Generalizing Case Frames Using a Thesaurus and the MDL Principle

Hang Li*
NEC Corporation

Naoki Abe*
NEC Corporation

A new method for automatically acquiring case frame patterns from large corpora is proposed. In particular, the problem of generalizing values of a case frame slot for a verb is viewed as that of estimating a conditional probability distribution over a partition of words, and a new generalization method based on the Minimum Description Length (MDL) principle is proposed. In order to assist with efficiency, the proposed method makes use of an existing thesaurus and restricts its attention to those partitions that are present as "cuts" in the thesaurus tree, thus reducing the generalization problem to that of estimating a "tree cut model" of the thesaurus tree. An efficient algorithm is given, which provably obtains the optimal tree cut model for the given frequency data of a case slot, in the sense of MDL. Case frame patterns obtained by the method were used to resolve PP-attachment ambiguity. Experimental results indicate that the proposed method improves upon or is at least comparable with existing methods.

1. Introduction

We address the problem of automatically acquiring case frame patterns (selectional patterns, subcategorization patterns) from large corpora. A satisfactory solution to this problem would have a great impact on various tasks in natural language processing, including the structural disambiguation problem in parsing. The acquired knowledge would also be helpful for building a lexicon, as it would provide lexicographers with word usage descriptions.

In our view, the problem of acquiring case frame patterns involves the following two issues: (a) acquiring patterns of individual case frame slots; and (b) learning dependencies that may exist between different slots. In this paper, we confine ourselves to the former issue, and refer the interested reader to Li and Abe (1996), which deals with the latter issue.

The case frame (case slot) pattern acquisition process consists of two phases: **extraction** of case frame instances from corpus data, and **generalization** of those instances to case frame patterns. The generalization step is needed in order to represent the input case frame instances more compactly as well as to judge the (degree of) acceptability of unseen case frame instances. For the extraction problem, there have been various methods proposed to date, which are quite adequate (Hindle and Rooth 1991; Grishman and Sterling 1992; Manning 1992; Utsuro, Matsumoto, and Nagao 1992; Brent 1993; Smadja 1993; Grefenstette 1994; Briscoe and Carroll 1997). The generalization problem, in contrast, is a more challenging one and has not been solved completely. A number of methods for generalizing values of a case frame slot for a verb have been

* C&C Media Res. Labs., NEC Corporation, 4-1-1 Miyazaki Miyamae-ku, Kawasaki 216, Japan. E-mail:{lihang,abe}@ccm.cl.nec.co.jp

proposed. Some of these methods make use of prior knowledge in the form of an existing thesaurus (Resnik 1993a, 1993b; Framis 1994; Almuallim et al. 1994; Tanaka 1996; Utsuro and Matsumoto 1997), while others do not rely on any prior knowledge (Pereira, Tishby, and Lee 1993; Grishman and Sterling 1994; Tanaka 1994). In this paper, we propose a new generalization method, belonging to the first of these two categories, which is both theoretically well-motivated and computationally efficient.

Specifically, we formalize the problem of generalizing values of a case frame slot for a given verb as that of estimating a conditional probability distribution over a partition of words, and propose a new generalization method based on the **Minimum Description Length principle** (MDL): a principle of data compression and statistical estimation from information theory.[1] In order to assist with efficiency, our method makes use of an existing thesaurus and restricts its attention on those partitions that are present as "cuts" in the thesaurus tree, thus reducing the generalization problem to that of estimating a "tree cut model" of the thesaurus tree. We then give an efficient algorithm that provably obtains the optimal tree cut model for the given frequency data of a case slot, in the sense of MDL. In order to test the effectiveness of our method, we conducted PP-attachment disambiguation experiments using the case frame patterns obtained by our method. Our experimental results indicate that the proposed method improves upon or is at least comparable to existing methods.

The remainder of this paper is organized as follows: In Section 2, we formalize the problem of generalizing values of a case frame slot as that of estimating a conditional distribution. In Section 3, we describe our MDL-based generalization method. In Section 4, we present our experimental results. We then give some concluding remarks in Section 5.

2. The Problem

2.1 The Data Sparseness Problem

Suppose that the data available to us are of the type shown in Table 1, which are slot values for a given verb (*verb,slot_name,slot_value* triples) automatically extracted from a corpus using existing techniques. By counting the frequency of occurrence of each noun at a given slot of a verb, the frequency data shown in Figure 1 can be obtained. We will refer to this type of data as **co-occurrence data**. The problem of generalizing values of a case frame slot for a verb (or, in general, a head) can be viewed as the problem of learning the underlying **conditional probability distribution** that gives rise to such co-occurrence data. Such a conditional distribution can be represented by a probability model that specifies the conditional probability $P(n \mid v, r)$ for each n in the set of nouns $\mathcal{N} = \{n_1, n_2, \ldots, n_N\}$, v in the set of verbs $\mathcal{V} = \{v_1, v_2, \ldots, v_V\}$, and r in the set of slot names $\mathcal{R} = \{r_1, r_2, \ldots, r_R\}$, satisfying:

$$\sum_{n \in \mathcal{N}} P(n \mid v, r) = 1. \tag{1}$$

This type of probability model is often referred to as a word-based model. Since the number of probability parameters in word-based models is large ($O(N \cdot V \cdot R)$), accurate

1 Recently, MDL and related techniques have become popular in corpus-based natural language processing and other related fields (Ellison 1991, 1992; Cartwright and Brent 1994; Stolcke and Omohundro 1994; Brent, Murthy, and Lundberg 1995; Ristad and Thomas 1995; Brent and Cartwright 1996; Grunwald 1996). In this paper, we introduce MDL into the context of case frame pattern acquisition.

Table 1
Example (*verb, slot_name, slot_value*) triple data.

verb	*slot_name*	*slot_value*
fly	arg1	bee
fly	arg1	bird
fly	arg1	bird
fly	arg1	crow
fly	arg1	bird
fly	arg1	eagle
fly	arg1	bee
fly	arg1	eagle
fly	arg1	bird
fly	arg1	crow

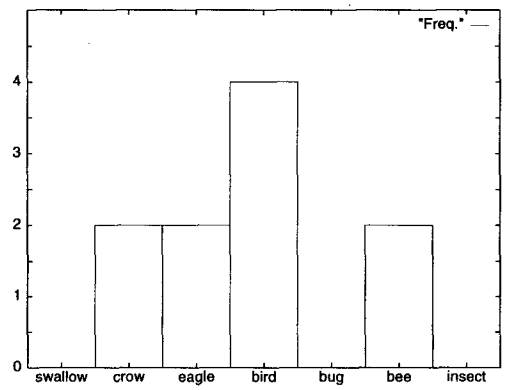

Figure 1
Frequency data for the subject slot of verb *fly*.

estimation of a word-based model is difficult with the data size that is available in practice—a problem usually referred to as the data sparseness problem. For example, suppose that we employ the maximum-likelihood estimation (or MLE for short) to estimate the probability parameters of a conditional probability distribution, as described above, given the co-occurrence data in Figure 1. In this case, MLE amounts to estimating the parameters by simply normalizing the frequencies so that they sum to one, giving, for example, the estimated probabilities of 0, 0.2, and 0.4 for *swallow*, *eagle*, and *bird*, respectively (see Figure 2). Since in general the number of parameters exceeds the size of data that is typically available, MLE will result in estimating most of the probability parameters to be zero.

To address this problem, Grishman and Sterling (1994) proposed a method of smoothing conditional probabilities using the probability values of similar words, where the similarity between words is judged based on co-occurrence data (see also Dagan, Marcus, and Makovitch [1992] and Dagan, Pereira, and Lee [1994]). More specifically, conditional probabilities of words are smoothed by taking the weighted average of those of similar words using the similarity measure as the weights. The advantage of this approach is that it does not rely on any prior knowledge, but it appears difficult to find a smoothing method that is both efficient and theoretically sound. As an alternative, a number of authors have proposed the use of class-based

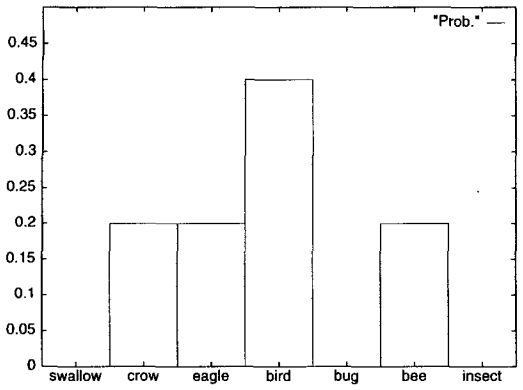

Figure 2
Word-based distribution estimated using MLE.

models, which assign (conditional) probability values to (existing) classes of words, rather than individual words.

2.2 Class-based Models

An example of the class-based approach is Resnik's method of generalizing values of a case frame slot using a thesaurus and the so-called selectional association measure (Resnik 1993a, 1993b). The selectional association, denoted $A(C \mid v,r)$, is defined as follows:

$$A(C \mid v,r) = P(C \mid v,r) \times \log \frac{P(C \mid v,r)}{P(C)} \tag{2}$$

where C is a class of nouns present in a given thesaurus, v is a verb and r is a slot name, as described earlier. In generalizing a given noun n to a noun class, this method selects the noun class C having the *maximum* $A(C \mid v,r)$, among all super classes of n in a given thesaurus. This method is based on an interesting intuition, but its interpretation as a method of estimation is not clear. We propose a class-based generalization method whose performance as a method of estimation is guaranteed to be near optimal.

