Clustering Words with the MDL Principle

Hang Li and Naoki Abe
Theory NEC Laboratory, RWCP*
c/o C&C Research Laboratories, NEC
4-1-1 Miyazaki Miyamae-ku, Kawasaki, 216 Japan
{lihang,abe}@sbl.cl.nec.co.jp

Abstract

We address the problem of automatically constructing a thesaurus by clustering words based on corpus data. We view this problem as that of estimating a joint distribution over the Cartesian product of a partition of a set of nouns and a partition of a set of verbs, and propose a learning algorithm based on the Minimum Description Length (MDL) Principle for such estimation. We empirically compared the performance of our method based on the MDL Principle against the Maximum Likelihood Estimator in word clustering, and found that the former outperforms the latter. We also evaluated the method by conducting pp-attachment disambiguation experiments using an automatically constructed thesaurus. Our experimental results indicate that such a thesaurus can be used to improve accuracy in disambiguation.

1 Introduction

Recently various methods for automatically constructing a thesaurus (hierarchically clustering words) based on corpus data have been proposed (Hindle, 1990; Brown et al., 1992; Pereira et al., 1993; Tokunaga et al., 1995). The realization of such an automatic construction method would make it possible to a) save the cost of constructing a thesaurus by hand, b) do away with subjectivity inherent in a hand made thesaurus, and c) make it easier to adapt a natural language processing system to a new domain. In this paper, we propose a new method for automatic construction of thesauri. Specifically, we view the problem of automatically clustering words as that of estimating a joint distribution over the Cartesian product of a partition of a set of nouns (in general, any set of words) and a partition of a set of verbs (in general, any set of words), and propose an estimation algorithm using simulated annealing with an energy function based on the Minimum Description Length (MDL) Principle. The MDL Principle is a well-motivated and theoretically sound principle for data compression and estimation in information theory and statistics. As a method of statistical estimation MDL is guaranteed to be near optimal.

We empirically evaluated the effectiveness of our method. In particular, we compared the performance of an MDL-based simulated annealing algorithm in hierarchical word clustering against that of one based on the Maximum Likelihood Estimator (MLE, for short). We found that the MDL-based method performs better than the MLE-based method. We also evaluated our method by conducting pp-attachment disambiguation experiments using a thesaurus automatically constructed by it and found that disambiguation results can be improved.

Since some words never occur in a corpus, and thus cannot be reliably classified by a method solely based on corpus data, we propose to combine the use of an automatically constructed thesaurus and a hand made thesaurus in disambiguation. We conducted some experiments in order to test the effectiveness of this strategy. Our experimental results indicate that combining an automatically constructed thesaurus and a hand made thesaurus widens the *coverage*[1] of our disambiguation method, while maintaining high *accuracy*[2].

2 The Problem Setting

A method of constructing a thesaurus based on corpus data usually consists of the following three steps: (i) Extract co-occurrence data (e.g. case frame data, adjacency data) from a corpus, (ii) Starting from a single class (or each word composing its own class), divide (or merge) word classes

*Real World Computing Partership

[1] 'Coverage' refers to the proportion (in percentage) of test data for which the disambiguation method can make a decision.

[2] 'Accuracy' refers to the success rate, given that the disambiguation method makes a decision.

based on the co-occurrence data using some similarity (distance) measure. (The former approach is called 'divisive', the latter 'agglomerative'.) (iii) Repeat step (ii) until some stopping condition is met, to construct a thesaurus (tree). The method we propose here consists of the same three steps.

