Automatic Text Categorization for Information Filtering

Ho Chao Yang

Department of Systems Engineering

& Engineering Management

The Chinese University of Hong Kong

Submitted in partial fulfillment of requirements for

the degree of Master of Philosophy

June 1998

論文摘要

現今，越來越多的全文字數據儲存在文字資料庫裡。在面對一個大型的文字資料庫時，自動文件歸類和信息過濾成爲了兩個主要的發展題目。在這篇論文裡，我們針對文件分類和信息過濾的新方法進行了研究和實驗。

首先，我們調查了現行以 Rule-based 和 Instance-based 爲主的自動文件分類方法。我們建議用一種新的技術，稱爲 IBRI。它把 rule-based 和 instance-based 的方法的好處結合起來。我們編寫了一個以 IBRI 爲核心的文字分類系統，並且以一個大型的文字資料庫爲對象進行了一連串的實驗，稱爲 Reuters-21578 文件庫。結果顯示我們的新方法優勝於其他以 rule-based 和 instance-based 爲主的方法。

我們進一步研究了幾種以 similarity-based 爲主的自動文件分方法。它們包括 *k*-NN algorithm 和 linear classifiers。我們建議用另一種新技術，稱爲 GIS。這種以 GIS 爲主的技術是特別針對文件分類加以研製出來的，它平衡了上述兩者的優點和缺點。我們分別編寫了以 GIS，ExpNet 和 linear classifiers 爲主的文字分類系統，並以兩個大型的文字資料庫進行了一連串

的實驗。這兩個資料庫包括 OHSUMED 資料庫和 Reuters-21578 資料庫。所有的結論顯示我們 的 GIS 方法優勝於最新的 *k*-NN 和 linear classifiers 方法。我們將 GIS 的方法加以改良，用來解決文字過濾的問題。並進行了廣泛的信息過濾實驗。結果亦顯示我們的新方法在信息過濾方面優勝於其他最新的方法。

Abstract

The volume of full-text data stored in text databases is increasing rapidly. As the size of a text database increases, retrieving a piece of useful information from the database takes considerable amount of time and effort. Automatic document categorization and information filtering are two major tasks dealing with a large text document corpus.

In this thesis, we conduct research on new techniques for document classification schemes learning and investigate their application to text categorization and text filtering problems. Several existing machine learning approaches in automatic document classification including linear classifiers, k-NN, RIPPER, SWAP-1 and instance-based methods are studied. Two new techniques for automatic document classification are proposed.

We first investigate existing rule-based and instance-based approaches. After identifying their shortcomings, we propose a new technique known as the IBRI algorithm which attempts to incorporate the advantages of the instance-based technique into a rule-based approach. We have implemented our approach and extensive experiments have been conducted on a large-scale, real-

world document corpus, namely the Reuters-21578 collection. The results show that our new approach outperforms rule-based and instance-based approaches.

Based on the idea of IBRI, we further investigate several recent approaches for document classification under the framework of similarity-based learning. They include two families of techniques, namely the k-nearest neighbor (k-NN) algorithm and linear classifiers. After identifying the weakness and strength of each technique, we propose another new technique known as the generalized instance set (GIS) algorithm by unifying the strengths of k-NN and linear classifiers and adapting to characteristics of document classification problems. We also explore some variants of our GIS approach. We have implemented our GIS algorithm, the ExpNet algorithm (a kind of recent k-NN algorithm), and some linear classifiers. Extensive experiments have been conducted on two common benchmark document corpora, namely the OHSUMED collection and the Reuters-21578 collection. The results show that our new approach outperforms the latest k-NN approach and linear classifiers in our experiments.

Our GIS approach is further refined to solve the text filtering problem. We compare the filtering performance of GIS, linear classifiers, and k-NN. Extensive information filtering experiments have been conducted on a benchmark document filtering corpus, namely the Foreign Broadcast Information Service (FBIS) document corpus. The results also show that our new approach outperforms the latest k-NN approach and linear classifiers in information filtering experiments.

Acknowledgment

I would like to express my deepest gratitude to my research advisor, Prof. Lam Wai. His constant encouragement and advises contributed a great deal in this research work.

I would also like to thank Prof. Cheng Chun Hung and Prof. Low Boon Toh for their valuable opinions on improving the quality of my work.

I am also grateful to all the friends I met in the Department of Systems Engineering and Engineering Management, CUHK. Their encouragement, companions and help made my life in CUHK delightful.

Calvin Chao-Yang, Ho.

June 1998.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this chapter, we first give the problem definition of automatic document categorization and information filtering tasks. Then we summarize the major contributions achieved by our research. Finally, the organization of the thesis is given.

1.1 Automatic Document Categorization

The volume of full-text data stored in text databases is increasing rapidly. The text data or documents include technical articles, memos, manuals, electronic mail, books, newspapers, magazines and journals. Text documents differ from the data stored in traditional database management systems. They are far less structured and less organized compared with traditional databases which have a more structured design. As the size of a text document collection increases,

retrieving a piece of information from the collection takes considerable amount of time and effort.

One way to organize a large document collection is to conduct *document classification*. Typically, there is a set of category labels which are known in advance. The aim of document classification is to assign a number of appropriate categories to each text document. Traditionally this task is performed manually by domain experts. Each incoming document has to be analyzed by the expert based on the content of the document. Obviously a large amount of human resources are required to carry out such classification task. For instance, the OHSUMED document collection composed of medical journal articles needs a great deal of manual work to classify each document into MeSH (Medical Subject Headings) categories [26]. Examples of MeSH categories include "Aortic Valve Prolapse", "Cardiac Tamponade" and so on. Clearly, it will be very helpful if we can automate this classification process. The goal of *automatic document categorization* is to learn a classification scheme from training examples of previously classified documents. The learned scheme can then be used to classify future text documents automatically. Figure 1.1 depicts the major tasks involved in an automatic document categorization problem. The purpose of the Document Pre-processing Task is to convert a document into an internal representation so that it can be processed. The purpose of the Classification Learning Task is to learn a classification scheme from the training documents. The purpose of the On-line Classification Task is to assign categories to new documents based on the learned classification scheme.

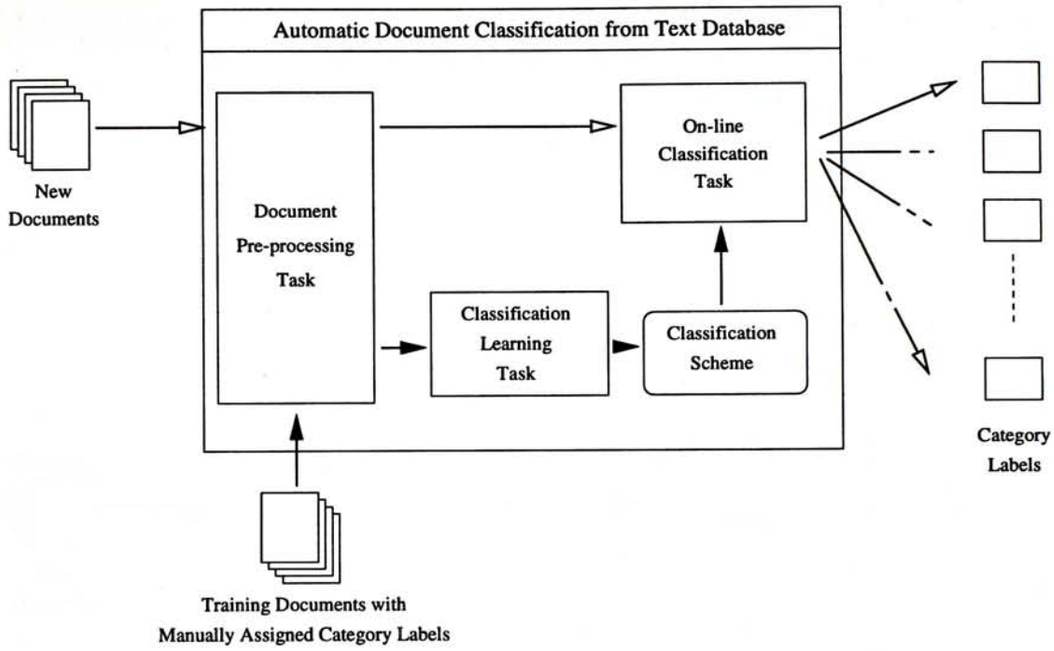

Figure 1.1: The major tasks in an automatic document categorization system

1.2 Information Filtering

The explosive growth of information makes it difficult for a user to search or keep up with the desired information. Information filters are becoming important for users to sift out relevant information. A text *information filtering* (IF) system helps a user to remove unwanted data from incoming stream of documents based on the document content and the information need. IF mainly deals with a relatively stable and long-term information need. An example of an IF service is the mailing lists on the Internet [17]. Hundreds of mailing lists covering a wide variety of topics exist. A user subscribes to lists that interest him and receives messages on those topics via email. However,

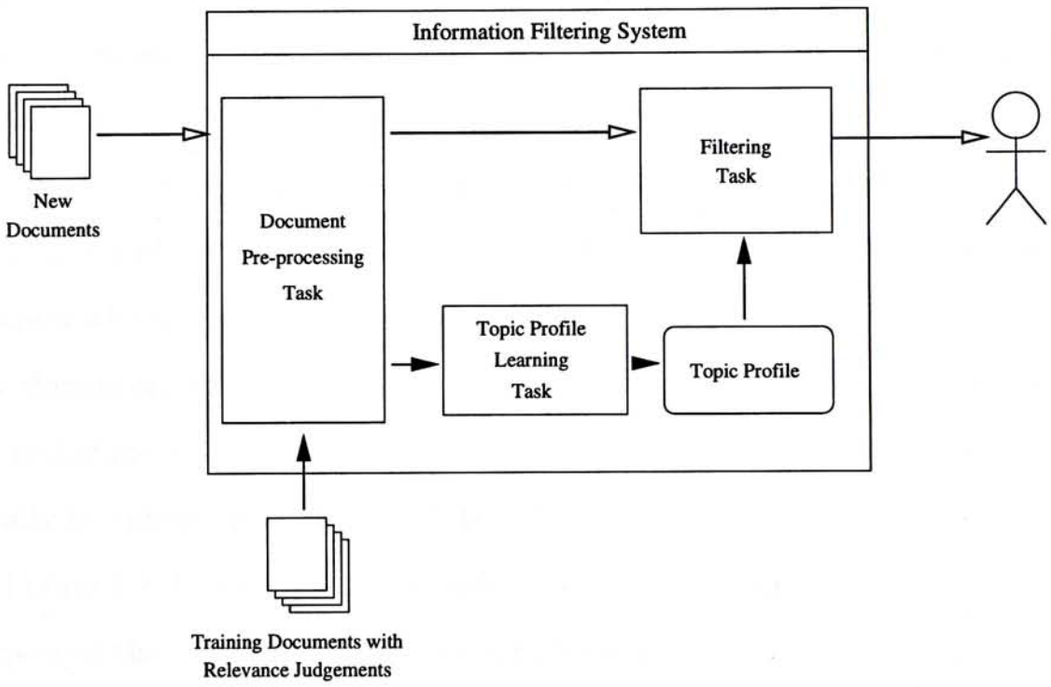

Figure 1.2: The major tasks in an information filtering system

the filtering task is done manually. Many text document collections contain relevance judgments specifying the set of documents relevant or not relevant to certain topics. For example, the documents corpus, Foreign Broadcast Information Service (FBIS), contains translated text documents or transcripts from various non-American broadcast and print publications [11]. This corpus comes with a list of topics specifying various information needs. An example of a topic is like: "A relevant document should describe non-commercial satellite launches". Associated with each topic, there is a set of sample documents which have been judged as relevant or not relevant to the topic. The aim of information filtering is to construct for each topic a topic profile or a filtering function which is able to make a binary decision to either accept or reject each new document as it arrives [10]. The document classification scheme learning technique employed in automatic document categorization problem can usually be refined to solve this IF problem.

Figure 1.2 depicts the major tasks of an information filtering system. The purpose of the Document Pre-processing Task is the same as that in automatic document categorization problem. The purpose of the Topic Profile Learning Task is to construct the Topic Profile from the training documents with relevance judgments. The purpose of the Filtering Task is to decide whether an incoming document should be presented to the user for a particular topic based on the learned Topic Profile.

1.3 Contributions

We conduct research on new techniques for document classification schemes learning and investigate their application to text categorization and text filtering problems. Several existing machine learning approaches in automatic document classification, including linear classifiers, k-NN, RIPPER, SWAP-1 and instance-based methods are studied. Two new techniques for automatic document classification are proposed, namely the IBRI and GIS. The GIS algorithm is further refined to solve the text filtering problem. The major contributions are summarized as follows:

- We propose a new technique known as the IBRI algorithm by incorporating the advantages of the instance-based technique into a rule-based approach. Extensive experiments have been conducted on a large-scale, real-world document corpus, namely the Reuters-21578 collection. The results show that our new approach outperforms rule-based and instance-based approaches.

- Based on the idea of IBRI, we propose another new technique known as the generalized instance set (GIS) algorithm by unifying the strengths of k-NN and linear classifiers and adapting to characteristics of document classification problems. Extensive experiments have been conducted on two common benchmark document corpora, namely the OHSUMED collection and the Reuters-21578 collection. The results show that our new approach outperforms the latest k-NN approach and linear classifiers in

our experiments.

- Our GIS approach is further refined to solve the text filtering problem. We compare the filtering performance of GIS, linear classifiers, and k-NN. Extensive information filtering experiments have been conducted on a benchmark document filtering corpus, namely the Foreign Broadcast Information Service (FBIS) document collection. The results also show that our new approach outperforms the latest k-NN approach and linear classifiers in information filtering experiments.

1.4 Organization of the Thesis

The organization of the thesis is as follows. In Chapter 2, we present a survey on several automatic document categorization and information filtering approaches. In particular, the rule-based and similarity-based machine learning approaches for automatic document categorization and some existing text filtering systems are described. Chapter 3 discusses the background of preprocessing text documents and the classification scheme learning strategy. In Chapter 4, a new approach in document classification scheme learning called IBRI is presented. Chapter 5 shows the experimental results of IBRI on automatic document categorization. In Chapter 6, we investigate several recent approaches for document classification under the framework of similarity-based learning. A new approach called GIS is presented. Chapter 7 shows the experimental results of our GIS approach for automatic document categorization.