We define the class-based model as a model that consists of a partition of the set $\mathcal{N}$ of nouns, and a parameter associated with each member of the partition. Here, a partition Γ of $\mathcal{N}$ is any collection of mutually disjoint subsets of $\mathcal{N}$ that exhaustively cover $\mathcal{N}$. The parameters specify the conditional probability $P(C \mid v,r)$ for each class (subset) C in that partition, such that

$$\sum_{C \in \Gamma} P(C \mid v,r) = 1. \tag{3}$$

Within a given class C, it is assumed that each noun is generated with equal probability, namely

$$\forall n \in C\colon\ P(n \mid v,r) = \frac{1}{|C|} \times P(C \mid v,r). \tag{4}$$

Here, we assume that a word belongs to a single class. In practice, however, many words have sense ambiguity and a word can belong to several different classes, e.g., *bird* is a member of both BIRD and MEAT. Thorough treatment of this problem is beyond the scope of the present paper; we simply note that one can employ an existing word-sense disambiguation technique (e.g.,Yarowsky 1992, 1994) in preprocessing, and use the disambiguated word senses as virtual words in the following

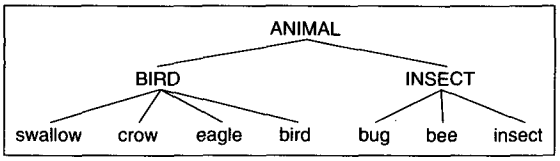

Figure 3
An example thesaurus.

case-pattern acquisition process. It is also possible to extend our model so that each word probabilistically belongs to several different classes, which would allow us to resolve both structural and word-sense ambiguities at the time of disambiguation.[2] Employing probabilistic membership, however, would make the estimation process significantly more computationally demanding. We therefore leave this issue as a future topic, and employ a simple heuristic of equally distributing each word occurrence in the data to all of its potential word senses in our experiments. Since our learning method based on MDL is robust against noise, this should not significantly degrade performance.

2.3 The Tree Cut Model

Since the number of partitions for a given set of nouns is extremely large, the problem of selecting the best model from among all possible class-based models is most likely intractable. In this paper, we reduce the number of possible partitions to consider by using a thesaurus as prior knowledge, following a basic idea of Resnik's (1992).

In particular, we restrict our attention to those partitions that exist within the thesaurus in the form of a cut. By thesaurus, we mean a tree in which each leaf node stands for a noun, while each internal node represents a noun class, and domination stands for set inclusion (see Figure 3). A cut in a tree is any set of nodes in the tree that defines a partition of the leaf nodes, viewing each node as representing the set of all leaf nodes it dominates. For example, in the thesaurus of Figure 3, there are five cuts: [ANIMAL], [BIRD, INSECT], [BIRD, bug, bee, insect], [swallow, crow, eagle, bird, INSECT], and [swallow, crow, eagle, bird, bug, bee, insect]. The class of tree cut models of a fixed thesaurus tree is then obtained by restricting the partition Γ in the definition of a class-based model to be those partitions that are present as a cut in that thesaurus tree.

Formally, a tree cut model M can be represented by a pair consisting of a tree cut Γ and a probability parameter vector θ of the same length, that is:

$$M = (\Gamma, \theta) \tag{5}$$

where Γ and θ are:

$$\Gamma = [C_1, C_2, \ldots, C_{k+1}], \theta = [P(C_1), P(C_2), \ldots, P(C_{k+1})] \tag{6}$$

where $C_1, C_2, \ldots, C_{k+1}$ is a cut in the thesaurus tree and $\sum_{i=1}^{k+1} P(C_i) = 1$ is satisfied. For simplicity we sometimes write $P(C_i), i = 1, \ldots, (k+1)$ for $P(C_i \mid v, r)$.

If we use MLE for the parameter estimation, we can obtain five tree cut models from the co-occurrence data in Figure 1; Figures 4–6 show three of these. For example,

2 The model used by Pereira, Tishby, and Lee (1993) is indeed along this direction.

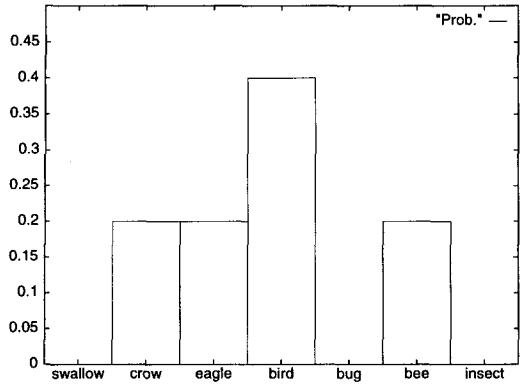

Figure 4
A tree cut model with [swallow, crow, eagle, bird, bug, bee, insect].

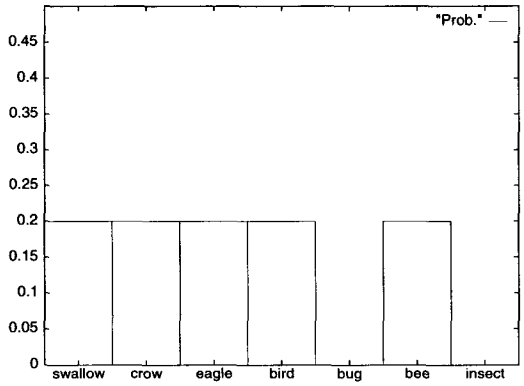

Figure 5
A tree cut model with [BIRD, bug, bee, insect].

$\hat{M}$ = ([BIRD, bug, bee, insect], [0.8, 0, 0.2, 0]) shown in Figure 5 is one such tree cut model. Recall that $\hat{M}$ defines a conditional probability distribution $P_{\hat{M}}(n \mid v, r)$ as follows: For any noun that is in the tree cut, such as *bee*, the probability is given as explicitly specified by the model, i.e., $P_{\hat{M}}(\text{bee} \mid \text{fly}, \text{arg1}) = 0.2$. For any class in the tree cut, the probability is distributed uniformly to all nouns dominated by it. For example, since there are four nouns that fall under the class BIRD, and *swallow* is one of them, the probability of *swallow* is thus given by $P_{\hat{M}}(\text{swallow} \mid \text{fly}, \text{arg1}) = 0.8/4 = 0.2$. Note that the probabilities assigned to the nouns under BIRD are smoothed, even if the nouns have different observed frequencies.

We have thus formalized the problem of generalizing values of a case frame slot as that of estimating a model from the class of tree cut models for some fixed thesaurus tree; namely, selecting a model that best explains the data from among the class of tree cut models.

3. Generalization Method Based On MDL

The question now becomes what strategy (criterion) we should employ to select the *best* tree-cut model. We adopt the Minimum Description Length principle (Rissanen 1978,

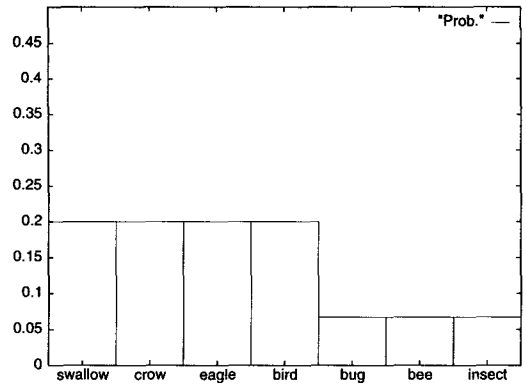

Figure 6
A tree cut model with [BIRD, INSECT].

Table 2
Number of parameters and KL distance from the empirical distribution for the five tree cut models.

Γ	Number of Parameters	KL Distance
[ANIMAL]	0	0.89
[BIRD, INSECT]	1	0.72
[BIRD, bug, bee, insect]	3	0.4
[swallow, crow, eagle, bird, INSECT]	4	0.32
[swallow, crow, eagle, bird, bug, bee, insect]	6	0

1983, 1984, 1986, 1989), which has various desirable properties, as will be described later.[3]

MDL is a principle of data compression and statistical estimation from information theory, which states that the best probability model for given data is that which requires the least code length in bits for the encoding of the model itself and the given data observed through it.[4] The former is the **model description length** and the latter the **data description length**.

In our current problem, it tends to be the case, in general, that a model nearer the root of the thesaurus tree, such as that in Figure 6, is simpler (in terms of the number of parameters), but tends to have a poorer fit to the data. In contrast, a model nearer the leaves of the thesaurus tree, such as that in Figure 4, is more complex, but tends to have a better fit to the data. Table 2 shows the number of free parameters and the **KL distance** from the empirical distribution of the data (namely, the word-based distribution estimated by MLE) shown in Figure 2 for each of the five tree cut models.[5] In the table, one can see that there is a trade-off between the simplicity of a model and the goodness of fit to the data.

In the MDL framework, the model description length is an indicator of model

3 Estimation strategies related to MDL have been independently proposed and studied by various authors (Solomonoff 1964; Wallace and Boulton 1968; Schwarz 1978; Wallace and Freeman 1992).
4 We refer the interested reader to Quinlan and Rivest (1989) for an introduction to the MDL principle.
5 The KL distance (also known as KL-divergence or relative entropy), which is widely used in information theory and statistics, is a measure of distance between two distributions (e.g., Cover and Thomas 1991). It is always nonnegative and is zero if and only if the two distributions are identical, but is asymmetric and hence not a metric (the usual notion of distance).

complexity, while the data description length indicates goodness of fit to the data. The MDL principle stipulates that the model that minimizes the sum total of the description lengths should be the best model (both for data compression and statistical estimation).

In the remainder of this section, we will describe how we apply MDL to our current problem. We will then discuss the rationale behind using MDL in our present context.

3.1 Calculating Description Length

We first show how the description length for a model is calculated. We use S to denote a sample (or set of data), which is a **multiset** of examples, each of which is an occurrence of a noun at a given slot r of a given verb v (i.e., duplication is allowed). We let $|S|$ denote the size of S as a multiset, and $n \in S$ indicate the inclusion of n in S as a multiset. For example, the column labeled *slot_value* in Table 1 represents a sample S for the subject slot of *fly*, and in this case $|S| = 10$.

Given a sample S and a tree cut Γ, we employ MLE to estimate the parameters of the corresponding tree cut model $\hat{M} = (\Gamma, \hat{\theta})$, where $\hat{\theta}$ denotes the estimated parameters.

The total description length $L(\hat{M}, S)$ of the tree cut model $\hat{M}$ and the sample S observed through $\hat{M}$ is computed as the sum of the model description length $L(\Gamma)$, parameter description length $L(\hat{\theta} \mid \Gamma)$, and data description length $L(S \mid \Gamma, \hat{\theta})$:

$$L(\hat{M}, S) = L((\Gamma, \hat{\theta}), S) = L(\Gamma) + L(\hat{\theta} \mid \Gamma) + L(S \mid \Gamma, \hat{\theta}). \tag{7}$$

Note that we sometimes refer to $L(\Gamma) + L(\hat{\theta} \mid \Gamma)$ as the model description length.