Suppose available to us are frequency data (co-occurrence data) between verbs and their case slot values extracted from a corpus (step (i)). We then view the problem of clustering words as that of estimating a probabilistic model (representing a probability distribution) that generates such data . We assume that the target model can be defined in the following way. First, we define a noun partition $\mathcal{P}_{\mathcal{N}}$ over a given set of nouns $\mathcal{N}$ and a verb partion $\mathcal{P}_{\mathcal{V}}$ over a given set of verbs $\mathcal{V}$. A noun partition is any set $\mathcal{P}_{\mathcal{N}}$ satisfying $\mathcal{P}_{\mathcal{N}} \subseteq 2^{\mathcal{N}}$, $\cup_{C_i \in \mathcal{P}_{\mathcal{N}}} C_i = \mathcal{N}$ and $\forall C_i, C_j \in \mathcal{P}_{\mathcal{N}}, C_i \cap C_j = \emptyset$. A verb partition $\mathcal{P}_{\mathcal{V}}$ is defined analogously. In this paper, we call a member of a noun partition 'a noun cluster', and a member of a verb partition a 'verb cluster'. We refer to a member of the Cartesian product of a noun partition and a verb partition ($\in \mathcal{P}_{\mathcal{N}} \times \mathcal{P}_{\mathcal{V}}$) simply as 'a cluster'. We then define a probabilistic model (a joint distribution), written $P(C_n, C_v)$, where random variable C_n assumes a value from a *fixed* noun partition $\mathcal{P}_{\mathcal{N}}$, and C_v a value from a *fixed* verb partition $\mathcal{P}_{\mathcal{V}}$. Within a given cluster, we assume that each element is generated with equal probability, i.e.,

$$\forall n \in C_n, \forall v \in C_v, P(n, v) = \frac{P(C_n, C_v)}{|C_n \times C_v|} \quad (1)$$

In this paper, we assume that the observed data are generated by a model belonging to the class of models just described, and select a model which best explains the data. As a result of this, we obtain both noun clusters and verb clusters. This problem setting is based on the intuitive assumption that similar words occur in the same context with roughly equal likelihood, as is made explicit in equation (1). Thus selecting a model which best explains the given data is equivalent to finding the most appropriate classification of words based on their co-occurrence.

3 Clustering with MDL

We now turn to the question of what strategy (or criterion) we should employ for estimating the best model. Our choice is the *MDL* (Minimum Description Length) principle (Rissanen, 1989), a well-known principle of data compression and statistical estimation from information theory. MDL stipulates that the best probability model for given data is that model which requires the least code length for encoding of the model itself, as well as the given data relative to it[3]. We refer to the code length for the model as 'the model description length' and that for the data 'the data description length.'

We apply MDL to the problem of estimating a model consisting of a pair of partitions as described above. In this context, a model with less clusters tends to be simpler (in terms of the number of parameters), but also tends to have a poorer fit to the data. In contrast, a model with more clusters is more complex, but tends to have a better fit to the data. Thus, there is a trade-off relationship between the simplicity of a model and the goodness of fit to the data. The model description length quantifies the simplicity (complexity) of a model, and the data description length quantifies the fit to the data. According to MDL, the model which minimizes the sum total of the two types of description lengths should be selected.

In what follows, we will describe in detail how the description length is to be calculated in our current context, as well as our simulated annealing algorithm based on MDL.

3.1 Calculating Description Length

We will now describe how the description length for a model is calculated. Recall that each model is specified by the Cartesian product of a partition of nouns and a partition of verbs, and a number of parameters for them. Here we let k_n denote the size of the noun partition, and k_v the size of the verb partition. Then, there are $k_n \cdot k_v - 1$ free parameters in a model.

Given a model M and data S, its total description length $L(M)$[4] is computed as the sum of the model description length $L_{mod}(M)$, the description length of its parameters $L_{par}(M)$, and data description length $L_{dat}(M)$. (We often refer to $L_{mod}(M) + L_{par}(M)$ as the model description length). Namely,

$$L(M) = L_{mod}(M) + L_{par}(M) + L_{dat}(M) \quad (2)$$

We employ the 'binary noun clustering method', in which k_v is fixed at $|\mathcal{V}|$ and we are to decide whether $k_n = 1$ or $k_n = 2$, which is then to be applied recursively to the clusters thus obtained. This is as if we view the nouns as entities and the verbs as features and cluster the entities based on their features. Since there are $2^{|\mathcal{N}|}$ subsets of the set of nouns $\mathcal{N}$, and for each 'binary' noun partition we have two different subsets (a special case of which is when one subset is $\mathcal{N}$ and the other the empty set $\emptyset$), the number of possible binary noun partitions is $2^{|\mathcal{N}|}/2 = 2^{|\mathcal{N}|-1}$. Thus for each binary noun partition we need $\log 2^{|\mathcal{N}|-1} = |\mathcal{N}| - 1$ bits[5] to describe it.[6] Hence $L_{mod}(M)$ is calculated

[3] We refer the interested reader to (Li and Abe, 1995) for explanation of rationals behind using the MDL principle in natural language processing.

