Hierarchical Bayesian Clustering for Automatic Text Classification

Makoto IWAYAMA
Advanced Research Laboratory
Hitachi Ltd
Hatoyama
Saitama 350-03, Japan
lvayamaiOharl hitachi co jp

Takenobu TOKUNAGA
Department of Computer Science
Tokyo Institute of Technology
Ookayama, Meguro
Tokyo 152, Japan
takedes titech ac jp

Abstract

Text classification, the grouping of texts into several clusters, has been used as a means of improving both the efficiency and the effective-Dess of text retrieval/categorization In this paper we propose a hierarchical clustering algorithm that constructs a Bet of clusters having the maximum Bayesian posterior probability, the probability that the given texts are classified into clusters We call the algorithm Hierarchical Bayesian Clustering (HBC) The advantages of HBC are experimentally verified from several viewpoints (1) HBC can re-construct the original clusters more accurately than do other non probabilistic algorithms (2) When a probabilistic text categorization is extended to a cluster-based one, the use of HBC offers better performance than does the use of non probabilistic algorithms

1 Introduction

Text classification., the grouping of texts into several clusters, has been used as a means of improving both the efficiency and the effectiveness of *ttzt retrieval/categorization* [Jardine and Van Rjjabergen, 1971, van Rijsbergen and Croft, 1975, Croft, 1980, Willett, 1983] For example, to retrieve texts relevant to a user's request, a simple strategy would be to search all the texts in a database by calculating a measure of the relevance of each text to the request This exhaustive search, however, would require more computation for larger databases Text classification helps to reduce the number of comparisons in an exhaustive search by clustering (grouping) similar textB into clusters in advance and comparing the request with the representative of each cluster Clustering is also assumed to improve the accuracy of retrieval, but this assumption has not been verified The retrieval/categorization model that incorporates text classification as a preliminary process is often called *cluster-based text retrieval/categorization*

In this paper we propose a probabilistic algorithm of hierarchical clustering and compare it with other clustering algorithms Almost all previous algorithms, such as single-link method and Ward's method, use the measure of distance between two objects and merge the closer ones [Cormack, 1971, Anderberg, 1973, Griffiths *et* a/, 1964, Willett, 1988], our algorithm, though, constructs a set of clusters that has the maximum Bayesian posterior probability, the probability that the given objects are classified into clusters This maximization is a general form of the well known *Maximum Likelihood* estimation, and we call the algorithm *Hierarchical Bayesian Clus tenng* (HBC)

Probabilistic models are becoming popular in the field of text retrieval/categorization owing to their solid formal grounding in probability theory [Croft, 1981, Fuhr, 1989, Kwok, 1990, Lewis, 1992] They retrieve those texts that have larger posterior probabilities of being relevant to a request When these models are extended to cluster-based text retrieval/categorization, however, the algorithm used for text clustering has still been a non probabilistic one [Croft, 1980] We think that better performance could be obtained by using exactly the same criterion in both clustering and retrieval/categorization, that is, searching for the maximum posterior probability In this paper we verify this assumption through preliminary experiments where we compare a probabilistic text categorization using HBC and the same categorization using non probabilistic clustering algorithms

The term "categorization" in this paper refers to the assignment of documents to predefined categories, whereas "classification" and "clustering" refers to only grouping of documents without identifying the meaning of groups

2 Hierarchical Bayesian Clustering

Like most agglomerative clustering algorithms [Cormack, 1971, Anderberg, 1973, Griffiths *et al*, 1984, Willett, 1988], HBC constructs a cluster hierarchy (also called *dendrogram)* from bottom to top by merging two clusters at a time At the begimng (I e , at the bottom level in a dendrogram), each datum belongB to a cluster whose only member is the datum itself For every pair of clusters, HBC calculates the probability of merging the pair and selects for the next merge the best one for which this probability is highest This merge step takes place N — 1 times for a collection of N data The last merge produces a single cluster containing the entire data set