The model description length $L(\Gamma)$ is a subjective quantity, which depends on the coding scheme employed. Here, we choose to assign the same code length to each cut and let:

$$L(\Gamma) = \log |\mathcal{G}| \tag{8}$$

where $\mathcal{G}$ denotes the set of all cuts in the thesaurus tree T.[6] This corresponds to assuming that each tree cut model is equally likely a priori, in the Bayesian interpretation of MDL. (See Section 3.4.)

The parameter description length $L(\hat{\theta} \mid \Gamma)$ is calculated by:

$$L(\hat{\theta} \mid \Gamma) = \frac{k}{2} \times \log |S| \tag{9}$$

where $|S|$ denotes the sample size and k denotes the number of free parameters in the tree cut model, i.e., k equals the number of nodes in Γ minus one. It is known to be best to use this number of bits to describe probability parameters in order to minimize the expected total description length (Rissanen 1984, 1986). An intuitive explanation of this is that the standard deviation of the maximum-likelihood estimator of each parameter is of the order $\frac{1}{\sqrt{|S|}}$, and hence describing each parameter using more than $-\log \frac{1}{\sqrt{|S|}} = \frac{1}{2} \cdot \log |S|$ bits would be wasteful for the estimation accuracy possible with the given sample size.

Finally, the data description length $L(S \mid \Gamma, \hat{\theta})$ is calculated by:

$$L(S \mid \Gamma, \hat{\theta}) = -\sum_{n \in S} \log \hat{P}(n) \tag{10}$$

6 Here and throughout, log denotes the logarithm to the base 2. For reasons why Equation 8 holds, see, for example, Quinlan and Rivest (1989).

Table 3
Calculating the description length for the model of Figure 5.

C	BIRD	bug	bee	insect
$f(C)$	8	0	2	0
$\|C\|$	4	1	1	1
$\hat{P}(C)$	0.8	0.0	0.2	0.0
$\hat{P}(n)$	0.2	0.0	0.2	0.0
Γ	[BIRD, bug, bee, insect]			
$L(\hat{\theta} \mid \Gamma)$	$\frac{(4-1)}{2} \times \log 10 = 4.98$			
$L(S \mid \Gamma, \hat{\theta})$	$-(2+4+2+2) \times \log 0.2 = 23.22$			

where for simplicity we write $\hat{P}(n)$ for $P_{\hat{M}}(n \mid v, r)$. Recall that $\hat{P}(n)$ is obtained by MLE, namely, by normalizing the frequencies:

$$\hat{P}(n) = \frac{1}{|C|} \times \hat{P}(C) \tag{11}$$

for each $C \in \Gamma$ and each $n \in C$, where for each $C \in \Gamma$:

$$\hat{P}(C) = \frac{f(C)}{|S|} \tag{12}$$

where $f(C)$ denotes the total frequency of nouns in class C in the sample S, and Γ is a tree cut. We note that, in fact, the maximum-likelihood estimate is one that minimizes the data description length $L(S \mid \Gamma, \hat{\theta})$.

With description length defined in the above manner, we wish to select a model with the minimum description length and output it as the result of generalization. Since we assume here that every tree cut has an equal $L(\Gamma)$, technically we need only calculate and compare $L'(\hat{M}, S) = L(\hat{\theta} \mid \Gamma) + L(S \mid \Gamma, \hat{\theta})$ as the description length. For simplicity, we will sometimes write just $L'(\Gamma)$ for $L'(\hat{M}, S)$, where Γ is the tree cut of $\hat{M}$, when $\hat{M}$ and S are clear from context.

The description lengths for the data in Figure 1 using various tree cut models of the thesaurus tree in Figure 3 are shown in Table 4. (Table 3 shows how the description length is calculated for the model of tree cut [BIRD, bug, bee, insect].) These figures indicate that the model in Figure 6 is the best model, according to MDL. Thus, given the data in Table 1 as input, the generalization result shown in Table 5 is obtained.

3.2 An Efficient Algorithm

In generalizing values of a case frame slot using MDL, we could, in principle, calculate the description length of every possible tree cut model and output a model with the minimum description length as the generalization result, if computation time were of no concern. But since the number of cuts in a thesaurus tree is exponential in the size of the tree (for example, it is easy to verify that for a complete b-ary tree of depth d it is of the order $O(2^{b^{d-1}})$), it is impractical to do so. Nonetheless, we were able to devise a

Table 4
Description length of the five tree cut models.

Γ	$L(\hat{\theta} \mid \Gamma)$	$L(S \mid \Gamma, \hat{\theta})$	$L'(\Gamma)$
[ANIMAL]	0	28.07	28.07
[BIRD, INSECT]	1.66	26.39	28.05
[BIRD, bug, bee, insect]	4.98	23.22	28.20
[swallow, crow, eagle, bird, INSECT]	6.64	22.39	29.03
[swallow, crow, eagle, bird, bug, bee, insect]	9.97	19.22	29.19

Table 5
Generalization result.

verb	*slot_name*	*slot_value*	probability
fly	arg1	BIRD	0.8
fly	arg1	INSECT	0.2

Here we let t denote a thesaurus (sub)tree, root(t) the root of the tree t.
Initially t is set to the entire tree.
Also input to the algorithm is a co-occurrence data.
algorithm Find-MDL(t) := cut

```
1.  if
2.      t is a leaf node
3.  then
4.      return([t])
5.  else
6.      For each child tree t_i of t c_i :=Find-MDL(t_i)
7.      c:= append(c_i)
8.      if
9.          L'([root(t)]) < L'(c)
10.     then
11.         return([root(t)])
12.     else
13.         return(c)
```

Figure 7
The algorithm: Find-MDL.

simple and efficient algorithm based on dynamic programming, which is guaranteed to find a model with the minimum description length.

Our algorithm, which we call Find-MDL, recursively finds the optimal MDL model for each child subtree of a given tree and appends all the optimal models of these subtrees and returns the appended models, unless collapsing all the lower-level optimal models into a model consisting of a single node (the root node of the given tree) reduces the total description length, in which case it does so. The details of the algorithm are given in Figure 7. Note that for simplicity we describe Find-MDL as outputting a tree cut, rather than a complete tree cut model.

Note in the above algorithm that the parameter description length is calculated as

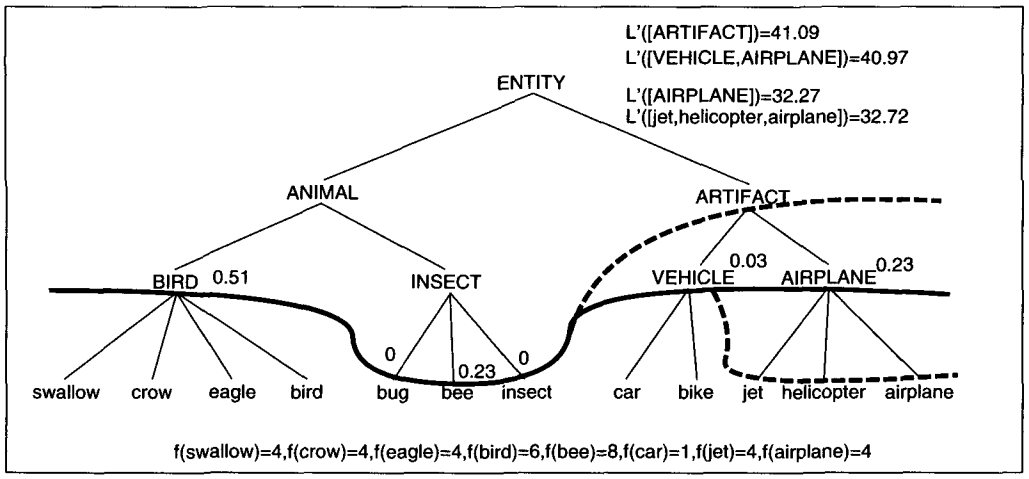

Figure 8
An example application of Find-MDL.

$\frac{k+1}{2} \log |S|$, where $k+1$ is the number of nodes in the current cut, both when t is the entire tree and when it is a proper subtree. This contrasts with the fact that the number of free parameters is k for the former, while it is $k+1$ for the latter. For the purpose of finding a tree cut with the minimum description length, however, this distinction can be ignored (see Appendix A).

Figure 8 illustrates how the algorithm works (on the co-occurrence data shown at the bottom): In the recursive application of Find-MDL on the subtree rooted at AIRPLANE, the if-clause on line 9 evaluates to true since $L'([\text{AIRPLANE}]) = 32.27$, $L'([\text{jet, helicopter, airplane}]) = 32.72$, and hence [AIRPLANE] is returned. Then in the call to Find-MDL on the subtree rooted at ARTIFACT, the same if-clause evaluates to false since $L'([\text{VEHICLE, AIRPLANE}]) = 40.97$, $L'([\text{ARTIFACT}]) = 41.09$, and hence [VEHICLE, AIRPLANE] is returned.

Concerning the above algorithm, we show that the following proposition holds:

Proposition 1
The algorithm Find-MDL terminates in time $O(N \times |S|)$, where N denotes the number of leaf nodes in the input thesaurus tree T and $|S|$ denotes the input sample size, and outputs a tree cut model of T with the minimum description length (with respect to the encoding scheme described in Section 3.1).

Here we will give an intuitive explanation of why the proposition holds, and give the formal proof in Appendix A. The MLE of each node (class) is obtained simply by dividing the frequency of nouns within that class by the total sample size. Thus, the parameter estimation for each subtree can be done independently from the estimation of the parameters outside the subtree. The data description length for a subtree thus depends solely on the tree cut within that subtree, and its calculation can be performed independently for each subtree. As for the parameter description length for a subtree, it depends only on the number of classes in the tree cut within that subtree, and hence can be computed independently as well. The formal proof proceeds by mathematical induction, which verifies that the optimal model in any (sub)tree is either the model

consisting of the root of the tree or the model obtained by appending the optimal submodels for its child subtrees.[7]

3.3 Estimation, Generalization, and MDL

When a discrete model (a partition Γ of the set of nouns $\mathcal{N}$ in our present context) is fixed, and the estimation problem involves only the estimation of probability parameters, the classic maximum-likelihood estimation (MLE) is known to be satisfactory. In particular, the estimation of a word-based model is one such problem, since the partition is fixed and the size of the partition equals $|\mathcal{N}|$. Furthermore, for a fixed discrete model, it is known that MLE coincides with MDL: Given data $S = \{x_i : i = 1, \ldots, m\}$, MLE estimates parameter $\hat{P}$, which maximizes the likelihood with respect to the data; that is:

$$\hat{P} = \arg\max_P \prod_{i=1}^{m} P(x_i). \tag{13}$$

It is easy to see that $\hat{P}$ also satisfies:

$$\hat{P} = \arg\min_P \sum_{i=1}^{m} -\log P(x_i). \tag{14}$$

This is nothing but the MDL estimate in this case, since $\sum_{i=1}^{m} -\log P(x_i)$ is the data description length.

