin{cases} 1 & \text{if } e{=}([pred : v] \wedge e_p) \preceq_{sf} s \\ 0 & \text{otherwise} \end{cases}$$

4.2.3 Independent-Case Model

Each feature function corresponds to a subcategorization frame s which has only one case of the given verb-noun collocation e. For each subcategorization frame s which has only one case, a binary-valued feature function $f_s(v, e_p)$ is defined to be true if and only if the given verb-noun collocation e has the same case and is also subsumed by s:

$$e = \begin{bmatrix} pred : v \\ p_1 : c_1 \\ \vdots \\ p_k : c_k \end{bmatrix}, \quad s = \begin{bmatrix} pred : v \\ p_i : c'_i \end{bmatrix} \quad (1 \leq i \leq k), \quad f_s(v, e_p) = \begin{cases} 1 & \text{if } e = ([pred : v] \wedge e_p) \preceq_{sf} s \\ 0 & \text{otherwise} \end{cases}$$

4.2.4 Independent-Frame Model

Each feature function corresponds to a partial subcategorization frames s_i in the tuple of independent partial subcategorization frames which can generate the given verb-noun collocation. First, for the given verb-noun collocation e, tuples of independent partial subcategorization frames which can generate e are collected into the set $SF(e)$ as below:[4] [5]

$$SF(e) = \left\{ \langle s_1, \ldots, s_n \rangle \middle| \langle s_1, \ldots, s_n \rangle \longrightarrow e \right\}$$

Then, for each partial subcategorization frame s, a binary-valued feature function $f_s(v, e_p)$ is defined to be true if and only if at least one element of the set $SF(e)$ is a tuple $\langle s_1, \ldots, s, \ldots, s_n \rangle$ that contains s:

$$f_s(v, e_p) = \begin{cases} 1 & \text{if } \exists \langle s_1, \ldots, s, \ldots, s_n \rangle \in SF(e = ([pred : v] \wedge e_p)) \\ 0 & \text{otherwise} \end{cases} \tag{24}$$

4.3 Parameter Estimation

Let $\mathcal{E}$ be the training corpus consisting of training events of the form (v, e_p). Let $\mathcal{F}$ be the full set of candidate features each element of which corresponds to a possible subcategorization frame. Then, given the empirical distribution $\tilde{p}(v, e_p)$ of the training sample, the set $\mathcal{S}(\subseteq \mathcal{F})$ of active features is found according to the feature selection algorithm in section 3.3, and the parameters of subcategorization frames are estimated according to IIS Algorithm(Della Pietra, Della Pietra, and Lafferty, 1997; Berger, Della Pietra, and Della Pietra, 1996). Finally, the conditional probability distribution $p_{\mathcal{S}}(e_p \mid v)$ is estimated.

$$p_{\mathcal{S}}(e_p \mid v) = \frac{\exp\Big(\sum_{f_i \in \mathcal{S}} \lambda_i f_i(v, e_p)\Big)}{\sum_{e_p} \exp\Big(\sum_{f_i \in \mathcal{S}} \lambda_i f_i(v, e_p)\Big)} \tag{25}$$

4.4 Subcategorization Preference in Parsing a Sentence

Suppose that, after estimating parameters of subcategorization preference from the training corpus $\mathcal{E}$ of verb-noun collocations, we obtain the set $\mathcal{S}$ of active features and the model $p_{\mathcal{S}}(e_p \mid v)$ incorporating these features. Now, we describe how to rank parse trees of a given input sentence according to the estimated parameters of subcategorization preference of verbs.

[4] More precisely, for a tuple $\langle s_1, \ldots, s_n \rangle$ of independent partial subcategorization frames to be included in the set $SF(e)$, the following requirement has to be satisfied: it is not possible to divide any of the partial frames $s_1, \ldots, s_n$ into more than one frame and to construct a finer-grained tuple $\langle s'_1, \ldots, s'_n, \ldots, s'_{n+k} \rangle$ of independent partial subcategorization frames.

[5] When applying the learned probabilistic model to the held-out test event e^{ts}, independence of the partial subcategorization frames are judged using the probabilities of partial subcategorization frames estimated from the *training data* (as described in section 2.3.4), then the set $SF(e^{ts})$ is constructed.