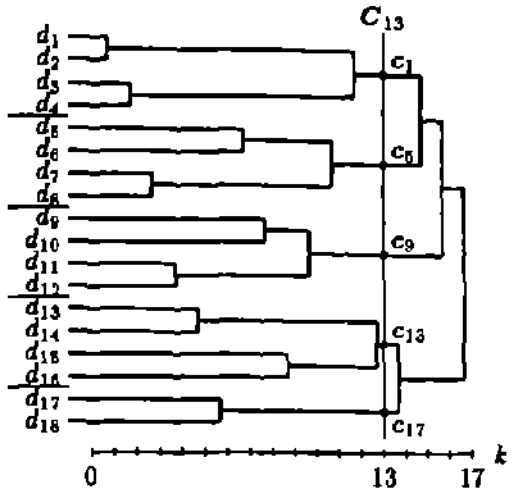

Figure 1 Example of a dendrogram

Figure 1 shows an example of a dendrogram

Formally, HBC selects the cluster pair whose merge results in the maximum value of the posterior probability $P(\boldsymbol{C}|\boldsymbol{D})$, where $\boldsymbol{D}$ is a collection of data (i e , $\boldsymbol{D} = \{d_1, d_2, \quad , d_N\}$) and $\boldsymbol{C}$ is a set of clusters (i e , $\boldsymbol{C} = \{c_1, c_2, \quad \}$) Each cluster $c_j \in \boldsymbol{C}$ is a set of data and the clusters being mutually exclusive At the initial stage, each cluster is a singleton set, $c_i = \{d_i\}$ for all i $P(\boldsymbol{C}|\boldsymbol{D})$ defines the probability that a collection of data $\boldsymbol{D}$ is classified into a set of clusters $\boldsymbol{C}$ Maximizing $P(\boldsymbol{C}|\boldsymbol{D})$ is a generalization of *Maximum Likelihood* estimation

To see the details of merge process, consider a merge step $k+1$ $(0 \leq k \leq N-1)$ By the step $k+1$, a data collection $\boldsymbol{D}$ has been partitioned into a set of clusters $\boldsymbol{C}_k$ That is each datum $d \in \boldsymbol{D}$ belongs to a cluster $c \in \boldsymbol{C}_k$ The posterior probability at this point becomes

$$\begin{aligned} P(\boldsymbol{C}_k|\boldsymbol{D}) &= \prod_{c \in \boldsymbol{C}_k} \prod_{d \in c} P(c|d) \\ &= \prod_{c \in \boldsymbol{C}_k} \prod_{d \in c} \frac{P(d|c)P(c)}{P(d)} \\ &= \frac{\prod_{c \in \boldsymbol{C}_k} P(c)^{|c|}}{P(\boldsymbol{D})} \prod_{c \in \boldsymbol{C}_k} \prod_{d \in c} P(d|c) \\ &= \frac{PC(\boldsymbol{C}_k)}{P(\boldsymbol{D})} \prod_{c \in \boldsymbol{C}_k} SC(c) \end{aligned} \tag{1}$$

Here $PC(\boldsymbol{C}_k)$ corresponds to the prior probability that N random data are classified into a set of clusters $\boldsymbol{C}_k$ This probability is defined as follows

$$PC(\boldsymbol{C}_k) = \prod_{c \in \boldsymbol{C}_k} P(c)^{|c|} \tag{2}$$

$SC(c)$ defines the probability that all the data in a cluster c are produced from the cluster and is defined as

$$SC(c) = \prod_{d \in c} P(d|c) \tag{3}$$

When the algorithm merges two clusters $c_x, c_y \in \boldsymbol{C}_k$, the set of clusters $\boldsymbol{C}_k$ is updated as follows

$$\boldsymbol{C}_{k+1} = \boldsymbol{C}_k - \{c_x, c_y\} + \{c_x \cup c_y\} \tag{4}$$

After the merge, the posterior probability is inductively updated as

$$P(\boldsymbol{C}_{k+1}|\boldsymbol{D}) = \frac{PC(\boldsymbol{C}_{k+1})}{PC(\boldsymbol{C}_k)} \frac{SC(c_x \cup c_y)}{SC(c_x)SC(c_y)} P(\boldsymbol{C}_k|\boldsymbol{D}) \tag{5}$$

Note that this updating is local and can be done efficiently since all we have to recalculate from the previous step is the probability for the merged new cluster, that is, $SC(c_x \cup c_y)$ As for the factor of $\frac{PC(\boldsymbol{C}_{k+1})}{PC(\boldsymbol{C}_k)}$, we use a well known estimate[1] the prior probability of a model (in this case, a cluster) is a decreasing function of the model size For instance, $P(c) \propto A^{-|c|}$ for some constant $A > 1$ According to this estimate,

$$PC(\boldsymbol{C}) = \prod_{c \in \boldsymbol{C}} P(c)^{|c|} \propto \prod_{c \in \boldsymbol{C}} A = A^{|\boldsymbol{C}|} \tag{6}$$