Chapter 8 discusses a new information filtering technique based on GIS. Chapter 9 shows the experimental results of this GIS-based information filtering. Chapter 10 gives the conclusions and future work.

Chapter 2

Related Work

In this chapter, we present a survey on automatic document categorization and information filtering approaches.

2.1 Existing Automatic Document Categorization Approaches

We first describe some existing rule-based approaches in automatic document categorization. Typically, rule induction methods attempt to find a compact covering rule set that completely partitions the examples into their correct classes [3, 4, 27, 29]. Two recently developed approaches, namely SWAP-1 and RIPPER are discussed.

Besides, two widely used similarity-based approaches for automatic document classification, namely linear classifiers and k-NN, are discussed [23,

41, 25]. In similarity-based learning, each document is mapped to an internal representation. A metric measuring the similarity of two documents is then designed. This similarity metric is used during the training phase as well as in the online classification.

2.1.1 Rule-Based Approach

SWAP-1

Step	Predictive value(%)	Rule
1	25	c6
2	30	c1
3	41	c1 & c3
4	46	c6 & c3
5	68	c6 & c3 & c5
6	90	c6 & c3 & c5 & c2
7	100	c8 & c3 & c5 & c2

Figure 2.1: An example of swapping rule components in SWAP-1.

SWAP-1 is a rule-based learning approach for automatic document categorization [39]. The covering rule set is induced by using a decision tree. The algorithm of SWAP-1 is shown in Table 2.1. SWAP-1 constructs the rule set by repeatedly searching the single best rule and adding to the rule set. Everytime when a single best rule is found, the documents covered by it are removed. This process is repeated until no document remains. The single best rule is found by constantly examining the current candidate rule to see whether it can improve the rule, in terms of predictive accuracy, before expanding it. SWAP-1 makes the single best replacement by considering all possible swaps

or deletion of rule components. If no improvement is found, it adds the single best component to the rule. Figure 2.1 shows an example of swapping rule components. Initially, a feature value comparison c6 is assigned to the rule randomly. Then it is swapped out in favor of the single best feature value comparison c1 in Step 2. Then in Step 3, c3 is the single best feature value comparison that can be added to the rule. However, in Step 4, c6 is swapped in again. It can be seen that the components being swapped out previously can be swapped in again if it can improve the predictive accuracy of the current rule. The swapping and adding component terminate when 100-percent predictive value is reached is Step 7. The predictive value is evaluated as the percentage of correct decisions. After the covering rule set has been found, a refinement step is needed to adjust the rule set to the right complexity fit, either by pruning or by applying a statistical test.

RIPPER

RIPPER is one recent rule-based learning algorithm which has been applied to document classification [5, 6]. Figure 2.2 is an example of a learned rule set for the category "sports", where "sport", "exercise", "outdoor", "homework", "exam", "play", "rule", "food" and "business" are stemmed words discovered from the rule learning process.

Initially, RIPPER sets the rule set to empty. Like a standard separate-and-conquer algorithm, it builds a rule set incrementally. Given a category, documents belong to this category are called positive documents, and the remaining ones are negative documents. When a rule is found, all documents covered by the rule are discarded including positive documents and negative documents. The rule is then added to the rule set. The remaining documents

Input: S a set of training documents
Initialize R_1 := empty set, $k = 1$, and $C_1 := S$

repeat
  create a rule B with a randomly chosen attribute as its left-hand side
  while(B is not 100-percent predictive) do
    make the single best swap for any component of B,
      including deleting the component, using documents in C_k
    if no swap is found, add the single best component to B
  endwhile
  P_k := rule B that is now 100-percent predictive
  E_k := documents in C that satisfy the single-best-rule P_k
  $R_{k+1} := R_k \cup \{P_k\}$
  $C_{k+1} := C_k - \{E_k\}$
  $k := k + 1$
until (C_k is empty)

find rule r in R_k that can be deleted without affecting performance on documents in S
while(r can be found)
  $R_{k+1} := R_k - \{r\}$
  $k := k + 1$
endwhile
output R_k and halt

Table 2.1: The SWAP-1 procedure.

are used to build other rules in the next iteration. The process is repeated until all positive documents are covered by the rule set. To build a rule, the training data set are split into a "growing set" and a "pruning set". A rule begins with an empty conjunction of conditions. Conditions are repeatedly added to the antecedent of the rule until the rule covers no negative documents from the "growing set". After a rule is found, it is simplified by greedily deleting conditions from the antecedent so as to improve the rule's performance on the "pruning set". Different *ad hoc* heuristic measures are used to guide the searching of new conditions and the simplifications.

In rule-based learning algorithms, classification of a new document is performed by matching each rule against it and selecting those it satisfies. If there is only one such rule, the assignment of the category can be determined appropriately. If there are none, the "default rule" is used. The default rule is constructed by considering the proportion of positive and negative documents in the entire training document collection. If more than one rule cover the document, two strategies are possible. One is to order the rules into a "decision list" and select only the first rule that fires. The other is to let different rules vote and select the decision receiving the most votes.

2.1.2 Similarity-Based Approach

k-Nearest Neighbor (k-NN)

K-nearest neighbor (k-NN) algorithms are one kind of similarity-based learning. In these algorithms, each document is mapped to an internal rep-

Assign category "sports" IF

(the document contains "sport") OR

(the document contains "exercise" and "outdoor") OR

(the document contains "exercise" but not "homework" and "exam") OR

(the document contains "play" and "rule") OR

⋮

Do not assign category "sports" IF

(the document contains "food" and "cook") OR

(the document contains "business" and ...) OR

⋮

Figure 2.2: An example of a learned rule set of RIPPER

resentation. A metric measuring the similarity of two documents is then designed. This similarity metric is used during the training phase as well as in the online classification. They have been applied to the automatic document categorization problem such as ExpNet [41]. In k-NN, each training document D_j as well as the request document X, which is the document being classified, are represented by vectors. Suppose D_j is represented by $(a_{ij}, ..., a_{nj})$ and X is by $(x_1, ..., x_n)$. To conduct classification, the similarity $\Delta(X, D_j)$ between each D_j and X is calculated. One common similarity function, namely the cosine similarity, is shown as follow:

$$\Delta(X, D_j) = \frac{\sum_{i=1}^{n} x_i a_{ij}}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} a_{ij}^2}}$$

The training documents are sorted by the similarity metric in descending order.

Then the k top-ranking documents are selected. The document is assigned to categories by considering the similarity metric of these k selected documents and their category association. For instance, the desired categories are those appear in the k top-ranking documents. A more advanced technique, such as the one used in ExpNet, is to calculate the score of each category by the k top-ranking documents. Specifically, in ExpNet, the document is assigned to categories with score greater than a certain threshold value. Figure 2.3 depicts an example of a request document X and its five nearest documents. The similarity value of X with the nearest document is 0.6. Suppose there are 6 pre-defined categories. The category labels of the nearest document are 1,3 and 4. Let k be 5 and the threshold value is 0.9. To decide the categories of the request document X. We first calculate the scores of X to all categories. To calculate the score of X to a category, we sum the similarity values of X with the documents within these 5 nearest documents and the degree of association of categories within these documents. For instance:

The score of X for category 1 is $0.60 + 0.31 = 0.91$

The score of X for category 2 is $0.53 + 0.31 = 0.84$

The score of X for category 3 is $0.60 + 0.40 = 1.00$

The score of X for category 4 is $0.60 + 0.40 = 1.00$

The score of X for category 5 is 0.00

The score of X for category 6 is 0.40

Only scores of categories 1,3,4 are greater than 0.9. Therefore, we assign categories 1,3,4 to the document X.

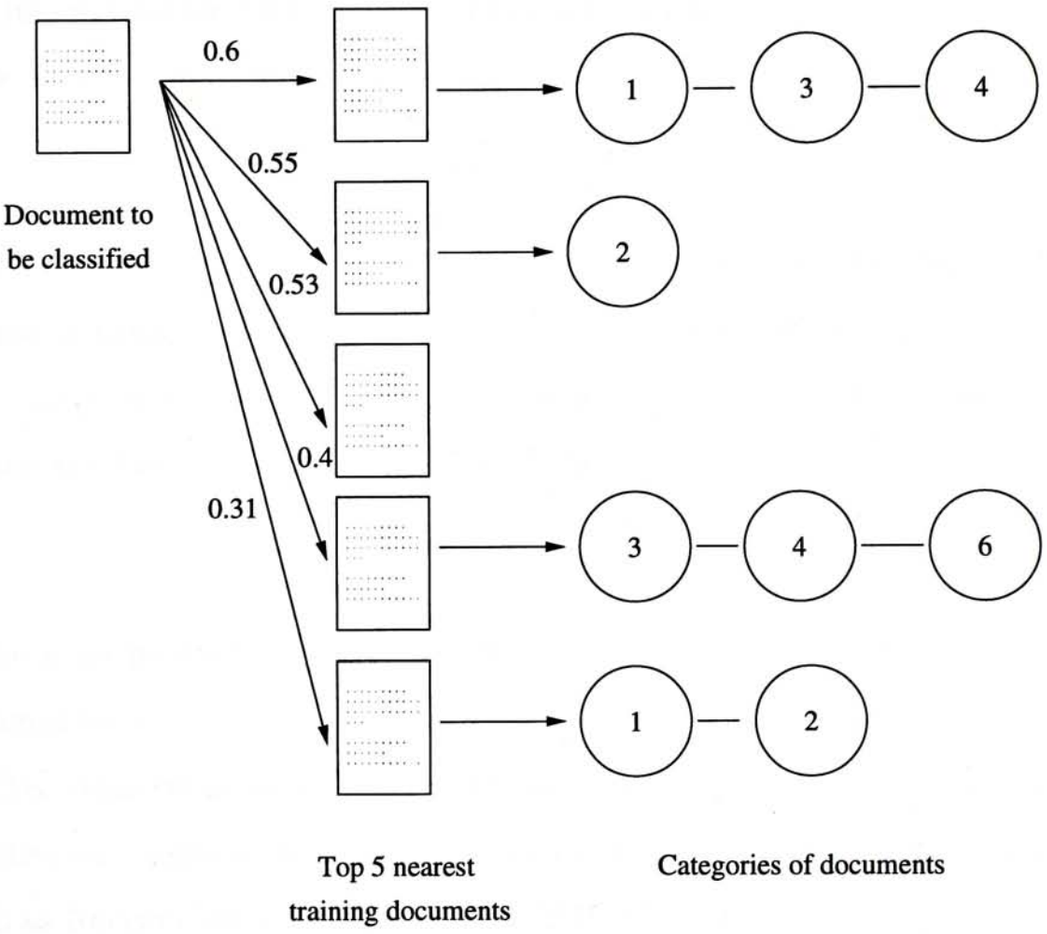

Figure 2.3: Example of the 5 nearest documents of a request document in the ExpNet algorithm.

Linear Classifiers

Linear classifiers are a family of document classification learning algorithms recently explored by Lewis [22]. We briefly describe this algorithm and point out its relationship with similarity-based algorithms.

For every category, there is a feature weight vector

$$W = (w_1, \ldots, w_n)$$

and each element w_i corresponds to the i-th feature. To determine whether or not a category is assigned to the request document X represented by $(x_1, ..., x_n)$, it computes the inner product δ between the document vector X and the feature weight vector W as follows:

$$\delta = \sum_{i=1}^{n} x_i w_i$$

If the inner product is greater than a certain threshold value, the category is assigned to X.

The elements in vector W are learned from all training examples including positive and negative documents. There are several weight learning techniques such as Rocchio [30] and Widrow-Hoff (WH) algorithm [40] .

The Rocchio algorithm is a batch algorithm. It produces a weight vector W according to the following formula:

$$W = \frac{\sum_{D \in D^+} D}{|D^+|} - \eta \frac{\sum_{D \in D^-} D}{|D^-|}$$

where η is the parameter that adjusts the relative impact of positive and negative documents. The above summation is taken as the vector addition.

$|S|$ denotes the cardinality of the set S. D^+ is the collection of positive training documents while D^- is the collection of negative training documents.

The weight vector updating function based on the WH algorithm processes each document in the training document collection one by one in each iteration. Let W_i denote the intermediate value of the weight vector at the i-th iteration. Initially, the elements in W_0 are set to all zeros. At each iteration, W_{i+1} is computed from W_i and the current document D_i.

$$W_{i+1} = W_i - 2\eta(W_i \cdot D_i - L_i)D_i$$

where $\eta > 0$ is a parameter that controls how quickly W_i is allowed to change. L_i is the category label of the document D_i. L_i is 1 if D_i is a positive document and 0 if D_i is a negative document. After all documents in the training document collection have been processed, the final generalized instance W_{n+1}, where n is total number of documents in the training document collection, is used as the weight vector for the category.

Suppose we treat the feature weight vector W as a special document D_W which summarizes all the original documents in the training collection. The decision of the assignment of the category can be viewed as considering the similarity between the request document X and this document D_W since the inner product is just a kind of similarity measure. Like the cosine similarity, the higher the metric value, the higher is the similarity. Note that it is equivalent to the cosine similarity if both vectors are normalized.

2.2 Existing Information Filtering Approaches

In this section, two recent information filtering systems are discussed, namely SIFT and NewsClip. Besides, two filtering approaches in TREC-6 are briefly described.

2.2.1 Information Filtering Systems

Stanford Information Filtering Tool (SIFT)

The Stanford Information Filtering Tool (SIFT) is a service provided over the World Wide Web (WWW) [38]. Users submit one or more subscription profiles, which are specified by a number of keywords, to the service using a WWW-browser. The profiles are matched against the entire news-feed articles maintained at Stanford. Matching articles are sent to the user by email.

There are two modes of communication between a user and a SIFT server, namely the interactive mode and the passive mode. In the interactive mode, the user can subscribe, test-run a profile, view, update, or cancel his or her subscriptions. In the passive mode, the user periodically receives information updates. The SIFT server sends out email messages that contain excerpts of news which are potentially relevant documents. After the user reads the excerpts, he may access the SIFT server to retrieve the entire documents.