4.4.1 Basic Model

Let w be the given input sentence, $T(w)$ be the set of parse trees of w, t be a parse tree in $T(w)$, $E(t)$ be the set of verb-noun collocations contained in t. Then, each parse tree is assigned the product of all the conditional probabilities $p_S(e_p^{ts} \mid v)$ of verb-noun collocations (v, e_p^{ts}) within it, which is denoted by $\phi(t)$:

$$\phi(t) \equiv \prod_{(v,e_p^{ts}) \in E(t)} p_S(e_p^{ts} \mid v) \tag{26}$$

A parse tree $t(\in T(w))$ with the greatest value of $\phi(t)$ is chosen as the best parse tree $\hat{t}$ of w.

$$\hat{t} = \arg \max_{t \in T(w)} \phi(t)$$

4.4.2 Heuristics of Case Covering

Along with the estimated conditional probabilities $p_S(e_p^{ts} \mid v)$ and the basic model above, we consider a heuristics concerning covering of the cases of verb-noun collocations as below and evaluate their effectiveness in the experiments of the next section.

Let (v, e_p^{ts}) be a test event which is not included in the training corpus $\mathcal{E}$ (i.e., $(v, e_p^{ts}) \notin \mathcal{E}$). Subcategorization preference of test events is determined according to whether each case p (and the leaf class marked by p) of e_p^{ts} is covered by at least one feature in $\mathcal{S}$.

More formally, we introduce *case covering relation* $\preceq_{cv}$ of a verb-noun collocation (v, e_p) and a feature set S:

$(v, e_p) \preceq_{cv} S$ iff. for each case p (and the leaf class c_l marked by p) of e_p, at least one subcategorization frame corresponding to a feature in S has the same case p and its sense restriction c_s subsumes c_l, i.e. $c_l \preceq_c c_s$

According to this factor, (v_1, e_{p1}) is preferred to (v_2, e_{p2}) if and only if the following condition holds:

$$(v_1, e_{p1}) \preceq_{cv} \mathcal{S}, \quad (v_2, e_{p2}) \not\preceq_{cv} \mathcal{S}$$

Ranking Parse Trees

This heuristics can be also incorporated into ranking parse trees of a given input sentence.

Let w be the given input sentence, $T(w)$ be the set of parse trees of w, t be a parse tree in $T(w)$, $E(t)$ be the set of verb-noun collocations contained in t. Let $E_{cv}^{\mathcal{S}}(t)$ $(\subseteq E(t))$ be the set of verb-noun collocations (v, e_p) for which $(v, e_p) \preceq_{cv} \mathcal{S}$ holds, and $E_{ncv}^{\mathcal{S}}(t)$ $(\subseteq E(t))$ be the set of verb-noun collocations (v, e_p) for which $(v, e_p) \preceq_{cv} \mathcal{S}$ does not hold. Then, subcategorization preference of parse trees is determined as follows. t_1 is preferred to t_2 if and only if one of the following conditions (i) $\sim$ (iii) holds:

(i) $|E_{cv}^{\mathcal{S}}(t_1)| > |E_{cv}^{\mathcal{S}}(t_2)|$

(ii) $|E_{cv}^{\mathcal{S}}(t_1)| = |E_{cv}^{\mathcal{S}}(t_2)|$, $\displaystyle\prod_{(v,e_p) \in E_{cv1}^{\mathcal{S}}} p_S(e_p \mid v) > \prod_{(v,e_p) \in E_{cv2}^{\mathcal{S}}} p_S(e_p \mid v)$

(iii) $|E_{cv}^{\mathcal{S}}(t_1)| = |E_{cv}^{\mathcal{S}}(t_2)|$, $\displaystyle\prod_{(v,e_p) \in E_{cv1}^{\mathcal{S}}} p_S(e_p \mid v) = \prod_{(v,e_p) \in E_{cv2}^{\mathcal{S}}} p_S(e_p \mid v)$,

$$\prod_{(v,e_p) \in E_{ncv1}^{\mathcal{S}}} p_S(e_p \mid v) > \prod_{(v,e_p) \in E_{ncv2}^{\mathcal{S}}} p_S(e_p \mid v)$$

5 Experiments and Evaluation

5.1 Corpus and Thesaurus

As the training and test corpus, we used the EDR Japanese bracketed corpus (EDR, 1995), which contains about 210,000 sentences collected from newspaper and magazine articles. From the EDR corpus, we extracted 153,014 verb-noun collocations of 835 verbs which appear more than 50 times