When the estimation problem involves model selection, i.e., the choice of a tree cut in the present context, MDL's behavior significantly deviates from that of MLE. This is because MDL insists on minimizing the sum total of the data description length *and* the model description length, while MLE is still equivalent to minimizing the data description length only. So, for our problem of estimating a tree cut model, MDL tends to select a model that is reasonably simple yet fits the data quite well, whereas the model selected by MLE will be a word-based model (or a tree cut model equivalent to the word-based model[8]), as it will always manage to fit the data.

In statistical terms, the superiority of MDL as an estimation method is related to the fact we noted earlier that even though MLE can provide the best fit to the given data, the estimation accuracy of the parameters is poor, when applied on a sample of modest size, as there are too many parameters to estimate. MLE is likely to estimate most parameters to be zero, and thus suffers from the data sparseness problem. Note in Table 4, that MDL avoids this problem by taking into account the model complexity as well as the fit to the data.

MDL stipulates that the model with the minimum description length should be selected both for data compression and estimation. This intimate connection between estimation and data compression can also be thought of as that between estimation and generalization, since in order to compress information, generalization is necessary. In our current problem, this corresponds to the generalization of individual nouns present in case frame instances in the data as classes of nouns present in a given thesaurus. For example, given the thesaurus in Figure 3 and frequency data in Figure 1, we would

7 The process of finding the MDL model tends to be computationally demanding and is often intractable. When the model class under consideration is restricted to tree structures, however, dynamic programming is often applicable and the MDL model can be efficiently found. For example, Rissanen (1995) has devised an algorithm for learning decision trees.

8 Consider, for example, the case when the co-occurrence data is given as $f(\text{swallow}) = 2, f(\text{crow}) = 2, f(\text{eagle}) = 2, f(\text{bird}) = 2$ for the problem in Section 2.

like our system to judge that the class BIRD and the noun *bee* can be the subject slot of the verb *fly*. The problem of deciding whether to stop generalizing at BIRD and *bee*, or generalizing further to ANIMAL has been addressed by a number of authors (Webster and Marcus 1989; Velardi, Pazienza, and Fasolo 1991; Nomiyama 1992). Minimization of the total description length provides a disciplined criterion to do this.

A remarkable fact about MDL is that theoretical findings have indeed verified that MDL, as an estimation strategy, is near optimal in terms of the rate of convergence of its estimated models to the true model as data size increases. When the true model is included in the class of models considered, the models selected by MDL converge to the true model at the rate of $O(\frac{k^* \cdot \log |S|}{2 \cdot |S|})$, where k^* is the number of parameters in the true model, and $|S|$ the data size, which is near optimal (Barron and Cover 1991; Yamanishi 1992).

Thus, in the current problem, MDL provides (a) a way of smoothing probability parameters to solve the data sparseness problem, and at the same time, (b) a way of generalizing nouns in the data to noun classes of an appropriate level, both as a corollary to the near optimal estimation of the distribution of the given data.

3.4 The Bayesian Interpretation of MDL and the Choice of Encoding Scheme

There is a Bayesian interpretation of MDL: MDL is essentially equivalent to the "posterior mode" in the Bayesian terminology (Rissanen 1989). Given data S and a number of models, the Bayesian estimator (posterior mode) selects a model $\hat{M}$ that maximizes the posterior probability:

$$\hat{M} = \arg\max_M (P(M) \cdot P(S \mid M)) \tag{15}$$

where $P(M)$ denotes the prior probability of the model M and $P(S \mid M)$ the probability of observing the data S given M. Equivalently, $\hat{M}$ satisfies

$$\hat{M} = \arg\min_M (-\log P(M) - \log P(S \mid M)). \tag{16}$$

This is equivalent to the MDL estimate, if we take $-\log P(M)$ to be the model description length. Interpreting $-\log P(M)$ as the model description length translates, in the Bayesian estimation, to assigning larger prior probabilities on simpler models, since it is equivalent to assuming that $P(M) = (\frac{1}{2})^{l(M)}$, where $l(M)$ is the description length of M. (Note that if we assign uniform prior probability $P(M)$ to all models M, then (15) becomes equivalent to (13), giving the maximum-likelihood estimate.)

Recall, that in our definition of parameter description length, we assign a shorter parameter description length to a model with a smaller number of parameters k, which admits the above interpretation. As for the model description length (for tree cuts) we assigned an equal code length to each tree cut, which translates to placing no bias on any cut. We could have employed a different coding scheme assigning shorter code lengths to cuts nearer the root. We chose not to do so partly because, for sufficiently large sample sizes, the parameter description length starts dominating the model description length anyway.

Another important property of the definition of description length is that it affects not only the effective prior probabilities on the models, but also the procedure for computing the model minimizing the measure. Indeed, our definition of model description length was chosen to be compatible with the dynamic programming technique, namely, its calculation is performable locally for each subtree. For a different choice of coding scheme, it is possible that a simple and efficient MDL algorithm like

Find-MDL may not exist. We believe that our choice of model description length is derived from a natural encoding scheme with reasonable interpretation as Bayesian prior, and at the same time allows an efficient algorithm for finding a model with the minimum description length.

3.5 The Uniform Distribution Assumption and the Level of Generalization

The uniform distribution assumption made in (4), namely that all nouns belonging to a class contained in the tree cut model are assigned the same probability, seems to be rather stringent. If one were to insist that the model be exactly accurate, then it would seem that the true model would be the word-based model resulting from no generalization at all. If we allow approximations, however, it is likely that some reasonable tree cut model with the uniform probability assumption will be a good approximation of the true distribution; in fact, a best model for a given data size. As we remarked earlier, as MDL balances between the fit to the data and the simplicity of the model, one can expect that the model selected by MDL will be a reasonable compromise.

Nonetheless, it is still a shortcoming of our model that it contains an oversimplified assumption, and the problem is especially pressing when rare words are involved. Rare words may not be observed at a slot of interest in the data simply because they are rare, and not because they are unfit for that particular slot.[9] To see how rare is too rare for our method, consider the following example.

Suppose that the class BIRD contains 10 words, *bird, swallow, crow, eagle, parrot, waxwing*, etc. Consider co-occurrence data having 8 occurrences of *bird*, 2 occurrences of *swallow*, 1 occurrence of *crow*, 1 occurrence of *eagle*, and 0 occurrence of all other words, as part of, say, 100 data obtained for the subject slot of verb *fly*. For this data set, our method would select the model that generalizes *bird, swallow*, etc. to the class BIRD, since the sum of the data and parameter description lengths for the BIRD subtree is $76.57 + 3.32 = 79.89$ if generalized, and $53.73 + 33.22 = 86.95$ if not generalized. For comparison, consider the data with 10 occurrences of *bird*, 3 occurrences of *swallow* and 1 occurrence of *crow*, and 0 occurrence of all other words, also as part of 100 data for the subject slot of *fly*. In this case, our method would select the model that stops generalizing at *bird, swallow, eagle*, etc., because the description length for the same subtree now is $86.22 + 3.32 = 89.54$ if generalized, and $55.04 + 33.22 = 88.26$ if not generalized. These examples seem to indicate that our MDL-based method would choose to generalize, even when there are relatively large differences in frequencies of words within a class, but knows enough to stop generalizing when the discrepancy in frequencies is especially noticeable (relative to the given sample size).

4. Experimental Results

4.1 Experiment 1: A Qualitative Evaluation

We applied our generalization method to large corpora and inspected the obtained tree cut models to see if they agreed with human intuition. In our experiments, we extracted verbs and their case frame slots (*verb, slot_name, slot_value* triples) from the tagged texts of the *Wall Street Journal* corpus (ACL/DCI CD-ROM1) consisting of 126,084 sentences, using existing techniques (specifically, those in Smadja [1993]), then

9 There are several possible measures that one could take to address this issue, including the incorporation of absolute frequencies of the words (inside and outside the particular slot in question). This is outside the scope of the present paper, and we simply refer the interested reader to one possible approach (Abe and Li 1996).

Table 6
Example input data (for the direct object slot of *eat*).

eat arg2 food	3	eat arg2 lobster	1	eat arg2 seed	1
eat arg2 heart	2	eat arg2 liver	1	eat arg2 plant	1
eat arg2 sandwich	2	eat arg2 crab	1	eat arg2 elephant	1
eat arg2 meal	2	eat arg2 rope	1	eat arg2 seafood	1
eat arg2 amount	2	eat arg2 horse	1	eat arg2 mushroom	1
eat arg2 night	2	eat arg2 bug	1	eat arg2 ketchup	1
eat arg2 lunch	2	eat arg2 bowl	1	eat arg2 sawdust	1
eat arg2 snack	2	eat arg2 month	1	eat arg2 egg	1
eat arg2 jam	2	eat arg2 effect	1	eat arg2 sprout	1
eat arg2 diet	1	eat arg2 debt	1	eat arg2 nail	1
eat arg2 pizza	1	eat arg2 oyster	1		

applied our method to generalize the *slot_values*. Table 6 shows some example triple data for the direct object slot of the verb *eat*.

There were some extraction errors present in the data, but we chose not to remove them, because in general there will always be extraction errors and realistic evaluation should leave them in.

When generalizing, we used the noun taxonomy of WordNet (version 1.4) (Miller 1995) as our thesaurus. The noun taxonomy of WordNet has a structure of directed acyclic graph (DAG), and its nodes stand for a word sense (a concept) and often contain several words having the same word sense. WordNet thus deviates from our notion of thesaurus—a tree in which each leaf node stands for a noun, each internal node stands for the class of nouns below it, and a noun is uniquely represented by a leaf node—so we took a few measures to deal with this.