[4] $L(M)$ depends on S, but we will leave S implicit.

[5] Throughout the paper 'log' denotes the logarithm to the base 2.

[6] For further explanation, see (Quinlan and Rivest, 1989).

as[7]

$$L_{mod}(M) = |\mathcal{N}| - 1 \qquad (3)$$

$L_{par}(M)$, often referred to as the parameter description length, is calculated by,

$$L_{par}(M) = \frac{k_n \cdot k_v - 1}{2} \cdot \log |S| \qquad (4)$$

where $|S|$ denotes the input data size, and $k_n \cdot k_v - 1$ is the number of (free) parameters in the model. It is known that using $\log \sqrt{|S|} = \frac{\log |S|}{2}$ bits to describe each of the parameters will (approximately) minimize the description length (Rissanen, 1989). Finally, $L_{dat}(M)$ is calculated by

$$L_{dat}(M) = - \sum_{(n,v) \in S} f(n,v) \cdot \log \hat{P}(n,v) \qquad (5)$$

where $f(n,v)$ denotes the observed frequency of the noun verb pair (n,v), and $\hat{P}(n,v)$ the estimated probability of (n,v), which is calculated as follows.

$$\forall n \in C_n, \forall v \in C_v, \hat{P}(n,v) = \frac{\hat{P}(C_n, C_v)}{|C_n \times C_v|} \qquad (6)$$

$$\hat{P}(C_n, C_v) = \frac{f(C_n, C_v)}{|S|} \qquad (7)$$

where $f(C_n, C_v)$ denotes the observed frequency of the noun verb pairs belonging to cluster (C_n, C_v).

With the description length of a model defined in the above manner, we wish to select a model having the minimum description length and output it as the result of clustering. Since the model description length L_{mod} is the same for each model, in practice we only need to calculate and compare $L'(M) = L_{par}(M) + L_{dat}(M)$.

3.2 A Simulated Annealing-based Algorithm

We could in principle calculate the description length for each model and select a model with the minimum description length, if computation time were of no concern. However, since the number of probabilistic models under consideration is super exponential, this is not feasible in practice. We employ the 'simulated annealing technique' to deal with this problem. Figure 1 shows our (divisive) clustering algorithm[8].

4 Advantages of Our Method

In this section, we elaborate on the merits of our method.

In statistical natural language processing, usually the number of parameters in a probabilistic model to be estimated is very large, and therefore such a model is difficult to estimate with a reasonable data size that is available in practice. (This problem is usually referred to as the 'data sparseness problem'.) We could smooth the estimated probabilities using an existing smoothing technique (e.g., (Dagan et al., 1992; Gale and Church, 1990)), then calculate some similarity measure using the smoothed probabilities, and then cluster words according to it. There is no guarantee, however, that the employed smoothing method is in any way consistent with the clustering method used subsequently. Our method based on MDL resolves this issue in a unified fashion. By employing models that embody the assumption that words belonging to a same class occur in the same context with equal likelihood, our method achieves the smoothing effect as a side effect of the clustering process, where the domains of smoothing coincide with the classes obtained by clustering. Thus, the coarseness or fineness of clustering also determines the degree of smoothing. All of these effects fall out naturally as a corollary of the imperative of 'best possible estimation', the original motivation behind the MDL principle.

In our simulated annealing algorithm, we could alternatively employ the Maximum Likelihood Estimator (MLE) as criterion for the best probabilistic model, instead of MDL. MLE, as its name suggests, selects a model which maximizes the likelihood of the data, that is, $\hat{P} = \arg\max_P \prod_{x \in S} P(x)$. This is equivalent to minimizing the 'data description length' as defined in Section 3, i.e. $\hat{P} = \arg\min_P \sum_{x \in S} -\log P(x)$. We can see easily that MDL generalizes MLE, in that it also takes into account the complexity of the model itself. In the presence of models with varying complexity, MLE tends to overfit the data, and output a model that is too complex and tailored to fit the specifics of the input data. If we employ MLE as criterion in our simulated annealing algorithm, it will result in selecting a very fine model with many small clusters, most of which will have probabilities estimated as zero. Thus, in contrast to employing MDL, it will not have the effect of smoothing at all.