Since the number of clusters $|\boldsymbol{C}|$ decreases one by one as the merge step proceeds, $\frac{PC(\boldsymbol{C}_{k+1})}{PC(\boldsymbol{C}_k)}$ reduces to a constant value A^{-1} regardless of the merged pair This means that we can drop the facter for a maximization task HBC calculates the updated $P(\boldsymbol{C}_{k+1}|\boldsymbol{D})$ for every merge candidate and merges the one that offers the maximum $P(\boldsymbol{C}_{k+1}|\boldsymbol{D})$ The HBC algorithm is summarized in Figure 2

Lastly, we will show an example of calculating the elemental probability $P(d|c)$ that a cluster c produces its member d Depending on the data representation form and available information for data, there could be various methods for calculating this probability In this paper we use Iwayama and Tokunaga's method [Iwayama and Tokunaga, 1994]

In [Iwayama and Tokunaga, 1994], each datum d is a document, and is represented as a set of terms (usually only nouns are used for terms) Because a cluster c is a set of documents, it is also represented as a set of terms that all the documents in c have Consider here an event $T = t$, that a randomly extracted term T from a set of terms is equal to t Conditioning $P(d|c)$ on each possible event gives

$$P(d|c) = \sum_t P(d|c, T = t)P(T = t|c) \tag{7}$$

If we assume conditional independence between c and d given $T = t$, we obtain

$$P(d|c) = \sum_t P(d|T = t)P(T = t|c) \tag{8}$$

[1] We have devised a more formal estimate based on the MDL principle [Rissanen, 1989] and have used the estimate to determine the best set of clusters in dendrogram, but this topic is outside the scope of this paper

Input
$D = \{d_1, d_2, \quad , d_N\}$ a collection of N data,

Initialize
$C_0 = \{c_1, c_2, \quad , c_N\}$ a set of clusters,
$c_i = \{d_i\}$ for $1 \leq i \leq N$
calculate $SC(c_i)$ for $1 \leq i \leq N$
calculate $SC(c_i \cup c_j)$ for $1 \leq i < j \leq N$

for $k = 1$ to $N - 1$ **do**
$(c_x, c_y) = \arg\max_{c_x, c_y} \frac{SC(c_x \cup c_y)}{SC(c_x)SC(c_y)}$
$C_k = C_{k-1} - \{c_x, c_y\} + \{c_x \cup c_y\}$
calculate $SC(c_x \cup c_z)$ for all $c_z \in C_k$ where $z \neq x$

Function $SC(c)$
return $\prod_{d \in c} P(d|c)$

Figure 2 Hierarchical Bayesian Clustering

Using Bayes' theorem, we can write this as

$$P(d|c) = P(d) \sum_t \frac{P(T = t|d)P(T = t|c)}{P(T = t)} \qquad (9)$$

Since each $P(d)$ appears in every estimation of $P(C|D)$ only once, $P(d)$ can be excluded for maximization purpose Other probabilities $P(T = t|d)$, $P(T = t|c)$, and $P(T = t)$ are estimated from the given data by using the simple estimations described below

- $P(T = t|d)$ relative frequency of a term t in a document d
- $P(T = t|c)$ relative frequency of a term t in a cluster c
- $P(T = t)$ relative frequency of a term t in the entire data set

The general framework of HBC is similar to Ward's method, a well known hierarchical clustering algorithm (see [Anderberg, 1973, Griffiths *et al*, 1984, Cormack, 1971] for the algorithm) Whereas Ward's method merges two clusters whose merge causes the least increase in the sum of the distances from each datum to the centroid of its cluster, HBC maximizes the probability that all the members of a cluster actually belong (or are categorized) to the cluster We think that in application domains like text categorization, HBC would work better than Ward's method because the cluster construction strategy of HBC is more directly related to the task of such applications Before we verify this assumption in section 4, we make some general comparisons between HBC and other clustering algorithms

3 Comparison to Other Clustering Algorithms

Comparing different clustering algorithms is very difficult, but plenty of criteria have been proposed (see [Dubes and Jain, 1979] for a good survey) In this section