First, we modified our algorithm Find-MDL so that it can be applied to a DAG; now, Find-MDL effectively copies each subgraph having multiple parents (and its associated data) so that the DAG is transformed to a tree structure. Note that with this modification it is no longer guaranteed that the output model is optimal. Next, we dealt heuristically with the issue of word-sense ambiguity by equally dividing the observed frequency of a noun between all the nodes containing that noun. Finally, when an internal node contained nouns actually occurring in the data, we assigned the frequencies of all the nodes below it to that internal node, and excised the whole subtree (subgraph) below it. The last of these measures, in effect, defines the "starting cut" of the thesaurus from which to begin generalizing. Since (word senses of) nouns that occur in natural language tend to concentrate in the middle of a taxonomy, the starting cut given by this method usually falls around the middle of the thesaurus.[10]

Figure 9 shows the starting cut and the resulting cut in WordNet for the direct object slot of *eat* with respect to the data in Table 6, where ⟨...⟩ denotes a node in WordNet. The starting cut consists of nodes ⟨plant...⟩,⟨food⟩,etc, which are the highest nodes containing values of the direct object slot of *eat*. Since ⟨food⟩ has significantly higher frequencies than its neighbors ⟨solid⟩ and ⟨fluid⟩, the generalization stops there according to MDL. In contrast, the nodes under ⟨life_form...⟩ have relatively small differences in their frequencies, and thus they are generalized to the node ⟨life_form...⟩. The same is true of the nodes under ⟨artifact⟩. Since ⟨...amount...⟩ has a much

10 Cognitive scientists have observed that concepts in the middle of a taxonomy tend to be more important with respect to learning, recognition, and memory, and their linguistic expressions occur more frequently in natural language—a phenomenon known as basic level primacy. See Lakoff (1987).

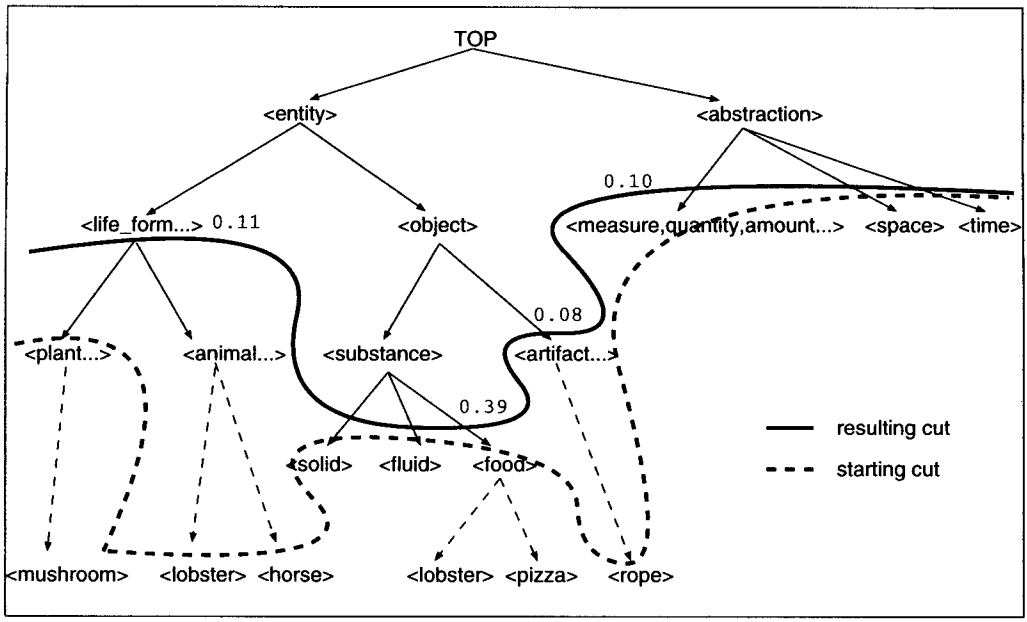

Figure 9
An example generalization result (for the direct object slot of *eat*).

higher frequency than its neighbors ⟨time⟩ and ⟨space⟩, the generalization does not go up higher. All of these results seem to agree with human intuition, indicating that our method results in an appropriate level of generalization.

Table 7 shows generalization results for the direct object slot of *eat* and some other arbitrarily selected verbs, where classes are sorted in descending order of their probability values. (Classes with probabilities less than 0.05 are discarded due to space limitations.)

Table 8 shows the computation time required (on a SPARC "Ultra 1" work station) to obtain the results shown in Table 7. (The computation time for loading the WordNet was excluded since it need be done only once.) Even though the noun taxonomy of WordNet is a large thesaurus containing approximately 50,000 nodes, our method still manages to efficiently generalize case slots using it. The table also shows the average number of levels generalized for each slot, namely, the average number of links between a node in the starting cut and its ancestor node in the resulting cut. (For example, the number of levels generalized for ⟨plant...⟩ is one in Figure 9.) One can see that a significant amount of generalization is performed by our method—the resulting tree cut is about 5 levels higher than the starting cut, on the average.

4.2 Experiment 2: PP-Attachment Disambiguation

Case frame patterns obtained by our method can be used in various tasks in natural language processing. In this paper, we test its effectiveness in a structural (PP-attachment) disambiguation experiment.

Disambiguation Methods. It has been empirically verified that the use of lexical semantic knowledge is effective in structural disambiguation, such as the PP-attachment problem (Hobbs and Bear 1990; Whittemore, Ferrara, and Brunner 1990). There have been

Table 7
Examples of generalization results.

Class	Probability	Example Words
Direct Object of *eat*		
⟨food,nutrient⟩	0.39	pizza, egg
⟨life_form,organism,being,living_thing⟩	0.11	lobster, horse
⟨measure,quantity,amount,quantum⟩	0.10	amount *of*
⟨artifact,article,artefact⟩	0.08	*as if eat* rope
Direct Object of *buy*		
⟨object,inanimate_object,physical_object⟩	0.30	computer, painting
⟨asset⟩	0.10	stock, share
⟨group,grouping⟩	0.07	company, bank
⟨legal_document,legal_instrument,official_document,...⟩	0.05	security, ticket
Direct Object of *fly*		
⟨entity⟩	0.35	airplane, flag, executive
⟨linear_measure,long_measure⟩	0.28	mile
⟨group,grouping⟩	0.08	delegation
Direct Object of *operate*		
⟨group,grouping⟩	0.13	company, fleet
⟨act,human_action,human_activity⟩	0.13	flight, operation
⟨structure,construction⟩	0.12	center
⟨abstraction⟩	0.11	service, unit
⟨possession⟩	0.06	profit, earnings

Table 8
Required computation time and number of generalized levels.

Verb	CPU Time (second)	Average Number of Generalized Levels
eat	1.00	5.2
buy	0.66	4.6
fly	1.11	6.0
operate	0.90	5.0
Average	0.92	5.2

many probabilistic methods proposed in the literature to address the PP-attachment problem using lexical semantic knowledge which, in our view, can be classified into three types.

The first approach (Hindle and Rooth 1991, 1993) takes doubles of the form $(verb, prep)$ and $(noun_1, prep)$, like those in Table 9, as training data to acquire semantic knowledge and judges the attachment sites of the prepositional phrases in quadruples of the form $(verb, noun_1, prep, noun_2)$—e.g., (see, girl, with, telescope)—based on the acquired knowledge. Hindle and Rooth (1991) proposed the use of the lexical association measure calculated based on such doubles. More specifically, they estimate $P(prep \mid verb)$ and $P(prep \mid noun_1)$, and calculate the so-called t-score, which is a measure of the statistical significance of the difference between $P(prep \mid verb)$ and $P(prep \mid noun_1)$. If the t-score indicates that the former probability is significantly larger,

Table 9
Example input data as doubles.

see in
see with
girl with
man with

Table 10
Example input data as triples.

see in park
see with telescope
girl with scarf
see with friend
man with hat

Table 11
Example input data as quadruples and labels.

see girl in park	ADV
see man with telescope	ADV
see girl with scarf	ADN

then the prepositional phrase is attached to *verb*, if the latter probability is significantly larger, it is attached to $noun_1$, and otherwise no decision is made.

The second approach (Sekine et al. 1992; Chang, Luo, and Su 1992; Resnik 1993a; Grishman and Sterling 1994; Alshawi and Carter 1994) takes triples $(verb, prep, noun_2)$ and $(noun_1, prep, noun_2)$, like those in Table 10, as training data for acquiring semantic knowledge and performs PP-attachment disambiguation on quadruples. For example, Resnik (1993a) proposes the use of the selectional association measure calculated based on such triples, as described in Section 2. More specifically, his method compares $\max_{Class_i \ni noun_2} A(Class_i \mid verb, prep)$ and $\max_{Class_i \ni noun_2} A(Class_i \mid noun_1, prep)$ to make disambiguation decisions.

The third approach (Brill and Resnik 1994; Ratnaparkhi, Reynar, and Roukos 1994; Collins and Brooks 1995) receives quadruples $(verb, noun_1, prep, noun_2)$ and labels indicating which way the PP-attachment goes, like those in Table 11, and learns a disambiguation rule for resolving PP-attachment ambiguities. For example, Brill and Resnik, (1994) propose a method they call transformation-based error-driven learning (see also Brill [1995]). Their method first learns IF-THEN type rules, where the IF parts represent conditions like (*prep* is with) and (*verb* is see), and the THEN parts represent transformations from (attach to *verb*) to (attach to $noun_1$), or vice versa. The first rule is always a default decision, and all the other rules indicate transformations (changes of attachment sites) subject to various IF conditions.

We note that, for the disambiguation problem, the first two approaches are basically unsupervised learning methods, in the sense that the training data are merely positive examples for both types of attachments, which could in principle be extracted from pure corpus data with no human intervention. (For example, one could just use unambiguous sentences.) The third approach, on the other hand, is a *supervised* learning method, which requires labeled data prepared by a human being.

Table 12
Number of different types of data.