Purely as a method of estimation as well, the superiority of MDL over MLE is supported by convincing theoretical findings (c.f. (Barron and Cover, 1991; Yamanishi, 1992)). For instance, the speed of convergence of the models selected by MDL to the true model is known to be near optimal. (The models selected by MDL converge to the true model approximately at the rate of $1/s$ where s is the number of parameters in the true model, whereas for MLE the rate is $1/t$, where t is the size of the domain, or in our context, the total number of elements of $\mathcal{N} \times \mathcal{V}$.) 'Consistency' is another desirable property of MDL, which is not shared by MLE. That is, the number of parame-

[7] The exact formulation of $L_{mod}(M)$ is subjective, and it depends on the exact coding scheme used for the description of the models.

[8] As we noted earlier, an alternative would be to employ an agglomerative algorithm.

```
Algorithm: Clustering
1. Divide the noun set $N$ into two subsets. Define a probabilistic model consisting of the partition
of nouns specified by the two subsets and the entire set of verbs.
2. do {
    2.1 Randomly select one noun, remove it from the subset it belongs to and add it to the other.
    2.2 Calculate the description length for the two models (before and after the move) as $L_1$ and
    $L_2$, respectively.
    2.3 Viewing the description length as the energy function for annealing, let $\Delta L = L_2 - L_1$.
    If $\Delta L < 0$, fix the move, otherwise ascertain the move with probability $P = \exp(-\Delta L/T)$.
} while (the description length has decreased during the past $10 \cdot |N|$ trials.)
    Here $T$ is the annealing temperature whose initial value is 1 and updated to be $0.9T$ after
    $10 \cdot |N|$ trials.
3. If one of the obtained subset is empty, then return the non-empty subset, otherwise recursively
apply Clustering on both of the two subsets.
```

Figure 1: Simulated annealing algorithm for word clustering

ters in the models selected by MDL converge to that of the true model (Rissanen, 1989). Both of these properties of MDL are empirically verified in our present context, as will be shown in the next section. In particular, we have compared the performance of employing an MDL-based simulated annealing against that of one based on MLE in hierarchical word clustering.

5 Experimental Results

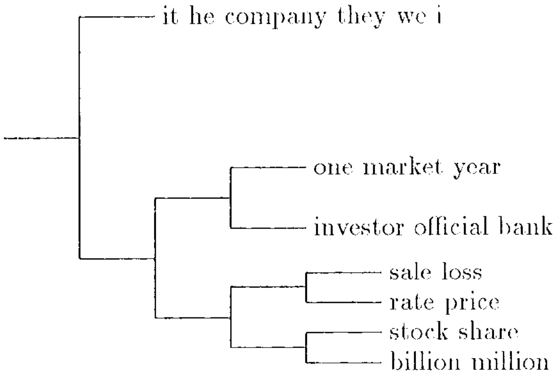

Figure 2: An example thesaurus

We describe our experimental results in this section.

5.1 Experiment 1: MDL v.s. MLE

We compared the performance of employing MDL as a criterion in our simulated annealing algorithm, against that of employing MLE by simulation experiments. We artificially constructed a *true* model of word co-occurrence, and then generated data according to its distribution. We then used the data to estimate a model (clustering words), and measured the KL distance[9] between the true model and the estimated model. (The algorithm used for MLE was the same as that shown in Figure 1, except the 'data description length' replaces the (total) description length' in Step 2.) Figure 3(a) plots the number of obtained noun clusters (leaf nodes in the obtained thesaurus tree) versus the input data size, averaged over 10 trials. (The number of noun clusters in the true model is 4.) Figure 3(b) plots the KL distance versus the data size, also averaged over the same 10 trials. The results indicate that MDL converges to the true model faster than MLE. Also, MLE tends to select a model overfitting the data, while MDL tends to select a model which is simple and yet fits the data reasonably well.

5.2 Experiment 2: Qualitative Evaluation

We extracted roughly 180,000 case frames from the bracketed WSJ (Wall Street Journal) corpus of the Penn Tree Bank (Marcus et al., 1993) as co-occurrence data. We then constructed a number of thesauri based on these data, using our method. Figure 2 shows an example thesaurus for the 20 most frequently occurred nouns in the data, constructed based on their appearances as subject and object of roughly 2000 verbs. The obtained thesaurus seems to agree with human intuition to some degree. For example, 'million' and 'billion' are classified in one noun cluster, and 'stock' and 'share' are classified together. Not all of the noun clusters, however, seem to be meaningful in the useful sense. This is probably because the data size we had was not large enough. Pragmatically speaking, however, whether the obtained thesaurus agrees with our intuition in itself is only of secondary concern, since the main purpose is to use the constructed thesaurus to help improve on a disambiguation task.