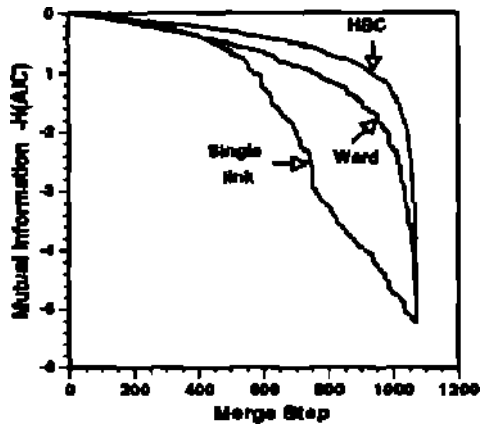

Figure 3 Mutual information at each merge step

we compare HBC with other two representative hierarchical clustering algorithms [Anderberg, 1973] (single-link method and Ward'e method) from the viewpoint that how accurately the original (i e , true in a sense) classification can be reconstructed from Scratch

The data set we used was a dictionary of contemporary words (in Japanese), called *Gendar yogo no Kiso*-tiJih(GK) [Jiyukokuminsya, 1992] It contains 18,476 word entries, each of which is classified into one of 149 categories The length of texts explaining each word varies from 13 to 1,938 characters (in kanji) and averages 287 characters To eliminate the effect of noise, we excluded smaller entries and smaller categories according to two thresholds[2]

- For an entry, the number of terms in the entry should be more than 100
- For a category, the number of entries that belong to the category should be more than 20

There remained 1,072 entries classified into 39 categories All texts were tagged by a Japanese morphological analyzer called JUMAN [Matsumoto, 1993], and each entry became a set of terms (I e , nouns) with relative frequencies

From these 1,072 entries, we constructed three cluster hierarchies using single-link method, Ward's method, and HBC, and we compared these hierarchies First we calculated the *mutual information 1(C,A)* between a set of constructed clusters *C* and the set of the original categories *A* For each entry d, a category *a € A* had been assigned by GK's editors beforehand and a cluster c € *C* is assigned by a clustering algorithm So there are two classifications one by human experts and tbe other by a clustering algorithm Since mutual information *I(C,A)* defines the amount of information that is commonly contained in *C* and *A,* the larger the value *I(C,A)* has, the more closely the set of clusters *C* approximates the original set of categories *A* Formally,

[3]In preliminary experiments, variations of tbe two thresh olds had Utile affect on the overall results We set the threshold values because the memory available to run the program was limited

		should be classified to the same cluster	
		yes	no
be actually classified to the same cluster	yes	a	b
	no	c	d

Table 1 Number of entry pairs for evaluating accuracy

$I(C, A)$ is defined as

$$I(C, A) = H(A) - H(A|C), \tag{10}$$

where $H(A)$ is the entropy of A and $H(A|C)$ is the conditional entropy of A given C Since $H(A)$ is independent of C, to compare different C's we have only to calculate $-H(A|C)$

$$-H(A|C) = -\sum_{c \in C} P(c) H(A|c)$$
$$= \sum_{c \in C} P(c) \sum_{a \in A} P(a|c) \log P(a|c) \tag{11}$$

The value of $-H(A|C)$ at every merge step of each clustering algorithm is plotted in Figure 3, where we can see that the value is always higher for HBC than for the other two algorithms This means that HBC could reconstruct the original categories more precisely than the other two We see that for all the algorithms $-H(A|C)$ decreases monotonically as the merging proceeds (i e , as the number of clusters decreases) This is because a larger number of clusters can express the original categories more precisely owing to the larger number of model parameters Since in the most extreme case, when no merge has occurred, every cluster needs to express only one entry for each, this set of clusters can directly encode the original categories without noise Note, however that a larger set of small clusters generally has less predictability for unseen new data because of overspecification to the given data

We also calculated the accuracy of a set of constructed clusters If two entries d_i and d_j have the same category attached beforehand and an algorithm assigns both to the same cluster, we say that clustering is positively correct with respect to the pair If d_i and d_j have different categories and an algorithm assigns two of them different clusters, we say that clustering is negatively correct Counting the correctness for every pair of entries yields the positive accuracy (PA) (also called sensitivity) and the negative accuracy (NA) (also called specificity) Referring to Table 1, we define PA and NA as follows

$$PA = \frac{a}{a + c} \tag{12}$$

$$NA = \frac{d}{b + d} \tag{13}$$

The averaged accuracy ($\frac{PA+NA}{2}$) at each merge step of each clustering algorithm is shown in Figure 4 Unlike $-H(A|C)$ in Figure 3, the averaged accuracy reaches a maximum value at some step between the beginning and the end At the beginning, since every entry is distributed to a unique cluster for each, NA becomes 1 but