The topic profile of SIFT can be expressed in one of two information retrieval models: the Boolean model and the vector space model. In the Boolean model, a Boolean conjunction of keywords is submitted. All keywords must

appear in the article in order to retrieve it. Keywords may be negated, in which case the presence of such a keyword will prevent the article from being retrieved. Multiple profiles must be submitted in order to provide a disjunction of terms. In the vector space model, a list of keywords in combination with a threshold set by the user. Each article is given a similarity score in the range 0-100. Articles that score above the threshold are retrieved.

The disadvantage of SIFT is to construct a topic profile by a user. This topic profile is either a set of keywords or keywords with their corresponding weights. The user have to care about which keywords should be put into the profile in order to specify his or her interests. Besides, multiple profiles must be submitted in order to provide a disjunction of terms in the topic profile of the Boolean model.

NewsClip

NewsClip is a programming language focused on news filtering. The complied filter operates off-line by comparing the user's *.newsrc* file with the contents of the news server. It applies its rules to all unread messages and rewrites the *.newsrc* file in such a way that messages that are filtered out are marked as already read. As a result, the user is free to employ any conventional news reading system.

NewClip allows a user to specify binary ratings. Articles rejected by the filter never reach the user since they have been marked as already read by NewsClip. However, the user must manually writes the filtering program using the specialized NewsClip language. The NewsClip program accepts, rejects or

weights articles based on C-like expressions that you write to describe what you want or do not want to see. For example, you might reject all articles in "rec.humor" that are cross-posted to "talk.bizarre," unless they are posted by a user at your own site with:

$$reject\ if\ is\ rec.humor\ \&\&\ is\ talk.bizarre\ \&\&\ domain(from)\ !=\ my\_domain;$$

The complied filter automatically applies the rules to the set of unseen messages. It runs as a batch program manually or automatically.

The major weakness of NewsClip is the requirement of specifying the user interest by NewsClip language. Users with no programming background cannot use these C-like expressions and , thus, this system is of no use for them.

2.2.2 Filtering in TREC

Two filtering approaches in the Sixth Text REtrieval Conference(TREC-6) [10] are briefly discussed. Basically, they formed the filtering query by using the Rocchio algorithm.

The University of Massachusetts develped the InRoute system in the filtering track [2]. InRoute is a variant of INQUERY modified to be more efficient for processing large numbers of queries on a stream of documents. InRoute is based on the basic retrieval model used in probabilistic belief network. The system uses a probabilistic belief network with the weighting scheme similar to the SMART weighting scheme.

During the training phase, InRoute gives the training document relevance

judgements. The unjudged documents are treated as not relevant. InRoute uses the Rocchio algorithm to perform the incremental profile updating. The threshold scores are modified to be halfway between the average relevant document score and the average nonrelevant score.

The AT&T Labs Research uses the inner-product similarity metric to find a "query-zone" [36]. They rank the training doucments by the inner-product similarity metric with the query and select the top 5000 documents for the query as the query-zone. Then, a feedback query is formed by summarizing the non-relevant document in the query-zone and all the relevent documents in the training corpus using the Rocchio's algorithm. The feedback query is further optimized to form the final filtering query. The filtering query is used to rank the training documents and a similarity threshold for the filtering query is selected that maximizes the evaluation measure on the training documents.

Chapter 3

Document Pre-Processing

In this chapter, we discuss some background of pre-processing of text documents and the classification scheme learning strategy [19].

3.1 Document Representation

The classification system first extracts indexes or identifiers which can characterize the documents. Identifiers are basically words or phrases in the content and they can be used to represent the document. This indexing process is a pre-processing step before the system conducts document classification. In the past, indexing was mostly performed by subject experts, or some by well trained persons with experience in assigning content descriptions. However, manual indexing is very time consuming. Besides, indexing experts may introduce unwanted variability and uncertainties that may adversely affect the clas-

sification result and retrieval effectiveness. An alternative approach is based on *automatic indexing* [34, 31].

With automatic indexing, terms are automatically extracted from the free-text vocabularies in documents. Consequently, this method does not need to know the subject area of the documents in advance. It can be applied to other areas without any manual configuration. We employ a method called the *term weighting scheme* [32].

The term weighting technique makes use of the number of occurrences of particular words in the documents of a collection [42, 24]. The main steps are described below:

1. Use a table, called the *stop list*, to eliminate common function words (e.g. and, of, an, but, the, etc.) from the text documents.

2. Each of the remaining words is reduced to a word-stem form so that all words exhibiting the same stem are represented in the same way (e.g. analyze, analyzes, and analyzing are all reduced to the stem ANALY) [12, 13].

3. Compute the term frequency f_{ij} for all stemmed words T_j in each document D_i.

As a result, each document is represented by a term vector of the form

$$D_i = (a_{i1}, a_{i2}, ..., a_{in})$$

where the coefficient a_{ik} represents the weight of the term T_k in document D_i.

These coefficients can be either binary or numeric. For the binary representation, a_{ik} is set to 1 when the term T_k is present in document D_i, and 0 when this term is absent. For numeric representation, the value of a_{ik} is determined from the effectiveness of this term to represent the document. Different kinds of numeric term weighting scheme have been proposed. One common method is the *term-frequency method* [32]. In this method, the value of a_{ik} is represented by the term frequency, f_{ik}, which is the number of occurrence of the term T_k in the document D_i. Thus we have

$$a_{ik} = f_{ik}$$

Another kind of representation is the *inverse document frequency method* [33, 32, 37]. In this method, we need to obtain the inverse document frequency, I_k, of a term T_k which is defined as:

$$I_k = \log \frac{N}{d_k}$$

where N is the number of documents in a collection and d_k is the number of documents in a collection in which the term T_k occurs. The weight a_{ik} is determined by :

$$a_{ik} = f_{ik} I_k$$

Once a document is represented as a vector, the similarity between document D_i and D_j, $SIM(D_i, D_j)$, can be calculated in a number of ways. One popular method is the inner product as follows:

$$SIM(D_i, D_j) = \sum_{k=1}^{n} a_{ik} \cdot a_{jk}$$

The similarity coefficient is in principle unbounded, it is customary in most applications to use normalized similarity coefficients whose value vary between 0 and 1 when the vector elements are nonnegative. Three typical normalized similarity coefficients of this kind are the Dice, cosine, and Jaccard coefficients as shown below.

Dice coefficient:

$$SIM(D_i, D_j) = \frac{2\sum_{k=1}^{n} a_{ik} \cdot a_{jk}}{\sum_{k=1}^{n} a_{ik}^2 + \sum_{k=1}^{n} a_{jk}^2}$$

Cosine coefficient:

$$SIM(D_i, D_j) = \frac{2\sum_{k=1}^{n} a_{ik} \cdot a_{jk}}{\sqrt{\sum_{k=1}^{n} a_{ik}^2 \sum_{k=1}^{n} a_{jk}^2}}$$

Jaccard coefficient:

$$SIM(D_i, D_j) = \frac{2\sum_{k=1}^{n} a_{ik} \cdot a_{jk}}{\sum_{k=1}^{n} a_{ik}^2 + \sum_{k=1}^{n} a_{jk}^2 - \sum_{k=1}^{n} a_{ik} \cdot a_{jk}}$$

Some advantages of the vector representation are the model's simplicity, the ease with which it accommodates weighted terms, and its ability to rank the retrieved documents.

3.2 Classification Scheme Learning Strategy

To tackle the automatic classification problem, we make use of a machine learning technique which discovers classification knowledge or scheme from a collection of training examples. Each example consists of a document and a set of manually assigned categories. Using the above representation, the training document collection becomes:

$$
\begin{array}{l|ll}
D_1 & a_{11}, ..., a_{1n} \ ; & C_{11}, ..., C_{1m_1} \\
D_2 & a_{21}, ..., a_{2n} \ ; & C_{21}, ..., C_{2m_2} \\
\vdots & & \\
D_t & a_{t1}, ..., a_{tn} \ ; & C_{t1}, ..., C_{1m_t}
\end{array}
$$

where a_{ij} denotes the weight of term T_j in document D_i and C_{ik} denotes a certain category assigned to document D_i.

Table A.1 shows a piece of sample document in Reuter-21578 document collection. The Reuters-21578 document collection will be discussed in Chapter 5 in more details. The documents of Reuters-21578 are in SGML format. Each document starts with an "open tag" of the form <REUTERS ... > and end with an "close tag" of the form < /REUTERS>. The list of TOPICS categories for the document are enclosed by the tags <TOPICS> and < /TOPICS>. If categories are present, each will be delimited by the tags <D> and < /D>. The main text of the document is enclosed by the tags <BODY> and < /BODY>. The sample document shown in Table A.1 can be represented as:

figur, regist, ..., linoil, show ; veg-oil, linseed, ..., wheat.

where "figur", "regist" and "linoil" are the stemmed words selected from the document. "veg-oil", "linseed", and "wheat" are the pre-defined TOPICS category labels in the document collection.

This classification scheme learning problem can be decomposed into subproblems related to individual categories. Since a fixed set of categories is

```
<REUTERS TOPICS="YES" LEWISSPLIT="TRAIN" CGISPLIT="TRAINING-SET"
OLDID="5549" NEWID="6">
<DATE>26-FEB-1987 15:14:36.41</DATE>
<TOPICS><D>veg-oil</D><D>linseed</D><D>lin-oil</D><D>soy-oil</D>
<D>sun-oil</D><D>soybean</D><D>oilseed</D><D>corn</D><D>sunseed</D>
<D>grain</D><D>sorghum</D><D>wheat</D></TOPICS>
<PLACES><D>argentina</D></PLACES>
<PEOPLE></PEOPLE>
<ORGS></ORGS>
<EXCHANGES></EXCHANGES>
<COMPANIES></COMPANIES>
<TEXT>
<TITLE>ARGENTINE 1986/87 GRAIN/OILSEED REGISTRATIONS</TITLE>
<DATELINE>    BUENOS AIRES, Feb 26 - </DATELINE>
<BODY>
Argentine grain board figures show crop registrations of grains,
oilseeds and their products to February 11, in thousands of tonnes,
showing those for future shipments month, 1986/87 total and 1985/86
total to February 12, 1986, in brackets:

  Bread wheat prev 1,655.8, Feb 872.0, March 164.6, total 2,692.4 (4,161.0).
  Maize Mar 48.0, total 48.0 (nil).
  Sorghum nil (nil)

  Oilseed export registrations were:
  Sunflowerseed total 15.0 (7.9)
  Soybean May 20.0, total 20.0 (nil)

  The board also detailed export registrations for subproducts, as follows,
  SUBPRODUCTS

  Wheat prev 39.9, Feb 48.7, March 13.2, Apr 10.0, total 111.8 (82.7).
  Linseed prev 34.8, Feb 32.9, Mar 6.8, Apr 6.3, total 80.8 (87.4).
  Soybean prev 100.9, Feb 45.1, MAr nil, Apr nil, May 20.0,
  total 166.1 (218.5).
  Sunflowerseed prev 48.6, Feb 61.5, Mar 25.1, Apr 14.5,
  total 149.8 (145.3).

  Vegetable oil registrations were :
  Sunoil prev 37.4, Feb 107.3, Mar 24.5, Apr 3.2, May nil,
    Jun 10.0, total 182.4 (117.6).
  Linoil prev 15.9, Feb 23.6, Mar 20.4, Apr 2.0, total 61.8, (76.1).
  Soybean oil prev 3.7, Feb 21.1, Mar nil, Apr 2.0, May 9.0,
    Jun 13.0, Jul 7.0, total 55.8 (33.7).

</BODY>
</TEXT>
</REUTERS>
```


Table 3.1: A sample document in the Reuters-21578 corpus.

known in advance, we can learn a separate classification scheme for each category from the training document collection. After the classification schemes of all categories are discovered, they can be used together in the on-line classification module to decide a set of categories for a new document. Suppose we have categories $C_1, ..., C_m$. Consider a particular category C_j, We try to learn a classification scheme for C_j. The training document collection for the category C_j can be viewed as:

D_1	$a_{11}, ..., a_{1n}$;	1
D_2	$a_{21}, ..., a_{2n}$;	0
$\vdots$		
D_t	$a_{t1}, ..., a_{tn}$;	1

The last column denotes the membership of the category to a document. If a document belongs to category C_j, the value of this entry is set to 1, otherwise, the value is set to 0. Using this training document collection, we can apply machine learning techniques to construct automatically a classification scheme for that category. Each term is regarded as a feature. Each document is represented as a feature vector. After all categories have been learned, we have m classification schemes available. The incoming document is converted into a system readable format and it is then matched against each classification scheme. Each classification scheme output its decision. This decision can be a binary decision or a weighted decision. The system finally decides a set of categories assigned to this document based on these decisions. Figure 3.1 depicts an example showing how classification schemes of separate categories

are used together to decide the set of categories for a new document.