Table 1: Examples of Selected Features for *"kau(buy,incur)"* (Independent-Frame Model($\alpha = 0.9$))

Order	Feature	Noun Class/*Example Nouns*	# of events
First 10 Selected Features			
1	*wo*(ACC):1404	*kippu(tickets), shouken(bills)*	22
2	*wo*(ACC):1524	*tochi(land)*	16
3	*wo*(ACC):1553	*kabu(stock)*	23
4	*wo*(ACC):14	Products	158
5	*wo*(ACC):1196	Currency, Unit	32
6	*wo*(ACC):1301	*ikari(anger)*	9
7	*wo*(ACC):1151	*hanpatsu(repulsion)*	11
8	*wo*(ACC):1462	Electronic Products	9
9	*wo*(ACC):1451	Container	2
10	*wo*(ACC):1302	*hankan(enmity)*	8
First 5 Selected Features with More Than One Cases			
30	*ga*(NOM):1259, *wo*(ACC):13	*ga*(NOM):Country, *wo*(ACC):*kokusai(government loan)*	2
53	*ni(for)*:1200, *wo*(ACC):14	*ni(for):watashi(I)*, *wo*(ACC):Products	1
54	*ni(for)*:121, *wo*(ACC):145	*ni(for)*:Human, *wo*(ACC):Products	1
61	*ni(for)*:12, *wo*(ACC):1404	*ni(for)*:Human, *wo*(ACC):*kippu(tickets)*	1
62	*ni(for)*:1205, *wo*(ACC):140	*ni(for):kodomo(child)*, *wo*(ACC):Products	1

in the corpus. These verb-noun collocations contain about 270 case-markers. We constructed the training set $\mathcal{E}$ from these 153,014 verb-noun collocations.

We used 'Bunrui Goi Hyou'(BGH) (NLRI, 1993) as the Japanese thesaurus. BGH has a six-layered abstraction hierarchy and more than 60,000 words are assigned at the leaves and its nominal part contains about 45,000 words. Five classes are allocated at the next level from the root node.

5.2 Feature Selection and Parameter Estimation

We conduct the feature selection procedure in section 3.3 and the parameter estimation procedure in section 3.2 under the following conditions: i) we limit the noun class generalization level of each feature to those which are above the level 5 from the root node in the thesaurus, ii) since verbs are independent of each other in our model learning framework, we collect verb-noun collocations of one verb into a training data set and conduct the model learning procedure for each verb separately.

For each verb, the size of the training data set is about 200 $\sim$ 500. The size of the set of candidate features varies according to the models: 200 $\sim$ 400 for independent-case model, 500 $\sim$ 1,300 for one-frame/independent-frame(independence parameter $\alpha = 0.5/0.9$) models, and 650 $\sim$ 1,550 for partial-frame model. In the independent-case model, each feature corresponds to a subcategorization frame with only one case, while in the one-frame/independent/frame/partial-frame models, each feature corresponds to a subcategorization frame with any number of cases. This is why the size of the set of candidate features is much smaller in the independent-case model than in other models. In the one-frame/independent-frame models, more restrictions are put on the definition of features than in the partial-frame model, and the sizes of the sets of candidate features are relatively smaller.

Examples of Selected Features

For a Japanese verb *"kau(buy, incur)"*, Table 1 shows examples of the selected features for the independent-frame model (independence parameter $\alpha = 0.9$). In the table, first 10 selected features, as well as first 5 selected features corresponding to (partial) subcategorization frames with more than one cases, are shown. In the tables, each feature is represented as the corresponding (partial) subcategorization frame which consists of pairs of a case-marking particle and the noun class restriction of the case. Each noun class restriction is represented as a Japanese noun class of BGH thesaurus. Noun classes of BGH thesaurus are represented as numerical codes, in which each

digit denotes the choice of the branch in the thesaurus. The classes starting with '11', '12', '13', '14', and '15' are subordinate to *abstract-relations, agents-of-human-activities, human-activities, products* and *natural-objects-and-natural-phenomena*, respectively. Each table consists of the order of the feature, the feature itself (which is represented as a (partial) subcategorization frame), noun class descriptions or example nouns in the (partial) subcategorization frames, and the number of the training verb-noun collocations for which the feature function returns true.