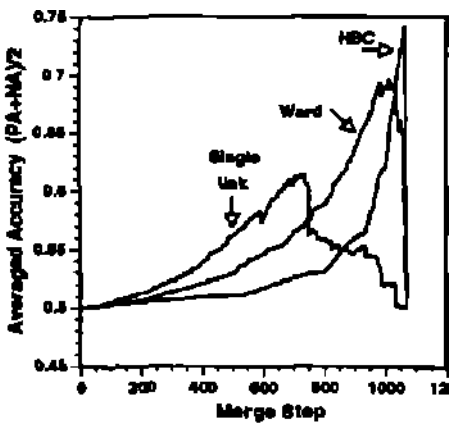

Figure 4 Average accuracy at each merge step

PA becomes 0 As the merging steps proceed, NA decreases and PA increases monotonically At the end, since all the entries are contained in the single cluster, PA becomes 1 but NA becomes 0 Usually, the best point for both NA and PA lies between the beginning and the end From the figure, we see that HBC offers the best upper bound of the averaged accuracy In addition, since the optimal set of clusters obtained by HBC is the smallest, HBC could construct a set of clusters that has the most generality (i e, the highest predictive power) for unseen data — from the standpoint of the MDL principle [Rissanen, 1989] stating that smaller models are better

4 Evaluation in Text Categorization

Text categonzation is the classification of documents with respect to a set of predefined categories We use text clustering to improve the efficiency of a text categorization based on Memory Based Reasoning (MBR) (a k nearest neighbor search) [Masand *et* a/, 1992] MBR solves a new task by looking up examples of tasks similar to the new task [Stanfill and Waltz, 1986] For example, to attach categories to a new document, MBR searches the nearest it documents in a large set of already categorized documents (called training data) and uses the categories attached to the searched k documents to determine categories of the new document Although this strategy offers promising performance [Masand *et al*, 1992], MBR requires a large amount of computational power for calculating a measure of the similarity between a new document and every example and for sorting the similarities For practical applications, parallel computing is usually necessary

We use a clustering algorithm for preliminary processing to reduce the number of comparisons by partitioning a large amount of training data into several clusters [Jardine and Van Rijsbergen, 1971, van Rysbergen and Croft, 1975, Croft, 1980, Willett, 1983] The original MBR searches the k nearest documents, but the cluster-based MBR searches the k nearest clusters of documents The documents in the searched k clusters are used to attach categories to a new document This pre-clustering