[9] The KL distance (relative entropy), which is widely used in information theory and statistics, is a measure of 'distance' between two distributions (Cover and Thomas, 1991). It is always non-negative and is zero iff the two distributions are identical.

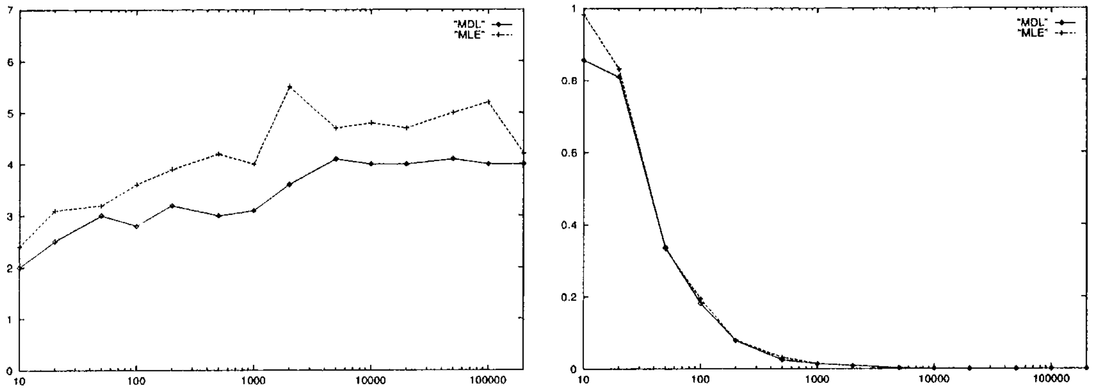

Figure 3: (a) Number of clusters versus data size and (b) KL distance versus data size

5.3 Experiment 3: Disambiguation

We also evaluated our method by using a constructed thesaurus in a pp-attachment disambiguation experiment.

We used as training data the same 180,000 case frames in Experiment 1. We also extracted as our test data 172 $(verb, noun_1, prep, noun_2)$ patterns from the data in the same corpus, which is *not* used in the training data. For the 150 words that appear in the position of $noun_2$ in the test data, we constructed a thesaurus based on the co-occurrences between heads and slot values of the frames in the training data. This is because in our disambiguation test we only need a thesaurus consisting of these 150 words. We then applied the learning method proposed in (Li and Abe, 1995) to learn case frame patterns with the constructed thesaurus as input using the same training data. That is, we used it to learn the conditional distributions $P(Class_1|verb, prep)$, $P(Class_2|noun_1, prep)$, where $Class_1$ and $Class_2$ vary over the internal nodes in a certain 'cut' in the thesaurus tree [10]. We then compare $P(noun_2|verb, prep)$ and $P(noun_2|noun_1, prep)$, which are estimated based on the case frame patterns, to determine the attachment site of $(prep, noun_2)$. More specifically, if the former is larger than the latter, we attach it to *verb*, and if the latter is larger than the former, we attach it to $noun_1$, and otherwise (including when both are 0), we conclude that we cannot make a decision. Table 1 shows the results of our pp-attachment disambiguation experiment in terms of 'coverage' and 'accuracy.' Here 'coverage' refers to the proportion (in percentage) of the test patterns on which the disambiguation method could make a decision. 'Base Line' refers to the method of always attaching $(prep, noun_2)$ to $noun_1$. 'Word-Based', 'MLE-Thesaurus', and 'MDL-Thesaurus' respectively stand for using word-based estimates, using a thesaurus constructed by employing MLE, and using a thesaurus constructed by our method. Note that the coverage of 'MDL-Thesaurus' significantly outperformed that of 'Word-Based', while basically maintaining high accuracy (though it drops somewhat), indicating that using an automatically constructed thesaurus can improve disambiguation results in terms of coverage.