Since about 75% of the verb-noun collocations in the training set have only one case-marked noun, all of the first 10 selected features have only one cases in both of the independent-frame/partial-frame models. However, the two models are different in the orders of the first 5 selected features with more than one cases. In the partial-frame model, those 5 features have much superior orders than in the independent-frame model. In the partial-frame model, less restrictions are put on the definitions of features than in the independent-frame model. Therefore, in the partial-frame model, the feature functions corresponding to (partial) subcategorization frames with more than one cases tend to return true for more verb-noun collocations than in the independent-frame model.

5.3 Evaluation of Subcategorization Preference

5.3.1 Evaluation Method

We evaluate the performance of the selected features and their estimated parameters in the following subcategorization preference task. Suppose that the following word sequence represents a verb-final Japanese sentence with a subordinate clause, where $N_x, \ldots, N_{2k}$ are nouns, $p_x, \ldots, p_{2k}$ are case-marking post-positional particles, and v_1, v_2 are verbs, and the first verb v_1 is the head verb of the subordinate clause.

$$N_x\text{-}p_x\text{-}N_{11}\text{-}p_{11}\text{-}\cdots\text{-}N_{1l}\text{-}p_{1l}\text{-}v_1\text{-}N_{21}\text{-}p_{21}\text{-}\cdots\text{-}N_{2k}\text{-}p_{2k}\text{-}v_2$$

We consider the subcategorization ambiguity of the post-positional phrase N_x-p_x: i.e, whether N_x-p_x is subcategorized for by v_1 or v_2.

We use held-out verb-noun collocations of the verbs v_1 and v_2 which are not used in the training. They are like those verb-noun collocations in the left side below. Next, we generate erroneous verb-noun collocations of v_1 and v_2 as those in the right side below, by choosing a case element $p_x : N_x$ at random and moving it from v_1 to v_2.

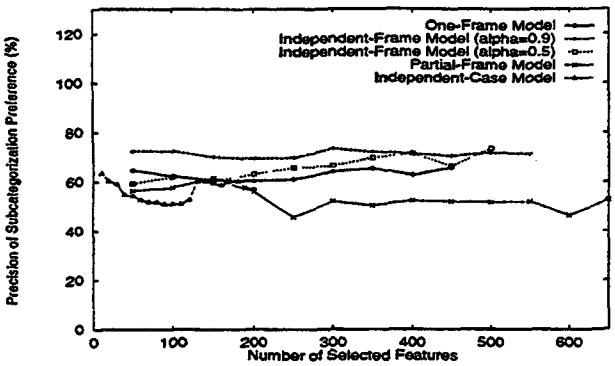
$$\boxed{\text{Correct}} \qquad\qquad \boxed{\text{Error}}$$

$$\begin{bmatrix} pred : v_1 \\ p_{11} : N_{11} \\ \vdots \\ p_{1l} : N_{1l} \\ \underline{p_x : N_x} \end{bmatrix} \begin{bmatrix} pred : v_2 \\ p_{21} : N_{21} \\ \vdots \\ p_{2l} : N_{2k} \end{bmatrix} \Longleftrightarrow \begin{bmatrix} pred : v_1 \\ p_{11} : N_{11} \\ \vdots \\ p_{1l} : N_{1l} \end{bmatrix} \begin{bmatrix} pred : v_2 \\ p_{21} : N_{21} \\ \vdots \\ p_{2k} : N_{2k} \\ \underline{p_x : N_x} \end{bmatrix}$$

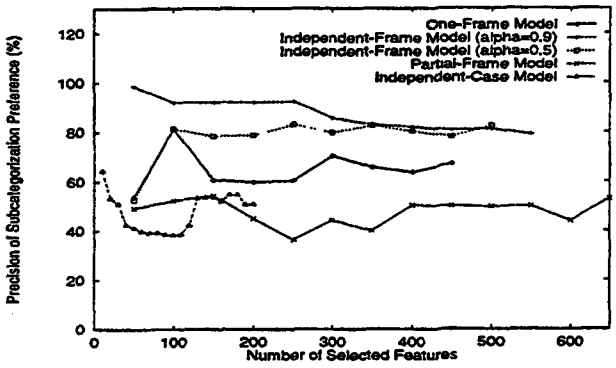
Then, we compare the products $\phi(t)$ (in the equation (26)) of the conditional probabilities of the constituent verb-noun collocations between the correct and the erroneous pairs, and calculate the rate of selecting the correct pair. We measure the following three types of precisions: i) the precision r_b of the *basic model* in section 4.4.1, ii) the precision r_h when incorporating the *heuristics* in section 4.4.2, iii) the precision r_c of those verb-noun collocations which satisfy the *case covering relation* $\preceq_{cv}$ with the set $\mathcal{S}$ of active features, i.e., this means that we collect verb-noun collocations (v_1, e_{p1}) and (v_2, e_{p2}) of the verbs v_1 and v_2 which satisfy the *case covering relation* $(v_1, e_{p1}), (v_2, e_{p2}) \preceq_{cv} \mathcal{S}$, and calculate the precision r_c.