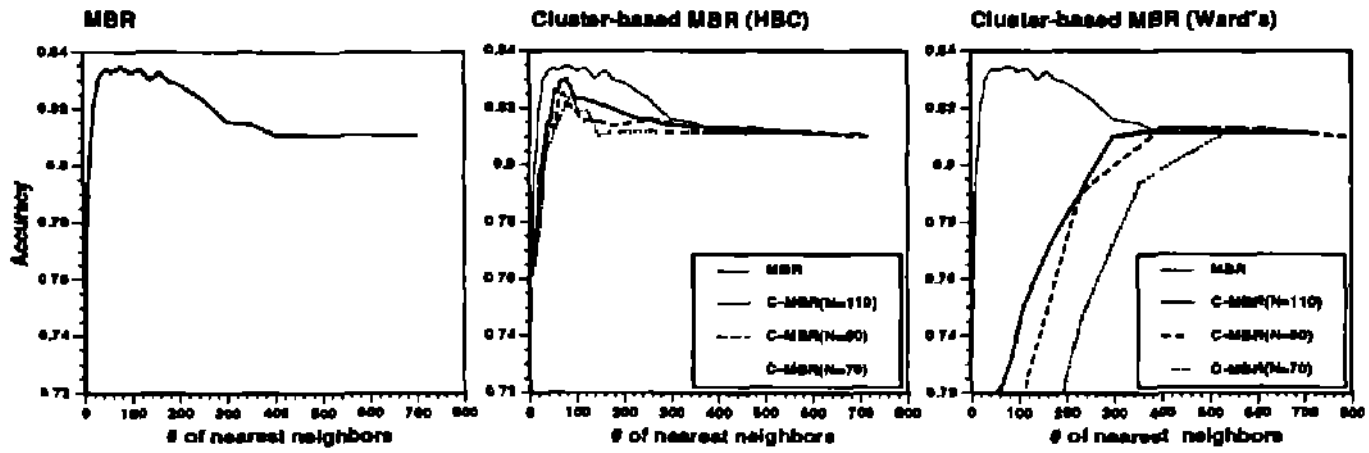

Figure 5 Performance of MBR and cluster-based MBR's

procedure could eliminate a great deal of computation if the number of clusters IB relatively small In addition, if the accuracy of the cluster-based MBR is comparable to that of the original MBR, the cluster-based version would be an efficient approximation of the original MBR This section describes our experimental investigate of how well HBC and Ward's method can help to approximate the performance of the original MBR

In experiments we used the GK data set described in section 3 To divide the data set into two Bets, one for examples with categories (training data) and the other for evaluation (test data), we used 4-fold cross-validation The 1,072 above-threshold entries were randomly divided into four Bets, one used for test data and the remaining three used for training data For each entry in the test data, an text categorization model assigned a category using the training data If the category assigned by the model was the same as the category assigned by GK's editors, we said that the categorization was correct We used the accuracy of categorization to measure the performance of a model Since there were four variations of test data depending on which set of the four we select, we did four series of experiments and averaged the results

To see the performance of the original MBR, we calculated the probability $P(\{d_{train}\}|d_{test})$ for every pair consisting of a test entry d_{1ei}, and a training entry $d|_{rQin}$ by using a method similar to that used in deducing the Eq (9) (Bee [iwayama and Tokunaga, 1994] for more details) To assign a category to each test entry d_{tettl} we searched the *k* nearest entries besed on the calculated probabilities Each d_{train} in the *k* nearest entries voted on the originally assigned category when the weight of the voting was log $P(\{d_{train}\}|d_{test})$ The category with the most votes was assigned to the test entry d_{test} Figure 5 shows the result of the accuracy obtained when varying *k* from 1 to the maximum value (I e , the number of the given training data) We can see that more correct categories are attached as *k,* the number of nearest neighbors, increases but also that there is more noise after .£ reaches some value

For cluster-based MBR we first constructed a cluster hierarchy from the given training data by using each of the two clustering algorithms, HBC and Ward's method From a constructed hierarchy we selected a set of clusters $C = \{c_1, c_2, \quad , c_N\}$ at some merge step For each pair consisting of a test entry $d_{(tlt}$ and a cluster c, € *C* we calculated $P(c_i|d_{test})$ by using the same method as in the original MBR and extracted the nearest *k* clusters from C All the members in the nearest *k* clusters become the approximated nearest entries of the test entry Using the same voting strategy as in the original MBR, a category was assigned to each test entry Figure 5 shows the results when the number of selected clusters *(N)* was set to 70, 90, and 110 The original MBR curve is also plotted for reference We can see that the accuracy of the cluster-based MBR with HBC approaches that of the original MBR as the number of constructed clusters increases Note that the original MBR corresponds to the extreme case where *N* is the maximum value (i e , the number of training data) We can also see the advantage of HBC over Ward's method Cluster-based MBR with HBC approximates the original MBR sufficiently well whereas the cluster-based MBR with Ward's method provides poor performance, especially when the number of nearest neighbors is small

5 Discussion

We have proposed a new hierarchical clustering algorithm that is based on maximizing the Bayesian posterior probability The advantages of this algorithm have been experimentally verified from several viewpoints, including its performance within an actual application of text categorization In summary,

- From the standpoints of the mutual information and the classification accuracy, HBC can re-construct the original clusters more accurately than do the single-link method and Ward's method
- When a probabilistic text categorization is extended to a cluster-based one, the use of HBC offeres better performance than does the use of non probabilistic algorithm like Ward's method

Since the HBC algorithm itself is general and not restricted to text clustering, it could be easily applied to various applications, such as automatic thesaurus construction We have in fact used the algorithm to construct noun hierarchies (a kind of thesaurus) using grammatical relations and have verified their contribution to disambiguation tasks [Tokunaga et *al*, 1995]