Training Data	
Average number of doubles per data set	91218.1
Average number of triples per data set	91218.1
Average number of quadruples per data set	21656.6
Test Data	
Average number of quadruples per data set	820.4

The generalization method we propose falls into the second category, although it can also be used as a component in a combined scheme with many of the above methods (see Brill and Resnik [1994], Alshawi and Carter [1994]). We estimate $P(noun_2 \mid verb, prep)$ and $P(noun_2 \mid noun_1, prep)$ from training data consisting of triples, and compare them: If the former exceeds the latter (by a certain margin) we attach it to *verb*, else if the latter exceeds the former (by the same margin) we attach it to $noun_1$.

In our experiments, described below, we compare the performance of our proposed method, which we refer to as MDL, against the methods proposed by Hindle and Rooth (1991), Resnik (1993b), and Brill and Resnik (1994), referred to respectively as LA, SA, and TEL.

Data Set. We used the bracketed corpus of the Penn Treebank (*Wall Street Journal* corpus) (Marcus, Santorini, and Marcinkiewicz 1993) as our data. First we randomly selected one of the 26 directories of the WSJ files as the test data and what remains as the training data. We repeated this process 10 times and obtained 10 sets of data consisting of different training data and test data. We used these 10 data sets to conduct cross-validation as described below.

From the test data in each data set, we extracted $(verb, noun_1, prep, noun_2)$ quadruples using the extraction tool provided by the Penn Treebank called "tgrep." At the same time, we obtained the answer for the PP-attachment site for each quadruple. We did not double-check if the answers provided in the Penn Treebank were actually correct or not. Then from the training data of each data set, we extracted $(verb, prep)$ and $(noun, prep)$ doubles, and $(verb, prep, noun_2)$ and $(noun_1, prep, noun_2)$ triples using tools we developed ourselves. We also extracted quadruples from the training data as before. We then applied 12 heuristic rules to further preprocess the data, which include (1) changing the inflected form of a word to its stem form, (2) replacing numerals with the word *number*, (3) replacing integers between 1,900 and 2,999 with the word *year*, (4) replacing *co.*, *ltd.*, etc. with the words *company*, *limited*, etc.[11] After preprocessing there still remained some minor errors, which we did not remove further, due to the lack of a good method for doing so automatically. Table 12 shows the number of different types of data obtained by the above process.

Experimental Procedure. We first compared the accuracy and coverage for each of the three disambiguation methods based on unsupervised learning: MDL, SA, and LA.

11 The experimental results obtained here are better than those obtained in our preliminary experiment (Li and Abe 1995), in part because we only adopted rule (1) in the past.

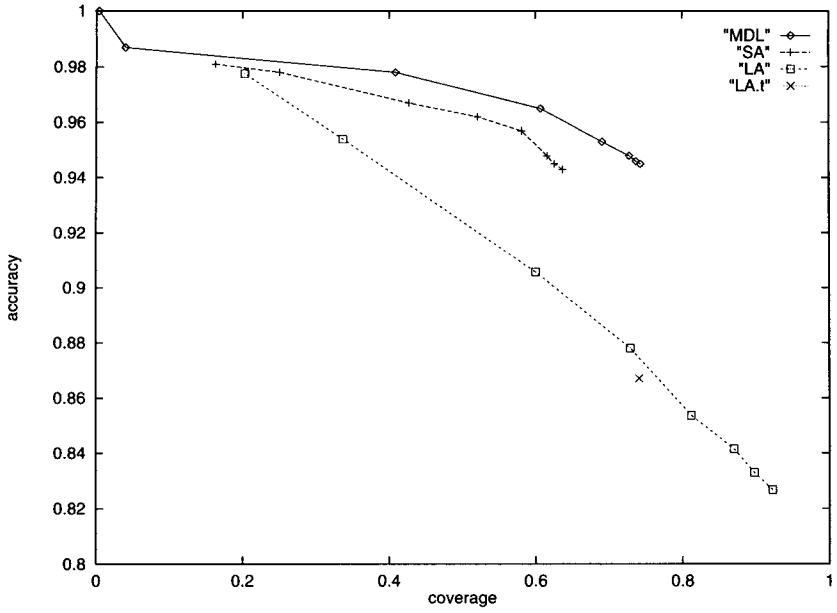

Figure 10
Accuracy-coverage curves for MDL, SA, and LA.

For MDL, we generalized $noun_2$ given $(verb, prep, noun_2)$ and $(noun_1, prep, noun_2)$ triples as training data for each data set, using WordNet as the thesaurus in the same manner as in experiment 1. When disambiguating, we actually compared $P(Class_1 \mid verb, prep)$ and $P(Class_2 \mid noun_1, prep)$, where $Class_1$ and $Class_2$ are classes in the output tree cut models dominating $noun_2$ in place of $P(noun_2 \mid verb, prep)$ and $P(noun_2 \mid noun_1, prep)$.[12] We found that doing so gives a slightly better result. For SA, we employed a somewhat simplified version in which $noun_2$ is generalized given $(verb, prep, noun_2)$ and $(noun_1, prep, noun_2)$ triples using WordNet, and $\max_{Class_i \ni noun_2} A(Class_i \mid verb, prep)$ and $\max_{Class_i \ni noun_2} A(Class_i \mid noun_1, prep)$ are compared for disambiguation: If the former exceeds the latter then the prepositional phrase is attached to *verb*, and otherwise to $noun_1$. For LA, we estimated $P(prep \mid verb)$ and $P(prep \mid noun_1)$ from the training data of each data set and compared them for disambiguation. We then evaluated the results achieved by the three methods in terms of accuracy and coverage. Here, coverage refers to the proportion as a percentage, of the test quadruples on which the disambiguation method could make a decision, and accuracy refers to the proportion of correct decisions among them.

In Figure 10, we plot the accuracy-coverage curves for the three methods. In plotting these curves, the attachment site is determined by simply seeing if the difference between the appropriate measures for the two alternatives, be it probabilities or selectional association values, exceeds a threshold. For each method, the threshold was set successively to 0, 0.01, 0.02, 0.05, 0.1, 0.2, 0.5, and 0.75. When the difference between the two measures is less than a threshold, we rule that no decision can be made. These curves were obtained by averaging over the 10 data sets.

12 Recall that a node in WordNet represents a word sense and not a word; $noun_2$ can belong to several classes in the thesaurus. We thus use $\max_{Class_i \ni noun_2}(P(Class_i \mid verb, prep))$ and $\max_{Class_i \ni noun_2}(P(Class_i \mid noun_1, prep))$ in place of $P(Class_1 \mid verb, prep)$ and $P(Class_2 \mid noun_1, prep)$.

Table 13
Results of PP-attachment disambiguation.

	Coverage(%)	Accuracy(%)
Default	100	56.2
MDL + Default	100	82.2
SA + Default	100	76.7
LA + Default	100	80.7
LA.t + Default	100	78.1
TEL	100	82.4

We also implemented the exact method proposed by Hindle and Rooth (1991), which makes disambiguation judgement using the t-score. Figure 10 shows the result as LA.t, where the threshold for t-score is set to 1.28 (significance level of 90 percent.) From Figure 10 we see that with respect to accuracy-coverage curves, MDL outperforms both SA and LA throughout, while SA is better than LA.

Next, we tested the method of applying a default rule after applying each method. That is, attaching $(prep, noun_2)$ to *verb* for the part of the test data for which no decision was made by the method in question.[13] We refer to these combined methods as MDL+Default, SA+Default, LA+Default, and LA.t+Default. Table 13 shows the results, again averaged over the 10 data sets.

Finally, we used the transformation-based error-driven learning (TEL) to acquire transformation rules for each data set and applied the obtained rules to disambiguate the test data. The average number of obtained rules for a data set was 2,752.3. Table 13 shows the disambiguation result averaged over the 10 data sets. From Table 13, we see that TEL performs the best, edging over the second place MDL+Default by a small margin, and then followed by LA+Default, and SA+Default. Below we discuss further observations concerning these results.

MDL and SA. According to our experimental results, the accuracy and coverage of MDL appear to be somewhat better than those of SA. As Resnik (1993b) pointed out, the use of selectional association $\log \frac{P(C|v,r)}{P(C)}$ seems to be appropriate for cognitive modeling. Our experiments show, however, that the generalization method currently employed by Resnik has a tendency to overfit the data. Table 14 shows example generalization results for MDL (with classes with probability less than 0.05 discarded) and SA. Note that MDL tends to select a tree cut closer to the root of the thesaurus tree. This is probably the key reason why MDL has a wider coverage than SA for the same degree of accuracy. One may be concerned that MDL is "overgeneralizing" here,[14] but as shown in Figure 10, its disambiguation accuracy does not seem to be degraded.

Another problem that must be dealt with concerning SA is how to remove noise (resulting, for example, from erroneous extraction) from the generalization results. Since SA estimates the ratio between two probability values, namely $\frac{P(C|v,r)}{P(C)}$, the generalization result may be lead astray if one of the estimates of $P(C \mid v, r)$ and $P(C)$ is unreliable. For instance, a high estimated value for $\langle$drop, bead, pearl$\rangle$ at *protect against*

13 Interestingly, for the entire data set it is more favorable to attach $(prep, noun_2)$ to $noun_1$, but for what remains after applying LA and MDL, it turns out to be more favorable to attach $(prep, noun_2)$ to *verb*.
14 Note that in Experiment 1, there were more data available, and thus the data were more appropriately generalized.

Table 14
Example generalization results for SA and MDL.