5.3.2 Results

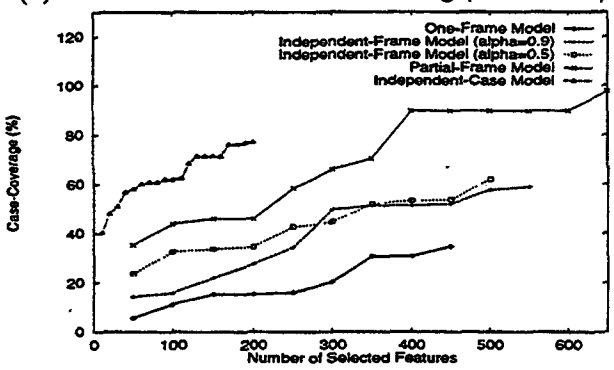
Figure 1 (a)$\sim$(c) compares the precisions r_c and r_h among the one-frame/independent-frame/partial-frame/independent-case models. We also compare the changes of the rate of the verb-noun collocations in the test set which satisfy the *case covering relation* $\preceq_{cv}$ with the set $\mathcal{S}$ of active features.



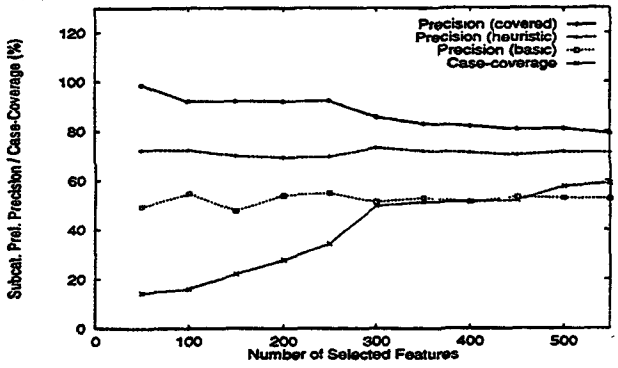
(a) With Heuristics of Case Covering (All Models)



(b) Precisions of Case-Covered Events (All Models)



(c) Case-Coverage of Test Data (All Models)



(d) Independent-Frame Model ($\alpha = 0.9$)

Figure 1: Changes in Case-Coverage of Test Data and Precisions of Subcategorization Preference

For the independent-frame model, we examined two different values of the independence parameter α, i.e., $\alpha = 0.5$ as a weak condition on independence judgment and $\alpha = 0.9$ as a strict condition on independence judgment. Figure 1 (d) shows the changes of the precisions r_b, r_h, and r_c as well as the case-coverage of the test data during the training for the independent-frame model (the independence parameter $\alpha = 0.9$). Both of the precisions r_c and r_h of the independent-frame model are higher than those of any other models. On the other hand, the case-coverage of the independent-frame model (as well as the that of one-frame model) is much lower than that of the partial-frame/independent-case models. The decrease of the case-coverage in the independent-frame/one-frame models is caused by the overfitting to the training data.[6]

In the case of the independent-frame model, precisions decrease in the order of r_c, r_h, and r_b. This means that the independent-frame model performs well in the task of subcategorization preference when the verb-noun collocations satisfy the case covering relation $\preceq_{cv}$ with the set $\mathcal{S}$ of active features. When the verb-noun collocations do not satisfy the case covering relation, we have to use the heuristics of case covering in section 4.4.2 and then the precision of subcategorization preference decreases. If we do not care whether the verb-noun collocations satisfy the case covering relation and do not use the heuristics of case covering, this means that we use the basic model in

[6]The reason why the overfitting to the training data occurs in the independent-frame/one-frame models can be explained by comparing the effects of the two values of the independence parameter α in the independent model. When α equals to 0.9, both r_c and r_h are slightly higher than when α equals to 0.5. Especially, when the number of selected features are less than 300, r_c is much higher when α equals to 0.9 than when α equals to 0.5, although the case-coverage of the test data is much lower. When the condition on independence judgment becomes more strict, the cases in the training data are judged as dependent on each other more often and then this causes the estimated model to overfit to the training data. In the case of the independent-frame model, overfit to the training data seems to result in higher performance in subcategorization preference task, although the case-coverage of the test data is caused to become lower.

section 4.4.1 and it performs worst as indicated by the precision r_b.