One issue we have not discussed in this paper is Lhe selection of an appropriate set of clusters from a constructed dendrogram For a collection of *N* data there are *N* possibilities of the set, each of which corresponds to a merge step Generally, a larger set of clusters (at earlier merge step) can express the given data more precisely but has poorer predictability for unseen data On the other hand, a smaller set (at later merge step) has the opposite characteristics Selecting the optimal set of clusters is a crucial issue for applications In an application of cluster-based MBR (described in section 4), the amount of computation is also affected by the size of the set larger sets needs more comparisons than do smaller sets For this general issue of model selection, several statistical measures, such as AIC [Akaike, 1974] and MDL [Rissanen, 19891, have been proposed Since our algorithm is based on probability theory, it would be easy to introduce these statistical measures into our algorithm

Acknowledgments

The authors are grateful to Hiroshi Motodafor beneficial discussions

References

[Akaike, 1974] H Akaike A new look at the statistical model identification *IEEE Trans on Automatic Control,* 19(6) 716-723, 1974

[Anderberg, 1973] M R Anderberg *Cluster Analysis for Applications* Academic Press, 1973

[Cormack, 1971} R M Cormack A review of classification *Journal of ike Royal Statistical Society,* 134 321-367, 1971

[Croft, 1980] W B Croft A model of cluster searching based on classification *Information Systems,* 5 189—195, I960

[Croft, 1981] W B Croft Document representation in probabilistic models of information retrieval *Journal of the American Society for Information Science,* 32(6) 451-157, 1981

[Dubes and Jain, 1979] R Dubes and A K Jam Validity studies in clustering methodologies *Pattern Recognition,* 11 235-254, 1979

[Fuhr, 1989] N Fuhr Models for retrieval with probabilistic indexing *Information Processing & Retrieval* 25(1)55-72, 1989

[Griffiths *ct al,* 1984] A Griffiths, L A Robinson, and P Willett Hierarchic agglomerative clustering methods for automatic document classification *Journal of Documentation,* 40(3) 175-205, 1984

[iwayama and Tokunaga, 1994] M Iwayama and T Tokunaga A probabilistic model for text categorization Based on a single random variable with multiple values In *Proceedings of 4th Conference on Applied Natural Language Processing,* pages 162-167, 1994

[Jardine and Van Rijsbergen, 1971] N J ax dine and C J Van Rijsbergen The use of hierarchic clustering in information retrieval *Information Storage and Retrieval,* 7 217-240, 1971

[jiyukokuminsya, 1992] Jiyukokuminsya Gendai yogo no kiBotisiki Jiyukokuminsya, 1992 (in Japanese)

[Kwok, 1990] K L Kwok Experiments with a component theory of probabilistic information retrieval based on single terms as document components *ACM Transactions on Information Systems,* 8(4) 363-386, 1990

[Lewis, 1992] D D Lewis An evaluation of phrasal and clustered representation on a text categorization task In *Proceedings of the Annual International A CM SIGIR Conference on Research and Development* ir. *Information Retrieval,* pages 37-50, 1992

[Masand *et al,* 1992] B Mas and, G Lin off, and D Waltz Classifying news stones using memory based reasoning In *Proceedings of the Annual International ACM SI GIR Conference on Research and Development* in *Information Retrieval,* pages 59-65, 1992

[Matsumoto, 1993] Y Matsumoto *JUMAN Users Manual* Kyoto University and Nara Institute of Science and Technology, 1993

[Rissanen, 1989] J Rissanen *Stochastic Complexity* in *Statistical Inquiry* World Scientific Publishing, 1989

[Stanfill and Waltz, 1986] C Stanfill and D Waltz Toward memory-based reasoning *Communications of the ACM,* 29(12) 1213-1228, 1986

[Tokunaga *et al,* 1995] T Tokunaga, M Iwayama, and H Tanaka Automatic thesaurus construction based on grammatical relations In *Proceedings of the Inter national Joint Conference on Artificial Intelligence,* 1995

[van Rijrbergen and Croft, 1975] C J van Rijsbergen and W B Crofl Document clustering An evaluation of some experiments with the granfield 1400 collection *Information Processing & Management,* 11 171-182, 1975

[Willett, 1983] P Willett Similarity coefficients and weighting functions for automatic document classification an empirical comparison *International Classification,* 10(3) 138-142, 1983

[Willett, 1988] P Willett Recent trends in hierarchic document clustering A critical review *Information Processing & Management,* 24(5) 577-597, 1988