Input			
Verb	Preposition	Noun	Frequency
protect	against	accusation	1
protect	against	damage	1
protect	against	decline	1
protect	against	drop	1
protect	against	loss	1
protect	against	resistance	1
protect	against	squall	1
protect	against	vagary	1

Generalization Result of MDL			
Verb	Preposition	Noun Class	Probability
protect	against	⟨act,human_action,human_activity⟩	0.212
protect	against	⟨phenomenon⟩	0.170
protect	against	⟨psychological_feature⟩	0.099
protect	against	⟨event⟩	0.097
protect	against	⟨abstraction⟩	0.093

Generalization Result of SA			
Verb	Preposition	Noun Class	SA
protect	against	⟨caprice,impulse,vagary,whim⟩	1.528
protect	against	⟨phenomenon⟩	0.899
protect	against	⟨happening,occurrence,natural_event⟩	0.339
protect	against	⟨deterioration,worsening,decline,declination⟩	0.285
protect	against	⟨act,human_action,human_activity⟩	0.260
protect	against	⟨drop,bead,pearl⟩	0.202
protect	against	⟨drop⟩	0.202
protect	against	⟨descent,declivity,fall,decline,downslope⟩	0.188
protect	against	⟨resistor,resistance⟩	0.130
protect	against	⟨underground,resistance⟩	0.130
protect	against	⟨immunity,resistance⟩	0.124
protect	against	⟨resistance,opposition⟩	0.111
protect	against	⟨loss,deprivation⟩	0.105
protect	against	⟨loss⟩	0.096
protect	against	⟨cost,price,terms,damage⟩	0.052

shown in Table 14 is rather odd, and is because the estimate of $P(C)$ is unreliable (too small). This problem apparently costs SA a nonnegligible drop in disambiguation accuracy. In contrast, MDL does not suffer from this problem since a high estimated probability value is only possible with high frequency, which cannot result just from extraction errors. Consider, for example, the occurrence of *car* in the data shown in Figure 8, which has supposedly resulted from an erroneous extraction. The effect of this datum gets washed away, as the estimated probability for VEHICLE, to which *car* has been generalized, is negligible.

On the other hand, SA has a merit not shared by MDL, namely its use of the association ratio factors out the effect of absolute frequencies of words, and focuses

Table 15
Some hard examples for LA.

Attached to *verb*	Attached to $noun_1$
acquire interest in year	acquire interest in firm
buy stock in trade	buy stock in index
ease restriction on export	ease restriction on type
forecast sale for year	forecast sale for venture
make payment on million	make payment on debt
meet standard for resistance	meet standard for car
reach agreement in august	reach agreement in principle
show interest in session	show interest in stock
win verdict in winter	win verdict in case

on their co-occurrence relation. Since both MDL and SA have pros and cons, it would be desirable to develop a methodology that combines the merits of the two methods (cf. Abe and Li [1996]).

MDL and LA. LA makes its disambiguation decision completely ignoring $noun_2$. As Resnik (1993b) pointed out, if we hope to improve disambiguation performance by increasing training data, we need a richer model such as those used in MDL and SA. We found that 8.8% of the quadruples in our entire test data were such that they shared the same *verb*, *prep*, $noun_1$ but had different $noun_2$, and their PP-attachment sites go both ways in the same data, i.e., both to *verb* and to $noun_1$. Clearly, for these examples, the PP-attachment site cannot be reliably determined without knowing $noun_2$. Table 15 shows some of these examples. (We adopted the attachment sites given in the Penn Tree Bank, without correcting apparently wrong judgements.)

MDL and TEL. We chose TEL as an example of the quadruple approach. This method was designed specifically for the purpose of resolving PP-attachment ambiguities, and seems to perform slightly better than ours.

As we remarked earlier, however, the input data required by our method (triples) could be generated automatically from unparsed corpora making use of existing heuristic rules (Brent 1993; Smadja 1993), although for the experiments we report here we used a parsed corpus. Thus it would seem to be easier to obtain more data in the future for MDL and other methods based on unsupervised learning. Also note that our method of generalizing values of a case slot can be used for purposes other than disambiguation.

5. Conclusions

We proposed a new method of generalizing case frames. Our approach of applying MDL to estimate a tree cut model in an existing thesaurus is not limited to just the problem of generalizing values of a case frame slot. It is potentially useful in other natural language processing tasks, such as the problem of estimating n-gram models (Brown et al. 1992) or the problem of semantic tagging (Cucchiarelli and Velardi 1997). We believe that our method has the following merits: (1) it is theoretically sound; (2) it is computationally efficient; (3) it is robust against noise. Our experimental results indicate that the performance of our method is better than, or at least comparable to, existing methods. One of the disadvantages of our method is that its performance

depends on the structure of the particular thesaurus used. This, however, is a problem commonly shared by any generalization method that uses a thesaurus as prior knowledge.

Appendix A: Proof of Proposition 1

Proof

For an arbitrary subtree T' of a thesaurus tree T and an arbitrary tree cut model $M = (\Gamma, \theta)$ of T, let $M_{T'} = (\Gamma_{T'}, \theta_{T'})$ denote the submodel of M that is contained in T'. Also for any sample S and any subtree T' of T, let $S_{T'}$ denote the subsample of S contained in T'. (Note that $M_T = M$, $S_T = S$.) Then define, in general for any submodel $M_{T'}$ and subsample $S_{T'}$, $L(S_{T'} \mid \Gamma_{T'}, \hat{\theta}_{T'})$ to be the data description length of subsample $S_{T'}$ using submodel $M_{T'}$, $L(\hat{\theta}_{T'} \mid \Gamma_{T'})$ to be the parameter description length for the submodel $M_{T'}$, and $L'(M_{T'}, S_{T'})$ to be $L(S_{T'} \mid \Gamma_{T'}, \hat{\theta}_{T'}) + L(\hat{\theta}_{T'} \mid \Gamma_{T'})$. (Note that, when calculating the parameter description length for a submodel, the sample size of the entire sample $|S|$ is used.)

First note that for any (sub)tree T, (sub)model $M_T = (\Gamma_T, \hat{\theta}_T)$ contained in T, and (sub)sample S_T contained in T, and T's child subtrees $T_i : i = 1, \ldots, k$, we have:

$$L(S_T \mid \Gamma_T, \hat{\theta}_T) = \sum_{i=1}^{k} L(S_{T_i} \mid \Gamma_{T_i}, \hat{\theta}_{T_i}) \tag{17}$$

provided that Γ_T is not a single node (root node of T). This follows from the mutual disjointness of the T_i, and the independence of the parameters in the T_i.

We also have, when T is a *proper* subtree of the thesaurus tree:

$$L(\hat{\theta}_T \mid \Gamma_T) = \sum_{i=1}^{k} L(\hat{\theta}_{T_i} \mid \Gamma_{T_i}). \tag{18}$$

Since the number of free parameters of a model in the entire thesaurus tree equals the number of nodes in the model *minus* one due to the stochastic condition (that the probability parameters must sum to one), when T equals the entire thesaurus tree, theoretically the parameter description length for a tree cut model of T should be:

$$\begin{aligned} L(\hat{\theta}_T \mid \Gamma_T) &= L(\hat{\theta} \mid \Gamma) \\ &= \sum_{i=1}^{k} L(\hat{\theta}_{T_i} \mid \Gamma_{T_i}) - \frac{\log |S|}{2} \end{aligned} \tag{19}$$

where $|S|$ is the size of the entire sample. Since the second term $-\frac{\log |S|}{2}$ in (19) is constant once the input sample S is fixed, for the purpose of finding a model with the minimum description length, it is irrelevant. We will thus use the identity (18) both when T is the entire tree and when it is a proper subtree. (This allows us to use the same recursive algorithm, Find-MDL, in all cases.)

It follows from (17) and (18) that the minimization of description length can be done essentially independently for each subtree. Namely, if we let $L'_{min}(M_T, S_T)$ denote the minimum description length (as defined by [17] and [18]) achievable for (sub)model M_T on (sub)sample S_T contained in (sub)tree T, $\hat{P}_S(\eta)$ the MLE estimate for node η

using the entire sample S, and root(T) the root node of tree T, then we have:

$$L'_{min}(M_T, S_T) = \min \left\{ \sum_{i=1}^{k} L'_{min}(M_{T_i}, S_{T_i}), \right. \\ \left. L'(([\text{root}(T)], [\hat{P}_S(\text{root}(T))]), S_T) \right\} \quad (20)$$

The rest of the proof proceeds by induction. First, when T is of a single leaf node, the submodel consisting solely of the node and the MLE of the generation probability for the class represented by T is returned, which is clearly a submodel with minimum description length in the subtree T. Next, inductively assume that Find-MDL(T') correctly outputs a (sub)model with the minimum description length for any tree T' of size less than n. Then, given a tree T of size n whose root node has at least two children, say $T_i : i = 1, \ldots, k$, for each T_i, Find-MDL(T_i) returns a (sub)model with the minimum description length by the inductive hypothesis. Then, since (20) holds, whichever way the if-clause on lines 8, 9 of Find-MDL evaluates to, what is returned on line 11 or line 13 will still be a (sub)model with the minimum description length, completing the inductive step.

It is easy to see that the running time of the algorithm is linear in both the number of leaf nodes of the input thesaurus tree and the input sample size. ■

Acknowledgments

We are grateful to K. Nakamura and T. Fujita of NEC C&C Res. Labs. for their constant encouragement. We thank K. Yaminishi and J. Takeuchi of C&C Res. Labs. for their suggestions and comments. We thank T. Futagami of NIS for his programming efforts. We also express our special appreciation to the two anonymous reviewers who have provided many valuable comments. We acknowledge the ACL for providing the ACL/DCI CD-ROM, LDC of the University of Pennsylvania for providing the Penn Treebank corpus data, and Princeton University for providing WordNet, and E. Brill and P. Resnik for providing their PP-attachment disambiguation program.

References

Abe, Naoki and Hang Li. 1996. Learning word association norms using tree cut pair models. *Proceedings of the Thirteenth International Conference on Machine Learning*, pages 3–11.

Almuallim, Hussein, Yasuhiro Akiba, Takefumi Yamazaki, Akio Yokoo, and Shigeo Kaneda. 1994. Two methods for ALT-J/E translation rules from examples and a semantic hierarchy. *Proceedings of the Fifteenth International Conference on Computational Linguistics*, pages 57–63.

Alshawi, Hiyan and David Carter. 1994. Training and scaling preference functions for disambiguation. *Computational Linguistics*, 20(4):635–648.

Barron, Andrew R. and Thomas M. Cover. 1991. Minimum complexity density estimation. *IEEE Transaction on Information Theory*, 37(4):1034–1054.

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

Brent, Michael R. and Timothy A. Cartwright. 1996. Distributional regularity and phonotactic constraints are useful for segmentation. *Cognition*, 61:93–125.

Brent, Michael R., Sreerama K. Murthy, and Andrew Lundberg. 1995. Discovering morphemic suffixes: A case study in minimum description length induction. *Proceedings of the Fifth International Workshop on Artificial Intelligence and Statistics*.

Brill, Eric. 1995. Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging. *Computational Linguistics*, 21(4):543–565.