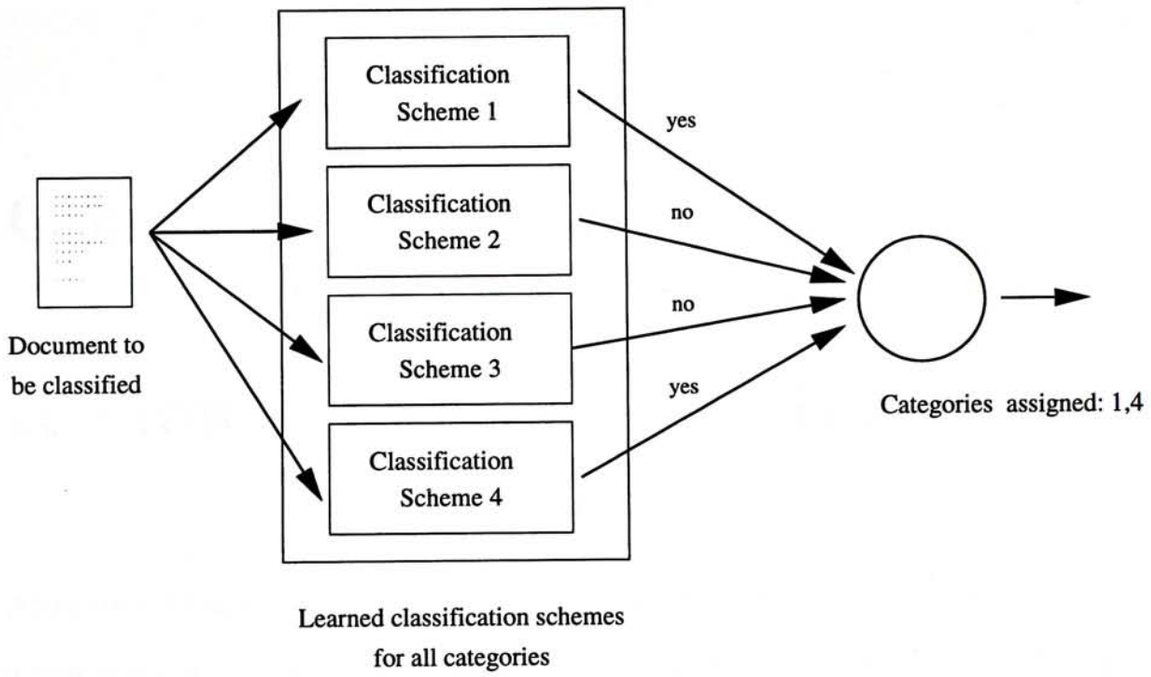

Figure 3.1: An example showing how classification schemes of separate categories can be used together in on-line classification to decide a set of categories for a new document.

Chapter 4

A New Approach - IBRI

After reviewing some existing approaches in automatic document classification, a new approach is proposed called IBRI that unifies the strengths of rule-based learning and instance-based approaches as well as adapts to characteristics of document categorization problems [15].

4.1 Overview of Our New IBRI Approach

There has been some research conducted for automatic document classification task as discussed in Chapter 2. The SWAP-1 and RIPPER algorithms are rule-based learning algorithms. The separate-and-conquer strategy of rule-based learning algorithms in the induction phase may introduce deficiency in document classification problems. It causes a dwindling number of examples to be available as the induction progresses. This effect may cause later rules

and later conditions within each rule to be learned with insufficient statistical support. This leads to greater noise sensitivity and thus discovering inaccurate rules.

Our new algorithm, called the IBRI algorithm(Instance-Based and Rule-Induction algorithm), attempts to incorporate the advantages of the instance-based technique into a rule-based approach. Rule-based approaches perform well at finding simple axis-parallel frontiers and are best suited to symbolic domains [3, 39, 5]. A typical rule-based classification scheme for a category, say C_j, has the form:

Assign category C_j	IF <*antecedent*> or
Do not assign category C_j	IF <*antecedent*>

The <*antecedent*> in the premise of a rule usually involves some kind of feature value comparison. The learning task can be viewed as a two-class learning problem. One class corresponds to a positive assignment of category C_j whereas the other class corresponds to a negative assignment. Rule induction methods attempt to find a compact covering rule set that completely partitions the examples into their correct classes. A rule is said to cover a document or a document is said to satisfy a rule if all the feature value comparisons in the antecedent of the rule are true for the document. The feature-valued rules in SWAP-1 [3] and set-valued rules in RIPPER [5] share some common properties.

The instance-based learning algorithm classifies new document by finding the "nearest" stored document, also known as an instance, in the training

document collection and deciding the assignment of the category based on this instance [1, 8, 9, 14]. This is a direct application of the concept of similarity or distance between documents. The advantages of this approach include its simplicity, low updating costs and inducing complex frontiers from relatively few documents. However, the learning performance is highly sensitive to the number of irrelevant features used to describe documents. Its storage requirement and computational cost increase exponentially with increasing number of learning documents.

In our IBRI algorithm, there is a rule induction process which tries to discover rules. One feature of our algorithm is that a learned rule may be an ordinary rule or a single example (i.e., an instance). To support this feature, we introduce a single distance metric which can measure the distance between a rule and a document as well as between an instance and a document. The IBRI algorithm consists of three phases. They are the Sampling, the Rule Induction and the Rule Refinement phases. The purpose of the Sampling phase is to extract the representative sample documents from the training document collection. We develop an advanced sampling technique to achieve this task.

The second phase is the Rule Induction phase. Unlike conventional rule induction algorithms, the learning process of our IBRI approach is specific-to-general. Rules are generalized by dropping contradicting features. Besides, documents covered by the rule are not removed. This conquering-without-separating strategy differs from the previous separate-and-conquer one

in which the covered documents are removed. Existing rule-based learning algorithms search for a complex or compact "covering" rule set or decision tree that partitions the training examples into their correct classes. This may not be effective in document classification problem. Our IBRI approach searches for a more robust rule set by considering the distance between rules and documents.

After we obtain a rule set from the Rule Induction phase, we conduct the Rule Refinement phase. In this phase, the generalized rule set is used to classify the training document collection again. The rules which win very few documents and have high error rate are dropped.

4.2 The IBRI Representation and Definitions

As mentioned above, there is a set of rules associated with each category in our IBRI approach. A rule can be expressed as:

$$
\begin{array}{ll}
C & \leftarrow <condition> \oplus\beta_{11}, <condition> \oplus\beta_{12}, \ldots \text{ OR} \\
C & \leftarrow <condition> \oplus\beta_{21}, <condition> \oplus\beta_{22}, \ldots \text{ OR} \\
\vdots & \\
\neg C & \leftarrow <condition> \oplus\beta_{11}, <condition> \oplus\beta_{12}, \ldots \text{ OR} \\
\neg C & \leftarrow <condition> \oplus\beta_{21}, <condition> \oplus\beta_{22}, \ldots \text{ OR} \\
\vdots &
\end{array}
$$

where C denotes the fact that the category C should be assigned and $\neg C$ denotes the fact that category C should not be assigned. The antecedent part is a conjunction of conditions. Each condition involves one feature value test. An instance example can be viewed as a rule which has a positive condition

for each feature present in the document. The meaning of $\oplus$ and β_{ij} will be explained below.

Let $D = (a_1, a_2, ..., a_n, c)$ be an example with value a_i for the feature i. c is 1 or 0 denote whether the document belongs to the category. a_i is 1 if the feature i is present in the document, otherwise, d_i is set to 0. Let $L = (r_1, r_2, ..., r_n, c)$ be a rule with condition r_i on the feature i. r_i being 1 denotes the fact that the feature i is present in the document. r_i being 0 denotes the fact that the feature i is not present in the document. r_i being -1 means that the feature i can be ignored in this rule. The distance $DIST(L, D)$ between a rule, L , and a document, D , is then defined as:

$$DIST(L, D) = \sum_{i=1}^{n} dist(a_i, r_i);$$

$$dist(e_i, r_i) = \begin{cases} 0 & \text{if } (a_i = \text{-1}) \text{ or } (a_i = r_i) \\ VDM(a_i, r_i) & \text{otherwise.} \end{cases}$$

where VDM is the Value Different Metric [7]. To explain VDM, consider the following table that records the frequency distribution of the feature A in positive and negative documents.

	Negative Document	Positive Document
Not Contain A	a	b
Contain A	c	d

where a is the number negative document not containing feature A; b is the number positive document not containing feature A; c is the number negative document containing feature A; and d is the number positive document

containing feature A. The VDM is defined as:

$$VDM(0,1) = VDM(1,0) = \left|\frac{a}{(a+b)} - \frac{c}{(c+d)}\right| + \left|\frac{b}{(a+b)} - \frac{d}{(c+d)}\right|$$

The VDM is a very good method for measuring the distance of features in domains with symbolic feature values. It takes into account the overall similarity of document classification for each possible value of each feature. Here, the VDM has been modified to consider two dichotomous category labels. The idea behind this metric is that values of a feature are similar if they occur with the same relative frequency for all categories. The distance between values of a feature can be derived statistically based on documents in the training document collection. An example of a rule set is :

sports	$\leftarrow$	sport$\in$document$\oplus$1.02.
sports	$\leftarrow$	exercise$\in$document$\oplus$0.5, outdoor$\in$document$\oplus$0.2.
sports	$\leftarrow$	exercise$\in$document$\oplus$1.4, homework$\notin$document$\oplus$0.07, ...
sports	$\leftarrow$	play$\in$document$\oplus$0.3, rule$\in$document$\oplus$1.8.
$\vdots$		
$\neg$sports	$\leftarrow$	food$\in$document$\oplus$1.3, cook$\in$document$\oplus$1.1.
$\neg$sports	$\leftarrow$	business$\in$document$\oplus$0.9, ...
$\vdots$		

When a document tries to match a rule, we calculate the distance between them. $a \oplus b$ means that distance b is added to the total distance if condition a is not valid. A rule is said to completely cover the document if the distance between them is zero, that is all conditions are true. A rule is said to "win" the document if it is the nearest rule to the document according to the distance metric. Therefore, a rule can win a document even if it does not completely

cover it. This is one main difference between our IBRI approach and other rule-based approaches, in which an accepted rule must cover some documents. We make use of the learned rule set to conduct on-line classification of a new document. Basically, we calculate the distance described above between the document and each rule. Then, the category assignment of the nearest rule is applied to this document.

4.3 The IBRI Learning Algorithm

Figure 4.1 shows the main steps in our IBRI approach. Let the training documents be E. Steps 1 and 2 corresponds to the Sampling phase. Step 3 to Step 10 corresponds to the Rule Induction phase, and Step 11 is the Rule Refinement phase. In the Sampling phase, the rule set R is sampled from the training documents E. Each rule in R at the output of the sampling is actually an instance (i.e., individual document). We develop a new technique to achieve this sampling task. At the beginning, K positive documents are randomly selected from E. The selected documents are generalized into a generalized rule by dropping those conditions not satisfying these K positive documents. Next, the training documents are ordered by their distances against this generalized rule. Then, x positive documents and y negative documents are evenly sampled from the ordered documents. This sampling strategy can ensure that the sampled documents are representative with respect to the training document collection. To make sure no useful document is missed, the rule set is used to

Procedure $IBRI(E)$

1 $R := Sampling(E)$
2 $S := R$

3 For each e in E, find the nearest rule r of e in R

4 For each r in R
5 Repeat
6 $r' := r$
7 Find the nearest example e of r' in S with the same category assignment
8 $r := Generalization(r', e)$
9 Until $(Utility(r, E) < 0)$
10 $r := r'$

11 $R := Refinement(R,E)$

Figure 4.1: The IBRI Approach

classify the training set E. The misclassified documents are added to the rule set to form the initial rule set R.

Before the Rule Inducting phase, the rule set R is used to classify the documents in E. Each document memorizes the distance to the nearest rule and the category assignment of that rule. This rule must not be the same document itself. This information is used to evaluate whether the generalized rule can "win" any document that it did not do so before.

In the induction process, each rule repeatedly finds the nearest document of the same category assignment and the rule does not cover this document before. Then it attempts to minimally generalize itself to cover it (i.e., the *Generalization* function in Step 8). This generalization is done by dropping conditions in which features are not satisfied by the nearest document. Every time when a rule is generalized, the rule is matched against all documents in E and the utility of this rule is calculated. The function *Utility* is responsible for calculating the utility of a rule as shown in Figure 4.2. Since each document memories its nearest rule and the distance with this rule, all we need to do is to check whether this new rule "wins" some documents that it did not win before. Therefore, only those documents that are misclassified previously and of the same category assignment with the rule need to be checked. If a previously misclassified document is now correctly classified, the utility is incremented either by *p_cost* or *n_cost* according to the category of the training document. If the reverse condition holds, the utility is decreased. If the final sum of increments and decrements is greater than or equal to zero, the new generalized

function $Utility(r, E)$

$utility := 0$

For each e in E

    If r wins e and it does not win before

        If (e is positive document)

            If r and e have the same category assignment

                $utility := utility + p\_cost$

            Else

                $utility := utility - p\_cost$

        Else

            If r and e have the same category assignment

                $utility := utility + n\_cost$

            Else

                $utility := utility - n\_cost$

return $utility$

Figure 4.2: The function *Utility*

rule is accepted. A generalized rule is accepted even if it apparently has the same utility as the old one. This is due to the Occam's Razor principle: when two theories appear to perform identically, the simpler one is preferred.

This induction process is repeated until the utility of a rule is less than zero. In a case where no generalization is done due to low utility, the initial sample document will be included, leading to an instance accepted into the final rule set. Therefore, the final rule set, in general, may contain some individual documents as well as ordinary rules.

The final step is to refine the generalized rule set by a statistical test. Duplicating rules are first removed. The generalized rules are used to classify the original training set E. By memorizing number of times a rule is used and number of times it classifies correctly, we can compute the predictive power of a rule. The predictive power of a rule is defined as the fraction of correct category assignments assigned by the rule. Only those rules having predictive power greater than a threshold will be accepted into the final rule set. This process can make sure that the over-fitted rule and noisy example documents will be discarded. For document classification, the negative documents are usually unstructured and large in size. Therefore, we have to choose the negative rules more carefully by setting the predictive power thresholds relatively higher than that of positive rules. These predictive power thresholds can also be used to control the recall and precision of the classification system. For instance, by setting the predictive power threshold of positive rule higher, the system tends to classify a document to be negative, thus, leading to higher precision but

lower recall.

Chapter 5

IBRI Experiments

We have implemented our IRBI approach and some variants of it. Extensive experiments have been conducted on a large-scale, real-world document corpus, namely the Reuters-21578 collection. The results show that our new approach outperforms RIPPER and instance-based approaches.

5.1 Experimental Setup

We have conducted experiments on a commonly used document collection, namely the Reuters-21578 collection which is a revised version of Reuters-22173. Previous experiments were conducted based on old Reuters-22173 collection. Therefore, the results of Reuters-22173 and Reuters-21578 cannot be compared directly. The Reuters-21578 collection contains Reuters newswire articles in 1987. The documents were assembled and labeled with categories

by experts from Reuters. There are 2,1578 documents in this collection. However, not all of the documents in this collection are properly categorized by experts. We follow the "ModApte" split which removed the documents without properly categorized and then divided the collection into a training document collection and a testing document collection. There are 9,603 training documents and 3,299 testing documents. There are 8,676 documents are removed. For each category, we used the training document collection to learn a classification scheme. To evaluate the effectiveness of the learned scheme, we used the scheme to classify documents in the testing document collection and compared the result with the manual classification. We chose those 90 categories that appear in at least one training document and one testing document.