Table 1: PP-attachment disambiguation results

	Coverage(%)	Accuracy(%)
Base Line	100	70.2
Word-Based	19.7	95.1
MDL-Thesaurus	33.1	93.0
MLE-Thesaurus	33.7	89.7
WordNet	49.4	88.2

We also tested the method proposed in (Li and Abe, 1995) of learning case frames patterns using an existing thesaurus. In particular, we used this method with WordNet (Miller et al., 1993) and using the same training data, and then conducted pp-attachment disambiguation experiment using the obtained case frame patterns. We show the result of this experiment as 'WordNet' in Table 1. We can see that in terms of 'coverage', 'WordNet' outperforms 'MDL-Thesaurus', but in terms of 'accuracy', 'MDL-Thesaurus' outperforms 'WordNet'. These results can be interpreted as follows. An automatically constructed thesaurus is more domain dependent and captures the domain dependent features better, and thus using it achieves high accuracy. On the other hand, since training data we had available is insufficient, its coverage is smaller than that of a hand made thesaurus. In practice, it makes sense to combine both types of thesauri. More specifically, an automatically constructed thesaurus can be used within its coverage, and outside its coverage, a hand made thesaurus can be used. Given the current state of the word clustering technique (namely, it requires data size that is usually not available, and it tends to be computationally demanding), this strategy is practical. We show the result of this combined

[10] Each 'cut' in a thesaurus tree defines a different noun partition. See (Li and Abe, 1995) for details.

Table 2: PP-attachment disambiguation results

	Coverage(%)	Accuracy(%)
MDL-Thesaurus + WordNet	54.1	87.1
MDL-Thesaurus + WordNet + LA + Default	100	85.5

method as 'MDL-Thesaurus + WordNet' in Table 2. Our experimental result shows that employing the combined method does increase the coverage of disambiguation. We also tested 'MDL-Thesaurus + WordNet + LA + Default', which stands for using the learned thesaurus and WordNet first, then the lexical association value proposed by (Hindle and Rooth, 1991), and finally the default (i.e. always attaching $prep, noun_2$ to $noun_1$). Our best disambiguation result obtained using this last combined method somewhat improves the accuracy reported in (Li and Abe, 1995) (84.3%).

6 Concluding Remarks

We have proposed a method of hierarchical clustering of words based on large corpus data. We conclude with the following remarks.

1. Our method of clustering words based on the MDL principle is theoretically sound. Our experimental results show that it is better to employ MDL than MLE as estimation criterion in hierarchical word clustering.
2. Using a thesaurus constructed by our method can improve pp-attachment disambiguation results.
3. At the current state of the art in statistical natural language processing, it is best to use a combination of an automatically constructed thesaurus and a hand made thesaurus for disambiguation purpose. The disambiguation accuracy obtained this way was 85.5%.

In the future, hopefully with larger training data size, we plan to construct larger thesauri as well as to test other clustering algorithms.

References

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

Peter F. Brown, 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.

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

Ido Dagan, Shaul Marcus, and Shaul Makovitch. 1992. Contextual word similarity and estimation from sparse data. *Proceedings of the 30th ACL*, pages 164-171.

Williams A. Gale and Kenth W. Church. 1990. Poor estimates of context are worse than none. *Proceedings of the DARPA Speech and Natural Language Workshop*, pages 283-287.

Donald Hindle and Mats Rooth. 1991. Structural ambiguity and lexical relations. *Proceedings of the 29th ACL*, pages 229-236.

Donald Hindle. 1990. Noun classification from predicate-argument structures. *Proceedings of the 28th ACL*, pages 268-275.

Hang Li 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.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*, 19(1):313-330.

George A. Miller, Richard Beckwith, Chirstiane Fellbaum, Derek Gross, and Katherine Miller. 1993. Introduction to WordNet: An on-line lexical database. *Anonymous FTP: clarity.princeton.edu*.

Fernando Pereira, Naftali Tishby, and Lillian Lee. 1993. Distributional clustering of english words. *Proceedings of the 31st ACL*, pages 183-190.

J. Ross Quinlan and Ronald L. Rivest. 1989. Inferring decision trees using the minimum description length principle. *Information and Computation*, 80:227-248.

Jorma Rissanen. 1989. *Stochastic Complexity in Statistical Inquiry*. World Scientific Publishing Co.

Takenobu Tokunaga, Makoto Iwayama, and Hozumi Tanaka. 1995. Automatic thesaurus construction based-on grammatical relations. *Proceedings of IJCAI'95*.

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