Brill, Eric and Philip Resnik. 1994. A rule-based approach to prepositional phrase attachment disambiguation.

Proceedings of the Fifteenth International Conference on Computational Linguistics, pages 1198–1204.

Briscoe, Ted and John Carroll. 1997. Automatic extraction of subcategorization from corpora. *Proceedings of the Fifth Conference on Applied Natural Language Processing*.

Brown, Peter F., Vincent J. Della Pietra, Peter V. deSouza, Jenifer C. Lai, and Robert L. Mercer. 1992. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):283–298.

Cartwright, Timothy A. and Michael R. Brent. 1994. Segmenting speech without a lexicon: The roles of phonotactics and speech source. *Proceedings of the First Meeting of the ACL Special Interest Group in Computational Phonology*, pages 83–90.

Chang, Jing-Shin, Yih-Fen Luo, and Keh-Yih Su. 1992. GPSM: A generalized probabilistic semantic model for ambiguity resolution. *Proceedings of the 30th Annual Meeting*, pages 177–184. Association for Computational Linguistics.

Collins, Michael and James Brooks. 1995. Prepositional phrase attachment through a backed-off model. *Proceedings of the Third Workshop on Very Large Corpora*.

Cover, Thomas M. and Joy A. Thomas. 1991. *Elements of Information Theory*. John Wiley & Sons Inc., New York.

Cucchiarelli, Alessandro and Paola Velardi. 1997. Automatic selection of class labels from a thesaurus for an effective semantic tagging of corpora. *Proceedings of the Fifth Conference on Applied Natural Language Processing*, pages 380–387.

Dagan, Ido, Shaul Marcus, and Shaul Makovitch. 1992. Contextual word similarity and estimation from sparse data. *Proceedings of the 30th Annual Meeting*, pages 164–171. Association for Computational Linguistics.

Dagan, Ido, Fernando Pereira, and Lillian Lee. 1994. Similarity-based estimation of word cooccurrence probabilities. *Proceedings of the 32nd Annual Meeting*, pages 272–278. Association for Computational Linguistics.

Ellison, T. Mark. 1991. Discovering planar segregations. *Proceedings of AAAI Spring Symposium on Machine Learning of Natural Language and Ontology*, pages 42–47.

Ellison, T. Mark. 1992. Discovering vowel harmony. In Walter Daelmans and David Powers, editors, *Background and Experiments in Machine Learning of Natural Language*, pages 205–207.

Framis, Francesc Ribas. 1994. An experiment on learning appropriate selectional restrictions from a parsed corpus. *Proceedings of the Fifteenth International Conference on Computational Linguistics*, pages 769–774.

Grefenstette, Gregory. 1994. *Explorations in Automatic Thesaurus Discovery*. Kluwer Academic Publishers, Boston.

Grishman, Ralph and John Sterling. 1992. Acquisition of selectional patterns. *Proceedings of the Fourteenth International Conference on Computational Linguistics*, pages 658–664.

Grishman, Ralph and John Sterling. 1994. Generalizing automatically generated selectional patterns. *Proceedings of the Fifteenth International Conference on Computational Linguistics*, pages 742–747.

Grunwald, Peter. 1996. A minimum description length approach to grammar inference. In S. Wemter, E. Riloff, and G. Scheler, editors, *Symbolic, Connectionist and Statistical Approaches to Learning for Natural Language Processing, Lecture Note in AI*. Springer Verlag, pages 203–216.

Hindle, Donald and Mats Rooth. 1991. Structural ambiguity and lexical relations. *Proceedings of the 29th Annual Meeting*, pages 229–236. Association for Computational Linguistics.

Hindle, Donald and Mats Rooth. 1993. Structural ambiguity and lexical relations. *Computational Linguistics*, 19(1):103–120.

Hobbs, Jerry R. and John Bear. 1990. Two principles of parse preference. *Proceedings of the Thirteenth International Conference on Computational Linguistics*, pages 162–167.

Lakoff, George. 1987. *Women, Fire, and Dangerous Things: What Categories Reveal about the Mind*. The University of Chicago Press.

Li, Hang and Naoki Abe. 1995. Generalizing case frames using a thesaurus and the MDL principle. *Proceedings of Recent Advances in Natural Language Processing*, pages 239–248.

Li, Hang and Naoki Abe. 1996. Learning dependencies between case frame slots. *Proceedings of the Sixteenth International Conference on Computational Linguistics*, pages 10–15.

Manning, Christopher D. 1992. Automatic acquisition of a large subcategorization dictionary from corpora. *Proceedings of the 30th Annual Meeting*, pages 235–242. Association for Computational Linguistics.

Marcus, Mitchell P., Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The

Penn Treebank. *Computational Linguistics*, 19(1):313–330.
Miller, George A. 1995. WordNet: A lexical database for English. *Communications of the ACM*, pages 39–41.
Nomiyama, Hiroshi. 1992. Machine translation by case generalization. *Proceedings of the Fourteenth International Conference on Computational Linguistics*, pages 714–720.
Pereira, Fernando, Naftali Tishby, and Lillian Lee. 1993. Distributional clustering of English words. *Proceedings of the 31st Annual Meeting*, pages 183–190. Association for Computational Linguistics.
Quinlan, J. Ross and Ronald L. Rivest. 1989. Inferring decision trees using the minimum description length principle. *Information and Computation*, 80:227–248.
Ratnaparkhi, Adwait, Jeff Reynar, and Salim Roukos. 1994. A maximum entropy model for prepositional phrase attachment. *Proceedings of ARPA Workshop on Human Language Technology*, pages 250–255.
Resnik, Philip. 1992. WordNet and distributional analysis: A class-based approach to lexical discovery. *Proceedings of AAAI Workshop on Statistically-based NLP Techniques*.
Resnik, Philip. 1993a. *Selection and Information: A Class-based Approach to Lexical Relationships*. Ph.D. Thesis, Univ. of Pennsylvania.
Resnik, Philip. 1993b. Semantic classes and syntactic ambiguity. *Proceedings of ARPA Workshop on Human Language Technology*.
Rissanen, Jorma. 1978. Modeling by shortest data description. *Automatic*, 14:37–38.
Rissanen, Jorma. 1983. A universal prior for integers and estimation by minimum description length. *The Annals of Statistics*, 11(2):416–431.
Rissanen, Jorma. 1984. Universal coding, information, predication and estimation. *IEEE Transaction on Information Theory*, 30(4):629–636.
Rissanen, Jorma. 1986. Stochastic complexity and modeling. *The Annals of Statistics*, 14(3):1080–1100.
Rissanen, Jorma. 1989. *Stochastic Complexity in Statistical Inquiry*. World Scientific Publishing Co., Singapore.
Rissanen, Jorma. 1995. Stochastic complexity in learning. *Proceedings of the Second European Conference on Computational Learning Theory (Euro Colt'95)*, pages 196–210.
Ristad, Eric Sven and Robert G. Thomas. 1995. New techniques for context modeling. *Proceedings of the 33rd Annual Meeting*. Association for Computational Linguistics.
Schwarz, G. 1978. Estimation of the dimension of a model. *Annals of Statistics*, 6:416–446.
Sekine, Satoshi, Jeremy J. Carroll, Sofia Ananiadou, and Jun'ichi Tsujii. 1992. Automatic learning for semantic collocation. *Proceedings of the Third Conference on Applied Natural Language Processing*, pages 104–110.
Smadja, Frank. 1993. Retrieving collocations from text: Xtract. *Computational Linguistics*, 19(1):143–177.
Solomonoff, R.J. 1964. A formal theory of inductive inference 1 and 2. *Information and Control*, 7:1–22;224–254.
Stolcke, Andreas and Stephen Omohundro. 1994. Inducing probabilistic grammars by bayesian model merging. In Rafael C. Carrasco and Jose Oncina, editors, *Grammatical Inference and Applications*. Springer Verlag, pages 106–118.
Tanaka, Hideki. 1994. Verbal case frame acquisition from a bilingual corpus: Gradual knowledge acquisition. *Proceedings of the Fifteenth International Conference on Computational Linguistics*, pages 727–731.
Tanaka, Hideki. 1996. Decision tree learning algorithm with structured attributes: Application to verbal case frame acquisition. *Proceedings of the Sixteenth International Conference on Computational Linguistics*, pages 943–948.
Utsuro, Takehito and Yuji Matsumoto. 1997. Learning probabilistic subcategorization preference by identifying case dependencies and optimal noun class generalization level. *Proceedings of the Fifth Conference on Applied Natural Language Processing*, pages 364–371.
Utsuro, Takehito, Yuji Matsumoto, and Makoto Nagao. 1992. Lexical knowledge acquisition from bilingual corpora. *Proceedings of the Fourteenth International Conference on Computational Linguistics*, pages 581–587.
Velardi, Paola, Maria Teresa Pazienza, and Michela Fasolo. 1991. How to encode semantic knowledge: A method for meaning representation and computer-aided acquisition. *Computational Linguistics*, 17(2):153–170.
Wallace, C. and D. M. Boulton. 1968. An information measure for classification. *Computer Journal*, 11:185–195.
Wallace, C. and P. Freeman. 1992. Single-factor analysis by minimum message length estimation. *Journal of Royal Statistical Society, B*, 54:195–209.

Webster, Mort and Mitch Marcus. 1989. Automatic acquisition of the lexical semantics of verbs from sentence frames. *Proceedings of the 27th Annual Meeting*, pages 177–184. Association for Computational Linguistics.

Whittemore, Greg, Kathleen Ferrara, and Hans Brunner. 1990. Empirical study of predictive powers of simple attachment schemes for post-modifier prepositional phrases. *Proceedings of the 28th Annual Meeting*, pages 23–30. Association for Computational Linguistics.

Yamanishi, Kenji. 1992. A learning criterion for stochastic rules. *Machine Learning*, 9:165–203.

Yarowsky, David. 1992. Word-sense disambiguation using statistical models of Roget's categories trained on large corpora. *Proceedings of the fourteenth International Conference on Computational Linguistics*, pages 454–460.

Yarowsky, David. 1994. Decision lists for lexical ambiguity resolution: Application to accent restoration in Spanish and French. *Proceedings of the 32nd Annual Meeting*, pages 88–95. Association for Computational Linguistics.