We have implemented our IBRI algorithm. To demonstrate the importance of the sampling and the rule learning processes in our approach, we have implemented an instance-based algorithm called IB that uses all the training documents as the learned rule set. We have also implemented a variant of our IBRI algorithm called the SIB algorithm that only consists of the Sampling phase and the Rule Refinement phase depicted in Step 1 and Step 11 in Figure 4.1. These learning approaches including IB, SIB, RIPPER, and IBRI were used in our experiments. For SIB and IBRI approaches, x is set to at most 100 positive training documents and y is set to at most 100 negative training documents in the Sampling phase. For RIPPER, the system is downloaded form public domain "http://www.research.att.com/ wcohen/ripperyes.html" and we used the heuristic measures used currently in the system.

5.2 Evaluation Metric

For evaluation, the classification performance of each category is measured. The overall effectiveness is then computed by calculating the mean across all categories. Consider a particular category, the effectiveness of the classification can be illustrated in a contingency table as follows [21]:

	Expert Says Yes	Expert Says No	
System Says Yes	q	r	$q+r$
System Says No	s	t	$s+t$
	$q+s$	$r+t$	$q+r+s+t$

where q is the number of documents belonging to the category and assigned to the category; r is the number of documents not belonging to the category but assigned to the category; s is the number of documents belonging to the category but not assigned to the category; t is the number of documents not belonging to the category and not assigned to the category. Some common effectiveness measures can then be defined in terms of these values:

$$(recall)\ R = \frac{q}{(q+s)}$$

$$(precision)\ P = \frac{q}{(q+r)}$$

Recall is the proportion of documents belonging to the category that the system successfully assigns to the category. Precision is the proportion of documents assigned to the category by the system that really belong to the category. An ideal classification system would have both recall and precision equal to 1.

However, perfect recall can be achieved by a system that puts every document in the category, while perfect precision can be achieved by a system that puts no documents in the category. Therefore, just using either recall or precision does not provide a fair evaluation to system. Hence, an effectiveness measure called F measure, proposed by Lewis [22], combines the recall and precision into a single score as follow:

$$F_\alpha = \frac{(\alpha^2 + 1)PR}{\alpha^2 P + R}$$

α ranges from 0 to infinity. For α equals to 0, F_α is the same as the precision. For α equal to infinity, F_α is the same as the recall. The variation of α between 0 and infinity corresponds to the relative weight associated with recall and precision. In order to strike a balance between the recall and precision, α is set to 1 in our experiments which gives an equal weight on the recall and precision in this effectiveness measure. After F_1 measures of all categories are computed, then mean F_1 value across categories is computed so as to get the overall evaluation of a classification system. We call it average F_1 measure, namely AFM.

5.3 Results

Table 5.1 summarizes the F_1 results of 90 categories used in the experiment. Table 5.2 highlights the F_1 results of the ten most frequent categories. The results show that seven out of ten categories outperform the other algorithms.

Category	IB	SIB	RIPPER	IBRB	Category	IB	SIB	RIPPER	IBRB
acq	0.718	0.803	0.839	0.876	meal-feed	0.621	0.774	0.811	0.718
alum	0.467	0.452	0.611	0.857	money-fx	0.650	0.650	0.516	0.631
barley	0.783	0.933	0.933	0.933	money-supply	0.346	0.346	0.529	0.357
bop	0.613	0.506	0.571	0.553	naphtha	0.000	0.000	0.000	0.000
carcass	0.471	0.516	0.390	0.651	nat-gas	0.440	0.440	0.473	0.500
castor-oil	0.000	0.000	0.000	0.000	nickel	1.000	1.000	1.000	1.000
cocoa	0.759	0.812	0.973	0.857	nkr	0.000	0.000	0.000	0.000
coconut	0.667	0.000	0.800	0.000	nzdlr	0.000	0.000	0.000	0.000
coconut-oil	0.000	0.000	0.000	0.000	oat	0.667	0.727	0.600	0.857
coffee	0.931	0.900	0.900	0.900	oilseed	0.740	0.759	0.775	0.721
copper	0.500	0.667	0.919	0.545	orange	0.778	0.778	0.846	0.842
copra-cake	0.000	0.000	0.000	1.000	palladium	0.000	0.000	0.000	0.000
corn	0.852	0.862	0.881	0.901	palm-oil	0.889	0.952	0.800	0.952
cotton	0.518	0.927	0.952	0.952	palmkernel	0.000	0.000	0.000	0.000
cotton-oil	0.000	0.000	0.667	0.000	pet-chem	0.000	0.000	0.261	0.000
cpi	0.353	0.461	0.429	0.396	platinum	0.000	0.000	0.000	0.000
cpu	0.400	0.000	0.000	0.000	potato	0.500	1.000	0.000	1.000
crude	0.596	0.754	0.746	0.734	propane	0.000	0.000	0.000	0.000
dfl	0.000	0.000	0.000	0.000	rand	0.000	0.000	0.000	0.000
dlr	0.552	0.325	0.621	0.452	rape-oil	0.000	0.000	0.000	0.000
dmk	0.000	0.000	0.000	0.000	rapeseed	0.875	0.800	0.800	0.800
earn	0.942	0.929	0.958	0.961	reserves	0.440	0.550	0.605	0.395
fuel	0.143	0.333	0.308	0.333	retail	0.400	0.174	0.000	0.500
gas	0.774	0.850	0.842	0.914	rice	0.400	0.873	0.873	0.873
gnp	0.645	0.794	0.720	0.767	rubber	0.700	0.609	0.667	0.667
gold	0.216	0.615	0.679	0.716	rye	0.000	0.000	0.000	0.000
grain	0.848	0.858	0.908	0.871	ship	0.361	0.482	0.780	0.784
groundnut	0.000	0.750	0.000	0.750	silver	0.667	0.400	0.200	0.615
groundnut-oil	0.000	0.000	0.000	0.000	sorghum	0.625	0.769	0.769	0.769
heat	0.667	0.400	0.667	0.400	soy-meal	0.353	0.560	0.750	0.560
hog	0.500	0.500	0.588	0.500	soy-oil	0.143	0.471	0.250	0.444
housing	0.571	0.667	0.667	0.667	soybean	0.712	0.738	0.732	0.729
income	0.444	0.000	0.250	0.250	strategic-metal	0.000	0.000	0.143	0.000
instal-debt	0.000	0.000	0.000	0.000	sugar	0.923	0.868	0.868	0.868
interest	0.500	0.504	0.393	0.579	sum-meal	1.000	1.000	0.000	1.000
ipi	0.256	0.211	0.261	0.333	sum-oil	0.000	0.000	0.000	0.000
iron-steel	0.421	0.476	0.625	0.522	sumseed	0.333	0.333	0.571	0.333
jet	0.000	0.000	0.000	0.000	tea	0.000	0.000	0.400	0.000
jobs	0.889	0.870	0.870	0.870	tin	0.737	0.588	0.889	0.588
l-cattle	0.000	0.000	0.000	0.000	trade	0.498	0.562	0.634	0.657
lead	0.133	0.692	0.235	0.692	veg-oil	0.556	0.738	0.744	0.741
lei	1.000	0.500	0.800	0.500	wheat	0.874	0.870	0.885	0.898
lin-oil	0.000	0.000	0.000	0.000	wpi	0.385	0.533	0.333	0.706
livestock	0.511	0.474	0.698	0.667	yen	0.182	0.471	0.000	0.247
lumber	0.000	0.000	0.250	0.000	zinc	0.727	0.788	0.812	0.788

Table 5.1: F_1 measures of categories with at least one positive training document and one positive testing document

Category	IB	SIB	RIPPER	IBRB
acq	0.718	0.803	0.839	**0.876**
corn	0.852	0.862	0.881	**0.901**
crude	0.596	**0.754**	0.746	0.734
earn	0.942	0.929	0.958	**0.961**
grain	0.848	0.858	**0.908**	0.871
interest	0.500	0.504	0.393	**0.579**
money-fx	0.650	**0.650**	0.516	0.631
ship	0.361	0.482	0.780	**0.784**
trade	0.498	0.562	0.634	**0.657**
wheat	0.874	0.870	0.885	**0.898**

Table 5.2: F_1 measures of the ten most frequent categories

	IB	SIB	RIPPER	IBRB
High Frequent Categories(10)	0.684	0.727	0.754	0.789
Low Frequent Categories(53)	0.287	0.292	0.292	0.331
All Categories(90)	0.413	0.444	0.458	0.483

Table 5.3: Average F_1 measures of categories

Table 5.3 summarizes the average F_1 across categories with different number of positive documents in the training document collection. The first row summarizes the average F_1 of the top ten most frequent categories (the number of positive documents in the training document collection is larger than 141). The second row summarizes the average F_1 of the low frequent categories (the number of positive documents in the training collecting is less then 30). There are 53 categories in this group. The third row summarizes the average F_1 of all categories. These results illustrate that our IBRI approach has 4.6%, 13.4% and 5.5% improvement over RIPPER in the above three groups respectively. The improvements are significant in the group of low frequent categories.

The results also show that the algorithm with document sampling (i.e., the SIB algorithm) outperforms the IB algorithm. The performance is further improved if our rule induction process is added, leading to our IBRI algorithm.

Chapter 6

A New Approach - GIS

Based on the idea of IBRI, we further investigate several recent approaches for document classification under the framework of similarity-based learning. They include two families of techniques, namely the k-nearest neighbor (k-NN) algorithm and linear classifiers. After identifying the weakness and strength of each technique, we propose another new technique known as the generalized instance set (GIS) algorithm by unifying the strengths of k-NN and linear classifiers and adapting to characteristics of document classification problems [18].

6.1 Motivation of GIS

Although IBRI performs very well in automatic document categorization problem, IBRI is not a suitable approach to solve the document filtering problem. Firstly, the learning of rule in IBRI is time consuming as each rule has to

repeatedly search the "nearest" instance. Secondly, the refinement of induced rule does not guarantee the removal of low predictive rules and these predictive rules would not induce error to other rules. Finally, the VDM of each attribute should be kept in the system. Therefore, GIS is developed by trying to generalize and refine the training instances in a more effective way by unifying the k-NN algorithms and linear classifier algorithms.

6.2 Similarity-Based Learning

Some recent document classification learning approaches can be regarded as similarity-based algorithms. In these algorithms, each document is mapped to an internal representation. A metric measuring the similarity of two documents is then designed. This similarity metric is used during the training phase as well as in the online classification. We have discussed two families of latest similarity-based document classification learning algorithms in Chapter 2, namely, the k-NN algorithm and linear classifiers.

Conceptually, a classifier is learned for each category given a category. The training document collection consists of positive and negative documents. Each training document, represented by a vector, is regarded as an *instance*. In the following discussion, we denote D_j as an instance in the training collection and it is represented as $(a_{1j}, \ldots, a_{nj})$. We denote X as a request document to be categorized and it is represented as $(x_1, \ldots, x_n)$. A common weighting schemes called the inverse document frequency scheme as described in Chapter 3 is used.

The weights are then normalized by multiplying each element in the vector by a constant $1/\sqrt{\sum_{i=1}^{n} a_{ij}^2}$.

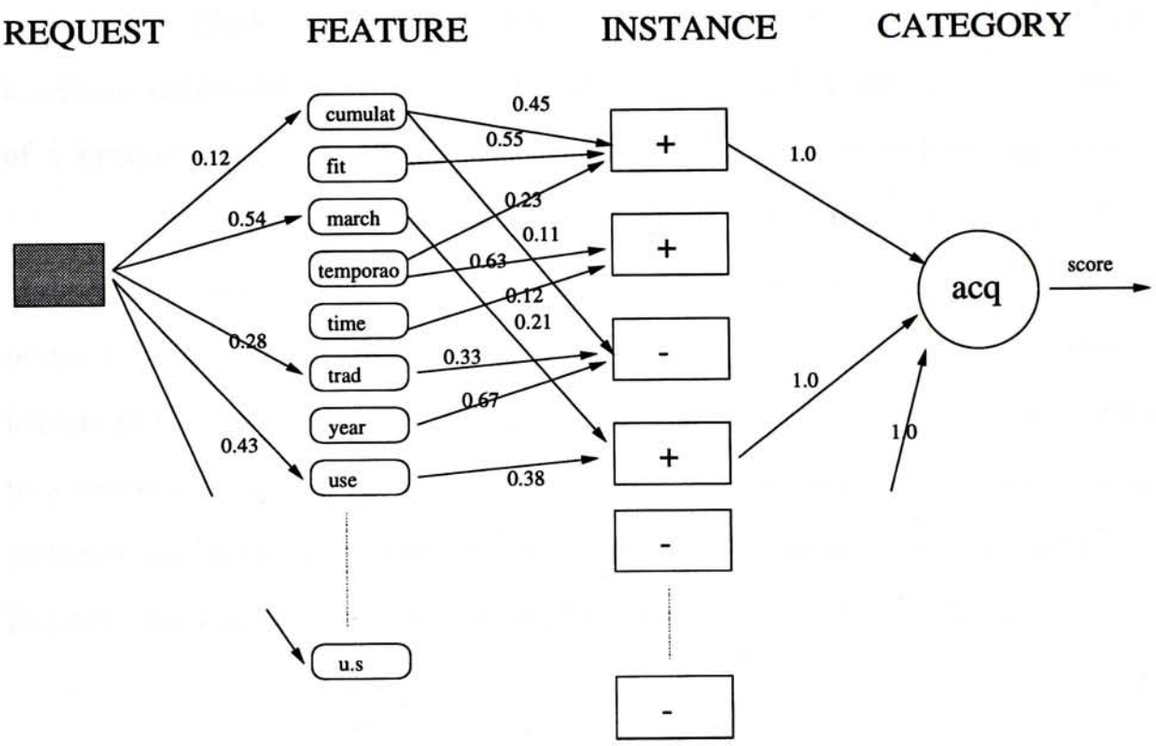

Figure 6.1: A k-NN algorithm

K-nearest neighbor (k-NN) algorithms are one kind of similarity-based learning. One recent example of a k-NN algorithm for document classification is known as the Expert Network (ExpNet), as described in Chapter 2, and it achieves good performance [41]. In a k-NN algorithm, each training document D_j as well as the request document X are represented by vectors as described above. To conduct classification, the similarity $\Delta(X, D_j)$ between each D_j and X is calculated. The training instances are sorted by the similarity metric in descending order. Then the k top-ranking instances are selected.