6 Conclusion

This paper proposed a novel method for learning probabilistic models of subcategorization preference of verbs. We proposed to consider the issues of case dependencies and noun class generalization in a uniform way. We adopted the maximum entropy model learning method and applied it to the task of model learning of subcategorization preference.[7] We described the results of the experiment on learning the models of subcategorization preference from the EDR Japanese bracketed corpus. We evaluated the performance of the selected features and their estimated parameters in the subcategorization preference task. In this evaluation task, the independent-frame model with the independence parameter $\alpha = 0.9$ performed best in the precision when incorporating the heuristics of case-covering, as well as in the precision of case-covered test events. As for further issues, it is important to improve the case-coverage of the independent-frame model without decreasing the precision of subcategorization preference. For this purpose, we have already invented a new feature selection algorithm which meets the above requirement on preserving high case-coverage with a relatively small number of active features.[8] We will report the details of applying this new algorithm to the task of model learning of subcategorization preference in the near future.

References

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

Black, E. 1993. Towards history-based grammars: Using richer models for probabilistic parsing. In *Proceedings of the 31st Annual Meeting of ACL*, pages 31–37.

Collins, M. 1996. A new statistical parser based on bigram lexical dependencies. In *Proceedings of the 34th Annual Meeting of ACL*, pages 184–191.

Della Pietra, S., V. Della Pietra, and J. Lafferty. 1997. Inducing features of random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(4):380–393.

EDR, (Japan Electronic Dictionary Research Institute, Ltd.), 1995. *EDR Electronic Dictionary Technical Guide.*

Haruno, M. 1995. Verbal case frame acquisition as data compression. In *Proceedings of the 5th International Workshop on Natural Language Understanding and Logic Programming*, pages 45–50.

Li, H. and N. Abe. 1995. Generalizing case frames using a thesaurus and the MDL principle. In *Proceedings of International Conference on Recent Advances in Natural Language Processing*, pages 239–248.

Li, H. and N. Abe. 1996. Learning dependencies between case frame slots. In *Proceedings of the 16th COLING*, pages 10–15.

Magerman, D. M. 1995. Statistical decision-tree models for parsing. In *Proceedings of the 33rd Annual Meeting of ACL*, pages 276–283.

NLRI, (National Language Research Institute), 1993. *Word List by Semantic Principles.* Syuei Syuppan. (in Japanese).

Resnik, P. 1993. Semantic classes and syntactic ambiguity. In *Proceedings of the Human Language Technology Workshop*, pages 278–283.

Rissanen, J. 1989. *Stochastic Complexity in Statistical Inquiry*, volume 15. World Scientific Publishing Company.

Utsuro, T. and Y. Matsumoto. 1997. Learning probabilistic subcategorization preference by identifying case dependencies and optimal noun class generalization level. In *Proceedings of the 5th ANLP*, pages 364–371.

[7] Previously, in Utsuro and Matsumoto (1997), we have proposed a method of learning probabilistic subcategorization preference, which is not based-on any probabilistic model learning techniques. Compared with this previous paper, one of the underlying motivations of the work in this paper is that, by applying general probabilistic model learning techniques such as the maximum entropy model learning method, it becomes easier to incorporate various kinds of linguistic information into the model of subcategorization preference. For example, as such linguistic information, we are planning to incorporate syntactic features concerning idiomatic expression as well as voice of the verb.

[8] The basic idea of the new algorithm is as follows: first, it starts from the independent-case model with relatively general sense restrictions which correspond to higher classes in the thesaurus. This starting model satisfies the requirement on high case-coverage. Then, the algorithm gradually examines the case dependencies as well as more specific sense restrictions which correspond to lower classes in the thesaurus. The model search process is controlled according to some information-theoretic model evaluation criteria such as the MDL principle (Rissanen, 1989).