The final score of the request document to each category is calculated by considering the similarity metric of these k selected instances and their category association. The document is assigned to categories with the score greater than a certain threshold value. Figure 6.1 illustrates a k-NN algorithm by means of a network representation. The network consists of three levels of nodes. The input nodes represent unique features in the training documents. The nodes in the intermediate level represent the training instances. The output nodes represent the categories in the corpus. A weight on the link between a feature in the input level and an instance in the intermediate level corresponds to a numeric weight in the document vector. A weight on the link between an instance and a category reflects the category association to a document. In ExpNet, the cosine similarity defined below is used for the metric Δ:

$$\Delta(X, D_j) = \frac{\sum_{i=1}^{n} x_i a_{ij}}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} a_{ij}^2}}$$

Also a weight on the link between an instance and a category can take on a value between 0 and 1. To implement k-NN algorithms, we can compute the similarity metrics of all training instances to the request document once. This set of similarity values can be used as a single classifier which can then be employed used for computing the final score for each category. Nevertheless, analytically, each category still corresponds to a separate classifier.

Linear classifiers are a family of document classification learning algorithms recently explored by Lewis [22]. We briefly describe this algorithm and point out its relationship with similarity-based algorithms.

For every category, there is a feature weight vector

$$W = (w_1, \ldots, w_n)$$

and each element w_i corresponds to the i-th feature. The elements in vector W are learned from all training examples including positive and negative instances. There are several weight learning techniques such as Rocchio [30] and Widrow-Hoff (WH) algorithm [40] . To determine whether or not a category is assigned to the request document X, it computes the inner product δ between the document vector X and the feature weight vector W as follows:

$$\delta = \sum_{i=1}^{n} x_i w_i$$

If the inner product is greater than a certain threshold value, the category is assigned to X. Figure 6.2 illustrates the linear classifier approach by means of a network representation.

Suppose we treat the feature weight vector as a special instance I which summarizes all the original instances in the training collection. The decision of the assignment of the category can be viewed as considering the similarity between the request document and this instance I since the inner product is just a kind of similarity measure. Like the cosine similarity, the higher the metric value, the higher is the similarity. Note that it is equivalent to the cosine similarity if both vectors are normalized. We call such special instance a *generalized instance* (GI).

k-NN algorithms directly make use of the training examples as instances for computing the similarity. One of the shortcomings is their sensitivity to

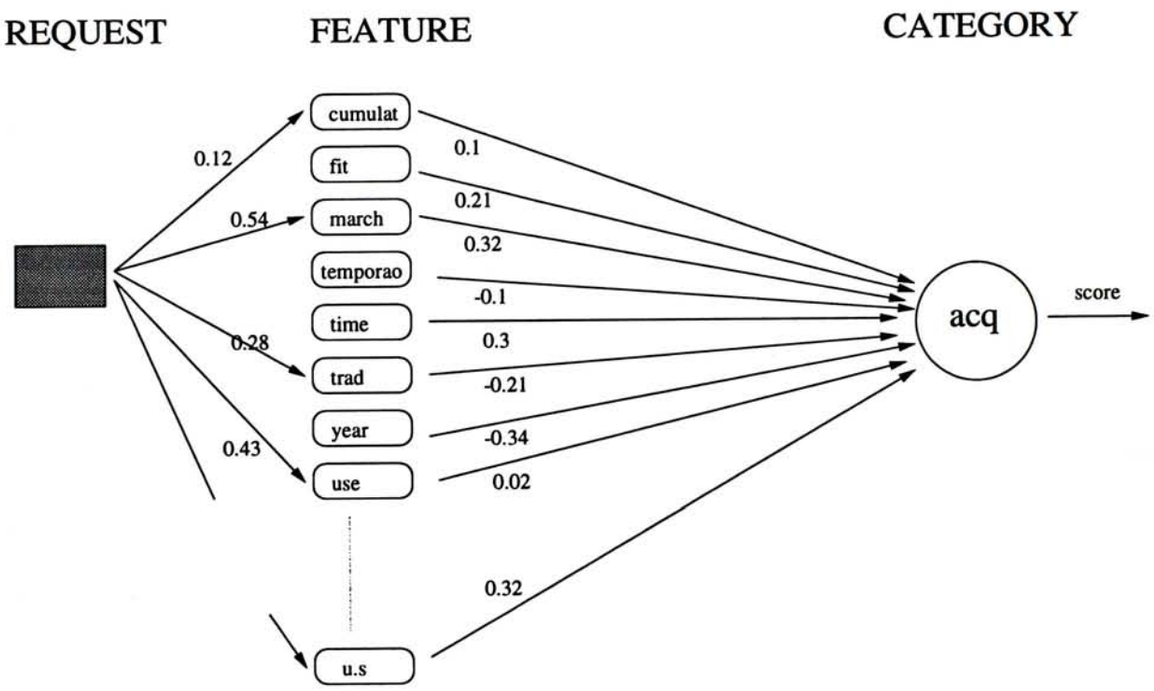

Figure 6.2: A linear classifier

noisy examples. Classification error occurs if the training instances in the neighborhood region of the request document are influenced by noisy examples. In document classification problems, it is quite common that instances in the training document collection contain some amount of noise due to various reasons such as missing appropriate features, typographical errors in texts, or wrong category assigned by human. The presence of such noisy examples will affect the classification performance. Besides, k-NN algorithms do not cope with irrelevant features effectively.

Linear classifiers can deal with noise to some extent via the generation of the generalized instance (GI). Since the GI replaces the whole collection of training instances by summarizing the contribution of positive and negative instances. As a result, the classification decision is not easily affected by noise. Besides, if a feature mainly appears in many positive training instances, its corresponding weight in the GI will have a larger magnitude. This is also true if a feature mainly appears in negative training instances. If a feature appears in negative and positive instances with approximately equal proportion, its weight in the GI will tend to zero. Therefore, linear classifiers can distill out certain relevant features to some extent. On the other hand, one drawback of linear classifiers is that they restrict the hypothesis space to the set of linear separable hyper-plane regions which has less expressiveness power than that of k-NN algorithms. In fact, a linear classifier can be viewed as a restricted representation of a k-NN algorithm. Figure 6.3 illustrates the relationship between a linear classifier and a k-NN algorithm. The shaded region is a

positive region containing the GI. Any document falls into this region will be classified as positive for the category by the linear classifier since its similarity to the GI is larger than the threshold. The dotted line represents a hyper-surface whose similarity to the boundary of the shaded region is the same as the similarity between the GI and the boundary of the shaded region. If we imagine that we place sufficiently many positive examples at the position of GI and sufficiently many negative examples along the hyper-surface, a k-NN algorithm under this particular example distribution is essentially equivalent to the original linear classifier.

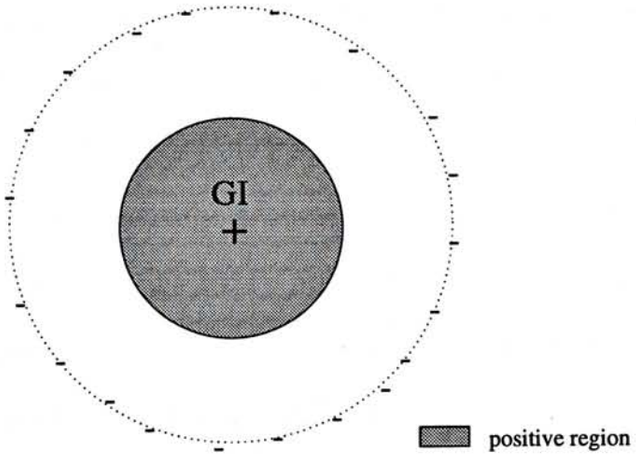

Figure 6.3: The relationship between a linear classifier and a k-NN algorithm

6.3 The Generalized Instance Set Algorithm (GIS)

After identifying the weakness of the existing similarity-based learning algorithms on document classification, we propose a new technique called the generalized instance set (GIS) algorithm which overcomes the weakness by unifying the strengths of the k-NN algorithm and linear classifiers and taking into account some characteristics of document classification. The main idea is to construct a set of *generalized instances* (GI) to replace the original training examples. Given a particular category, it can be observed that the regularity among positive examples is usually more explicit than that of negative examples. The pattern or classification knowledge induced from a pool of similar positive examples are relatively accurate. On the other hand, negative examples close to such pool are likely noise (i.e., incorrect negative instances). By selectively substituting appropriate positive and negative examples in the positive example pool, we can essentially remove some noisy examples. Based on this idea, we propose the GIS algorithm which focuses on refining the original instances and constructs a set of generalized instances. These generalized instances can remove some of the noisy documents and non-relevant attributes from the training instances. However, this algorithm does not attempt to handle all kind of noise and there is a limit for the algorithm for dealing with noise. The outline of the GIS algorithm is given in Figure 6.4. It automatically selects a representative positive instance and performs a generalization, via the

Input: The training set T

The category C

Procedure GIS(T,C)

1) Let G and G' be generalized instances.
2) GS be generalized instance set, and GS is initialized to empty.
3) Repeat
4) Select a positive instance as G.
5) Rank instances in T according to the similarity metric with G.
6) Compute $Rep(G)$.
7) Repeat
8) $G' := G$.
9) $G := Generalize(G', k)$.
10) Rank instances in T according to the similarity metric with G.
11) Compute $Rep(G)$.
12) Until $Rep(G) < Rep(G')$
13) Add G' to GS.
14) Remove top k instances from T.
15) Until no positive instances in T.
16) Return GS.

Figure 6.4: The Generalized Instance Set (GIS) algorithm

function *Generalize* in Step 9, using k nearest neighbors which may include positive and negative instances. A generalized instance G is formed after the generalization process. This G will be evaluated by the function *Rep* denoting the representative power. If the representative power of G is better than the old one (i.e. G'), we use G as a new point and repeat the search and generalization task again. The algorithm will continue to search for the best local generalized instance as illustrated from Step 7 to Step 12 in the algorithm. If there is no further improvement in terms of the representative power, the last generalized instance G' is added to the generalized instance set GS and the corresponding k nearest neighbors are removed from the training document collection. This process is repeated until no positive instance remains in the training document collection. As the learning progresses, it constructs a number of generalized instances and stores them in GS. Figure 6.5 illustrates the GIS algorithm by means of a network representation.

The representative power function $Rep(G)$ for a generalized instance G is defined as follows:

$$Rep(G) = \sum_{I^+ \in K} (k - rank(I^+))$$

where K is the set of k nearest neighbors of G, I^+ is a positive instance in K and $rank(I^+)$ denotes the ranking of the instance I^+ in the set K according to the similarity metric. Large value for $Rep(G)$ implies that more positive instances are found in the set of k nearest neighbors of G.

A variety of methods can be used for the generalization task in Step 9. We have tried two methods based on the Rocchio and Widrow-Hoff (WH)

algorithms. Let P_k and N_k be the set of positive and negative instances in k nearest neighbors of G respectively. The generalization process based on the Rocchio algorithm is given as follows:

$$G' = \frac{\sum_{I \in P_k} I}{|P_k|} - \eta \frac{\sum_{I \in N_k} I}{|N_k|}$$

where η is the parameter that adjusts the relative impact of positive and negative neighboring instances. The summation is taken as the vector addition. $|S|$ denotes the cardinality of the set S.

The generalization function based on the WH algorithm processes each instance in k nearest neighbors of G one by one in each iteration. Let G_i denote the intermediate value of the generalized instance at the i-th iteration. Initially, the elements in the generalized instance is set to all zeros denoted by $G_0 = \overrightarrow{0}$. At each iteration, G_{i+1} is computed from G_i and the current instance I_i.

$$G_{i+1} = G_i - 2\eta(G_i \cdot I_i - L_i)I_i$$

$$G' = G_{k+1}$$

where $\eta > 0$ is a parameter that controls how quickly G_i is allowed to change. L_i is the class label of the instance I_i. L_i is 1 if I_i is a positive instance and 0 if I_i is a negative instance. The final generalized instance G_{k+1} is the required result for G'.

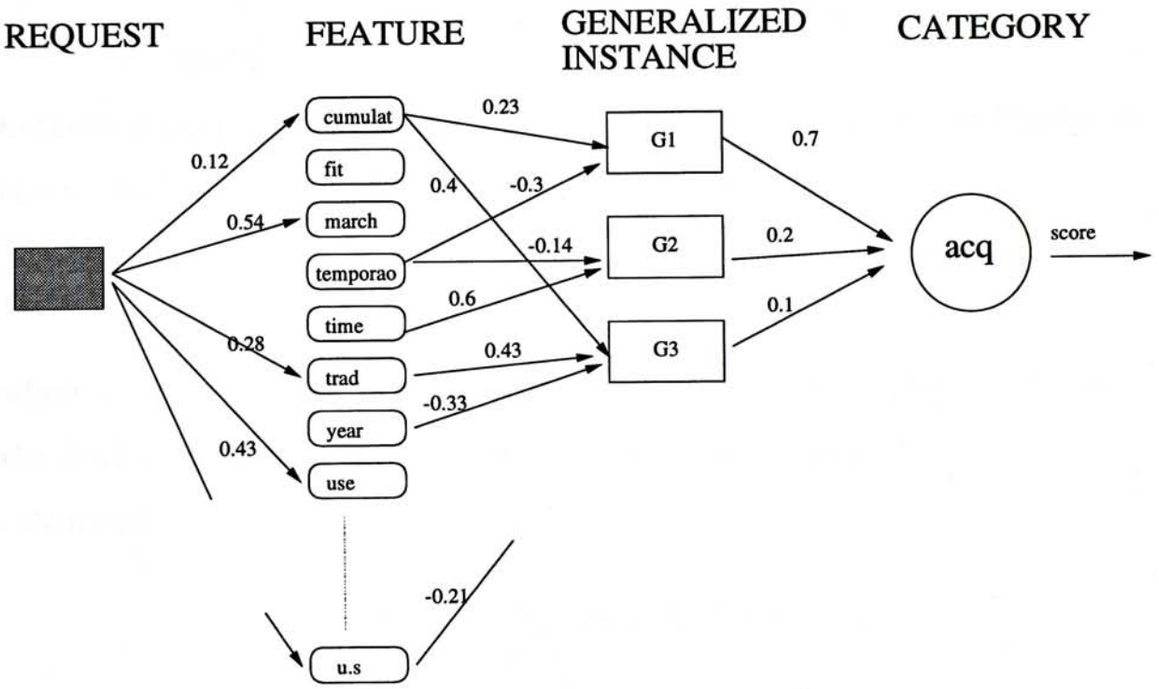

Figure 6.5: The Generalized Instance Set (GIS) algorithm in network representation

6.4 Using GIS Classifiers for Classification

The GIS algorithm learns from the training document collection and produces a classifier represented by a set of generalized instances. The classifier can then be used for classification by computing a score of a request document X. This score is computed as the weighted sum of the similarity metric of each generalized instance G. We define $Assoc(G, C)$ as the association factor between the generalized instance G and the category C. This association factor can be easily calculated during the learning phase as follows:

$$Assoc(G, C) = \frac{|P_k|}{P}$$

where P is the number of positive instances in the training set. As a result, the final score denoted by the function $Score$ for the request document X is calculated by:

$$Score(X, C) = \sum_{G \in GS} \Delta(G, X) Assoc(G, C)$$

If this score is greater than a threshold value θ, the category C is assigned to document X.

The parameters of the GIS algorithm such as k, η and θ of each category can be determined manually or automatically from training document collection. To determine the parameters automatically, we first search for different values of k. We start with a large value of k. For a particular k, each generalized instance is evaluated separately. The global accuracy of the final generalized instance set is evaluated again. k is decreased until the global accuracy of the

generalized instance set decreases. Then k is fixed and we use similar method to search for η and θ.

6.5 Time Complexity

The time complexity of linear classifiers, k-NN and GIS are compared in Big-O notation. Let n_1 and n_2 represent the number of instances in the training set and testing set. The time for computing distance between two instances and time for generalizing two instances are assumed to be constant.

In general, the document categorization task consists of the training phase and the testing phase. Classifiers are learned in the training phase and used to classify the testing instances in the testing phase.

For linear classifier algorithms, time for constructing the generalized instance, classifier, for a category is $O(n_1)$ since generalization between two documents are assumed to be constant. In the testing phase, the classifier takes $O(n_2)$ time to classify the n_2 testing instances. Therefore, the total time of classifying the instances of a category is $O(n_1 + n_2)$.

For k-NN algorithms, no training of classifier is necessary. For a given testing instance, computing the similarity scores and selecting the top k ranking instances take $O(n_1 + n_1 \log n_1)$ time. There are total n_2 testing instances in the testing set. Therefore, the total time for k-NN to classify the instances in the testing set is $O(n_2) \cdot O(n_1 + n_1 \log n_1) = O(n_1 n_2 \log n_1)$.

For the GIS algorithms, the critical parts are computing similarity scores

and ranking n_1 instances in Step 5 and Step 10. These steps take $O(n_1 \log n_1)$ time, similar to k-NN. The time complexity for computing the representative power in Steps 6 and Step 11, and for generalizing top k instances in Step 9 take $O(k)$. Typically k is a small constant. Step 10 is enclosed by the "repeat" cycle in Step 7. Experimental results show that this "repeat" cycle is independent of the number of training instances, and usually less then 10. Therefore, the time complexity for this "repeat" cycle can be reasonably assumed to be a constant. For the "repeat" cycle in Step 3, the worst case is P, where P is the number of positive training instances. However, the average case is likely to be substantially smaller than the worst case. The worst case will only happen if k is set to 1. On average, this "repeat" cycle executes P/k, which is a small constant (e.g. between 1-5). Therefore, the time complexity of the GIS algorithm for a category is $O(n_1 \log n_1)$. Similar to linear classifiers, time for classifying n_2 testing instances is $O(n_2)$. The total time is therefore $O(n_1 \log n_1) + O(n_2) = O(n_1 \log n_1)$.

The time complexity of GIS in classifying the testing instances of a category is $O(n_1 \log n_1)$ which is between the time complexity of linear classifiers $O(n_1)$ and k-NN $O(n_1 n_2 \log n_1)$. Besides, the space of storing the weight vectors of linear classifiers and generalized instances of GIS are substantially smaller than that of k-NN, in which all raw instances have to be kept.

To perform classification of a request document X, k-NN classifiers need to compute n_1 similarity scores. However, the GIS algorithm only requires computation of g similarity score where g is the total number of the learned

generalized instances. Typically, g is much less than n_1. Therefore, the GIS algorithm can be faster than the k-NN algorithm during online classification.

The online classification speed of GIS can be further accelerated by combining all the generalized instances before performing the online classification. Since the request document X and the generalized instances GI's are all normalized. The cosine similarity function can be viewed as a dot product:

$$\begin{aligned}
\Delta(X, D_j) &= \frac{\sum_{i=1}^{n} x_i a_{ij}}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} a_{ij}^2}} \\
&= \sum_{i=1}^{n} x_i a_{ij} \\
&= X \cdot D_j
\end{aligned} \tag{6.1}$$

The *Score* function can then be rewritten as:

$$\begin{aligned}
Score(X, C) &= \sum_{G \in GS} \Delta(G, X) Assoc(G, C) \\
&= \sum_{G \in GS} Assoc(G, C) G \cdot X \\
&= \left(\sum_{G \in GS} Assoc(G, C) G \right) X \\
&= \overline{G} \cdot X \\
&where \quad \overline{G} = \sum_{G \in GS} Assoc(G, C) G
\end{aligned} \tag{6.2}$$

The combined generalized instance $\overline{G}$ of a category can be computed before online classification. Therefore, each request document needs to compute the similarity score with $\overline{G}$ only. Assume all features are used in learning the classifiers. The number of non-zero feature values of the combined generalized instance in the GIS algorithm is much less than that of the feature weight vector of a linear classifier. It is because the feature weight vector of a linear classifier combines all the training instances in the training collection, but the generalized instances of GIS only selectively combine the top k nearest training instances. This is one of the advantages of GIS that the irrelevant features are filtered out automatically. Therefore, the online classification speed of GIS can be faster than that of linear classifiers.

Chapter 7

GIS Experiments

We have implemented our GIS algorithm, the ExpNet k-NN algorithm, the basic Rocchio algorithm and the basic Widrow-Hoff (WH) algorithm. Extensive experiments have been conducted on two large-scale document corpora, namely the OHSUMED collection and the Reuters-21578 collection. The results show that our GIS approach outperforms other approaches used in the experiments.

7.1 Experimental Setup

We have conducted experiments on two commonly used document corpora in document classification research, namely the OHSUMED collection and the Reuters-21578 collection. The Reuters-21578 collection has been introduced in Chapter 3 and Chapter 5. The OHSUMED collection is a bibliographical doc-

ument collection developed by Hersh and his colleagues at the Oregon Health Sciences University. We used 50,216 documents in 1991 which have abstracts. There are total 14,626 distinct main headings appeared in the OHSUMED records. In our research, we chose the set of 119 MeSH categories from the heart disease categories. These 119 MeSH heart disease categories was extracted by Yang from the April 1994 (5th Ed.) UMLS CD-ROM, distributed by the National Library of Medicine. The OHSUMED corpus is difficult to learn for a good classifier since the documents are very noisy.

In order to demonstrate the importance of considering negative instances in the process of generalization, we also implemented a variant of our GIS algorithm in which the *Generalize* function in Step 9 of the GIS algorithm only accepts positive instances and ignores negative instances. We use GIS-RP and GIS-WP to denote the GIS variants incorporated with Rocchio and WH respectively used in the generalization process.

To measure the performance, two common evaluation metrics are used, namely the averaged F_1 measures (AFM) as well as and the micro-averaged recall and precision break-even point measures (MBE) [22]. The averaged F_1 measures have been discussed in Chapter 5. In MBE, no tuning set is needed in searching the threshold score. For each category, the testing documents are ranked by their similarity with the learned classifier. Then, a threshold is selected among these ranked testing documents so that the recall and precision are the same. After that, the total number of false positive, false negative, true positive, and true negative are computed across all categories. These totals

are used to compute the micro-recall and micro-precision. Then we use the interpolation to find the break-even point where the micro-recall and micro-precision are equal.

Different value of parameters have been tried on each algorithm to ensure that the experimental results can reflect the best performance. For the OHSUMED corpus, the values of η tried for the Rocchio algorithm include 0.0, 0.5, 0.65, 0.75, 1.0; the values of k tried for the k-NN algorithm include 50, 100, 150, 200, 300, 500; and we follow the set up of the WH algorithm in [23] using $\eta = 1/(4d^2)$ where d is the maximum value of $\sqrt{\sum d_i^2}$ in the training set. We used $d = 1$ since all the documents have been normalized by the cosine normalization. Therefore, $\eta = 0.25$ for WH algorithm. For the Reuters-21578 corpus, the values of η tried for the basic Rocchio algorithm and the basic WH algorithm are the same as that in the OHSUMED corpus; the values of k tried for the k-NN algorithm in this corpus are 30, 50, 70, 100, 150, 200. Then, we chose the parameter with the best performance to represent the performance of each algorithm. For GIS algorithm, dynamic parameter searching technique described in Section 6.4 is used for searching through the parameter settings used in Rocchio, WH and k-NN algorithms.

In addition to the single train-and-test experiment setup, we also try n-fold cross-validation setup [35]. The single train-and-test setup is to evaluate the performance of the methods by splitting the document collections into a training collection and a testing collection according to the time-stamp of the documents. That is, documents with time-stamp before a certain date are

assigned to the training document collection, while the remaining are assigned to the testing document collection. In Reuters-21578 collection, we divided the documents into 9,603 training documents and 3,299 testing documents. In OHSUMED collection, the first 33,478 documents were used as the training document collection and the remaining 16,738 documents were used for testing. In micro-averaged recall and precision break-even point measure (MBE), we chose those categories that appear in at least one training document and one testing document. There are 90 categories in the Reuters-21578 corpus and 84 categories in OHSUMED corpus. In averaged F_1 measures (AFM), the training documents are further divided into a growing set and a tuning set. The growing collection is used to construct the classifiers while the tuning document collection is used to search automatically the optimal threshold for each category. We chose those categories having at least one positive document in the growing document collection, the tuning document collection and the testing document collection. There are 66 categories in Reuters-21578 corpus and 60 categories in OHSUMED corpus. Therefore, the result in Reuters-21578 corpus cannot directly compare with the result in Chapter 5 in which 90 categories are selected and no tuning set is needed.

For n-fold cross-validation setup, the document collection is divided into n partitions. A classifier is learned using (n-1) partitions and tested on the single remaining portion. This process is repeated n times. The overall performance is the average of all n trials. In our experiments, n is set to five. That is, the documents in each collection are divided into five equal portions according to

the time stamp of the documents. There are 2580 documents and 10043 documents in each portion of Reuters-21578 document collection and OHSUMED document collection respectively. For each fold, there are 10320 training documents and 2580 testing documents in Reuters-21578 corpus. There are 40172 training documents and 10043 testing documents in OHSUMED corpus. The categories are chosen provided that at least one set of splitting satisfies the conditions in the single train-and-test setup. For micro-averaged recall and precision break-even point measure (C-MBE), there are 97 categories in the Reuters-21578 corpus and 88 categories in the OHSUMED corpus. For averaged F_1 measure (C-AFM), there are 70 categories in the Reuter-21578 corpus and 71 categories in the OHSUMED corpus. The number of category for MBE and AFM are different since the training collection of AFM is further divided into a growing collection and a tuning collection. The growing collection is used to construct the classifier while the tuning collection is used to search the threshold score of the learned classifier. Only those categories with at least one positive growing document ,one tuning document and one testing document in the collection are selected. Therefore, categories selected in AFM are always smaller than that of MBE.

Typically, the n-fold cross-validation gives a more reliable result since it averages the results from each fold. For automatic document categorization problem, we commonly adopt the single train-and-test splitting method so that the results can be compared across previous work done by the researchers. As a result, for each method, four set of experiments were conducted for each

corpus. They are averaged F_1 measure (AFM), micro-averaged recall and precision break-even point measure (MBE), cross-validation of averaged F_1 measure (C-AFM) and cross-validation of micro-averaged recall and precision break-even point measure (C-MBE).

7.2 Results

Figure 7.1, Table 7.1 to Table 7.10 show the experimental results on the Reuters-21578 corpus and the parameter values corresponding to the best result of each algorithm. Figure 7.1 depicts the micro-averaged break-even point measures of each algorithm on all 90 categories in the Reuters-21578 corpus. It shows that both GIS-R and GIS-W perform much better than the basic linear classifiers including Rocchio and WH. Our GIS-based algorithms achieve better performance than the k-NN algorithm although k-NN is better than Rocchio classifier. Table 7.1 summarizes the performance of all categories in Reuters-21578 corpus. Using the C-AFM metric, GIS-W algorithms has 15.8% improvement over Rocchio, 10.8% improvement over WH and 21.3% improvement over k-NN. Table 7.2 to Table 7.5 summarize the micro-averaged break-even point measures and F_1 measures of the ten most frequent categories in the Reuters-21578 corpus. The results on both corpora clearly demonstrate that our GIS-based algorithms, in general, achieve better performance than other approaches used in the experiments.

Table 7.8, Table 7.9 and Table 7.10 summarize the computational time of

each algorithm in Reuters-21578 corpus. Table 7.8 shows the average computational time of all categories in Reuters-21578 corpus. The average training time of GIS-R is larger than Rocchio. This is due to the ranking of instances in Step 10. However, the training time of GIS-W is much less than WH. This is because the computational time of multiplication in WH is very time consuming. The results also show that the online classification time of GIS is much smaller than that of linear classifiers and k-NN. Table 7.9 shows the average computational time of the ten most frequent categories in Reuters-21578. Table 7.10 shows the average computational time of the low frequent categories, (the number of positive training instances is less than or equal to 20), in the Reuters-21578 corpus. These two tables show that the less the number of positive training instances, the less the training time and online classification time are needed for GIS. This is because less "repeat" cycles are needed in Step 7 and less number of non-zero features in the generalized instances of GIS.

Figure 7.2 and Table 7.11 to Table 7.17 show the experimental results for the OHSUMED corpus and the parameter values corresponding to the best result of each algorithm. Figure 7.2 depicts the micro-averaged break-even point of each algorithm on all 84 categories in the OHSUMED corpus. It shows that both GIS-R and GIS-W perform much better than the basic linear classifiers including Rocchio and WH. The GIS algorithms achieve better performance than the k-NN algorithm although k-NN is better than Rocchio classifier. Table 7.12 to Table 7.15 summarize the micro-averaged break-even point measures and F_1 measures of the ten most frequent categories in the

OHSUMED corpus. Table 7.11 summarizes the performance of all categories in OHSUMED corpus. The last three columns show the improvement of GIS over the other three algorithms respectively. Using the C-AFM metric, GIS-R algorithms has 15.5% improvement over Rocchio, 26.6% improvement over WH and 12.8% improvement over k-NN.

Table 7.6, Table 7.7, Table 7.16 and Table 7.17 show the experimental results of the GIS and its variants on the Reuters-21578 and OHSUMED corpora. The results on both corpora consistently show that the GIS-R or GIS-W algorithm perform much better than GIS-RP and GIS-WP algorithm. These results indicate the importance of considering negative instances during the generalization process.

More detailed results of experiments on GIS are listed in the Appendix A and Appendix B. Appendix A includes the performance of all categories on different evaluation metrics including microaveraged break-even point measures of the Reuters-21578 and the OHSUMED corpora, F_1 measures of the Reuters-21578 and the OHSUMED corpora, cross-validation microaveraged break-even point measures of the Reuters-21578 and the OHSUMED corpora and cross-validation F_1 measures of the Reuters-21578 and the OHSUMED corpora. Appendix B includes the computational time of all categories of Rocchio, WH, k-NN and GIS on the Reuters-21578 corpus.

	Rocchio	WH	k-NN	GIS-R	GIS-W	Improvement(%)		
MBE(90)	0.781	0.822	0.820	0.842	**0.860**	10.1	4.6	4.9
AFM(66)	0.516	0.543	0.529	0.572	**0.584**	13.2	7.6	9.4
C-MBE(97)	0.765	0.787	0.793	0.820	**0.843**	10.2	7.1	6.3
C-AFM(70)	0.486	0.508	0.464	0.533	**0.563**	15.8	10.8	21.3

Table 7.1: Performance of all categories in the Reuters-21578 corpus. The last three columns show the percentage of improvement of GIS over Rocchio, WH and k-NN respectively.

Category	Rocchio	WH	k-NN	GIS-R	GIS-W
acq	0.827	0.902	0.875	**0.930**	0.923
corn	0.614	**0.850**	0.700	0.832	**0.850**
crude	0.795	0.839	0.818	0.837	**0.850**
earn	0.957	0.954	0.963	0.972	**0.977**
grain	0.803	0.900	0.816	0.829	**0.913**
interest	0.697	0.662	0.707	0.738	**0.750**
money-fx	0.585	0.680	0.652	0.708	**0.761**
ship	0.804	0.816	0.793	0.827	**0.883**
trade	0.732	0.723	0.754	0.766	**0.771**
wheat	0.713	**0.839**	0.713	0.811	**0.839**
Average	0.753	0.812	0.779	0.825	**0.852**

Table 7.2: Recall and precision break-even point measures of the ten most frequent categories in the Reuters-21578 corpus. The parameters, which correspond to the best results are: $\eta = 1$ for Rocchio, $\eta = 1/4$ for WH, $k = 70$ for k-NN.

Category	Rocchio	WH	k-NN	GIS-R	GIS-W
acq	0.757	0.771	0.819	**0.887**	0.874
corn	0.639	0.808	0.717	0.800	**0.813**
crude	0.774	0.816	0.792	0.807	**0.841**
earn	0.927	0.891	0.944	0.942	**0.964**
grain	0.751	0.899	0.799	0.836	**0.925**
interest	0.663	0.641	0.671	0.661	**0.706**
money-fx	0.688	0.707	0.685	0.739	**0.772**
ship	0.699	**0.784**	0.71	0.707	0.778
trade	0.704	0.674	0.735	0.731	**0.754**
wheat	0.731	0.847	0.742	0.829	**0.885**
Average	0.733	0.784	0.761	0.794	**0.831**

Table 7.3: Cross-Validation recall and precision break-even points for ten most frequent categories in Reuter-21578 corpus. The parameters, which correspond to the best results are: $\eta = 1$ for Rocchio, $\eta = 1/4$ for WH, $k = 100$ for k-NN.

Category	Rocchio	WH	k-NN	GIS-R	GIS-W
acq	0.856	0.844	0.894	**0.919**	0.874
corn	0.600	0.894	0.735	**0.923**	0.863
crude	0.850	0.789	0.799	**0.859**	0.776
earn	0.964	0.952	0.965	0.962	**0.987**
grain	0.722	0.877	0.800	0.790	**0.914**
interest	0.703	0.667	**0.759**	0.748	0.725
money-fx	0.369	0.608	0.664	0.676	**0.768**
ship	0.763	0.330	0.816	**0.827**	0.519
trade	0.736	0.691	0.760	**0.777**	0.772
wheat	0.696	0.818	0.680	0.812	**0.853**
Average	0.726	0.747	0.787	**0.829**	0.805

Table 7.4: F_1 measures of the ten most frequent categories in the Reuters-21578 corpus. The parameters, which correspond to the best results are: $\eta = 1$ for Rocchio, $\eta = 1/4$ for WH, $k = 100$ for k-NN.

Category	Rocchio	WH	KNN	GIS-R	GIS-W
acq	0.749	0.705	0.816	0.882	**0.895**
corn	0.623	0.802	0.685	0.787	**0.838**
crude	0.746	0.782	0.778	**0.807**	0.799
earn	0.925	0.848	0.940	0.946	**0.964**
grain	0.742	0.861	0.759	0.834	**0.926**
interest	0.653	0.561	0.653	0.678	**0.720**
money-fx	0.640	0.660	0.594	**0.739**	0.771
ship	0.654	0.595	0.664	**0.734**	0.625
trade	0.655	0.557	0.704	0.739	**0.758**
wheat	0.710	0.822	0.748	0.838	**0.875**
Average	0.710	0.719	0.734	0.798	**0.817**

Table 7.5: Cross-validation F_1 measures for the ten most frequent categories in Reuters-21578 corpus. The parameters, which correspond to the best results are: $\eta = 1$ for Rocchio, $\eta = 1/4$ for WH, $k = 150$ for k-NN.

	GIS-RP	GIS-WP	GIS-R	GIS-W
MBE	0.712	0.717	0.842	0.860

Table 7.6: Micro-averaged recall and precision break-even point measures of our GIS algorithm and some of its variants for 90 categories in the Reuters-21578 corpus.

Category	GIS-RP	GIS-WP	GIS-R	GIS-W
acq	0.566	0.556	0.930	0.923
corn	0.602	0.690	0.832	0.850
crude	0.758	0.828	0.837	0.850
earn	0.952	0.947	0.972	0.977
grain	0.769	0.829	0.829	0.913
interest	0.662	0.646	0.738	0.750
money-fx	0.529	0.552	0.708	0.761
ship	0.693	0.600	0.827	0.883
trade	0.695	0.672	0.766	0.771
wheat	0.741	0.769	0.811	0.839
Average	0.697	0.709	0.825	0.852

Table 7.7: Recall and precision break-even point measures for the ten most frequent categories of our GIS algorithm and some of its variants for the ten most frequent categories in the Reuters-21578 corpus.

System	Training Time	Test Time	Total Time
Rocchio	2.49	2.45	4.94
WH	14.63	2.48	17.11
k-NN	0.00	489.27	489.27
GIS-W	6.71	0.83	7.54
GIS-R	4.85	0.85	5.70

Table 7.8: Comparison of computational time (in seconds) for the Reuters-21578 corpus.

System (10 most frequent categories)	Training Time	Test Time	Total Time
Rocchio	2.49	2.47	4.96
WH	14.65	2.50	17.15
k-NN	0.00	489.65	489.65
GIS-W	25.05	1.39	26.44
GIS-R	17.78	1.50	19.28

Table 7.9: Average computational time (in seconds) of the 10 most frequent categories in the Reuters-21578 corpus.

System (low frequency)	Training Time	Test Time	Total Time
Rocchio	2.49	2.45	4.93
WH	14.63	2.48	17.10
k-NN	0.00	488.83	488.83
GIS-W	3.44	0.69	4.13
GIS-R	2.53	0.70	3.23

Table 7.10: Average computational time (in seconds) of categories with number of positive training documents less than 20, total 47 categories, in the Reuters-21578 corpus.

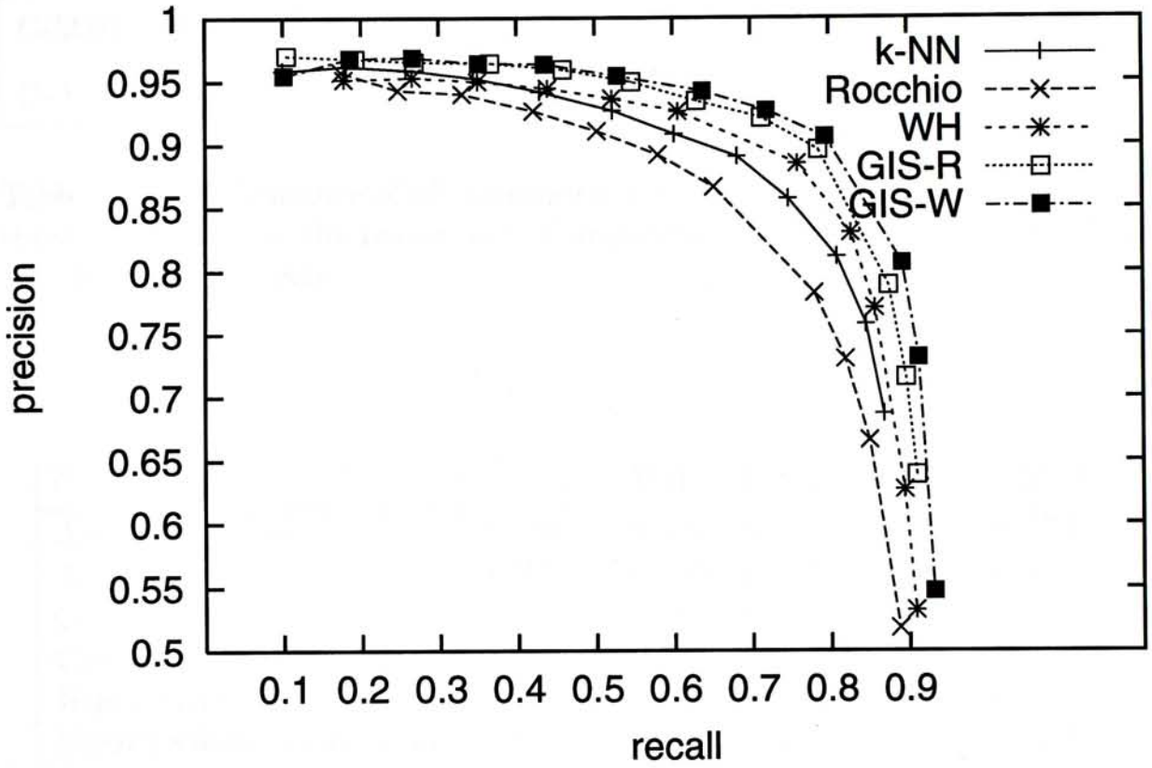

Figure 7.1: Micro-recall/micro-precision performance of 90 categories in the Reuters-21578 corpus.

	Rocchio	WH	k-NN	GIS-R	GIS-W	Improvement(%)		
MBE(84)	0.492	0.519	0.521	**0.572**	0.550	16.3	10.2	9.8
AFM(60)	0.354	0.344	0.358	0.381	**0.395**	11.6	14.8	10.3
C-MBE(88)	0.477	0.502	0.507	0.540	**0.548**	14.9	9.2	8.1
C-AFM(71)	0.297	0.271	0.304	**0.343**	0.341	15.5	26.6	12.8

Table 7.11: Performance of all categories in the OHSUMED corpus. The last three columns show the percentage of improvement of GIS over Rocchio, WH and k-NN respectively.

Category	Rocchio	WH	k-NN	GIS-R	GIS-W
Angina Pectoris	0.323	0.485	0.496	0.543	**0.574**
Arrhythmia	0.475	0.460	0.460	**0.547**	0.443
Coronary Arteriosclerosis	0.218	0.289	0.291	**0.364**	0.327
Coronary Disease	0.523	0.565	0.551	0.579	**0.581**
Heart Arrest	0.641	0.586	0.583	**0.641**	0.563
Heart Defects, Congenital	0.440	0.462	0.484	0.484	**0.527**
Heart Diseases	0.194	0.177	0.194	**0.228**	0.194
Heart Failure, Congestive	0.473	0.558	0.552	0.598	**0.621**
Myocardial Infraction	0.737	0.762	0.759	**0.810**	0.806
Tachycardia	0.608	0.634	**0.673**	**0.673**	0.653
Average	0.463	0.497	0.504	**0.547**	0.529

Table 7.12: Recall and precision break-even point measures of the ten most frequent categories in the OHSUMED corpus. The parameters, which correspond to the best results are: $\eta = 1$ for Rocchio, $\eta = 1/4$ for WH, $k = 100$ for k-NN.

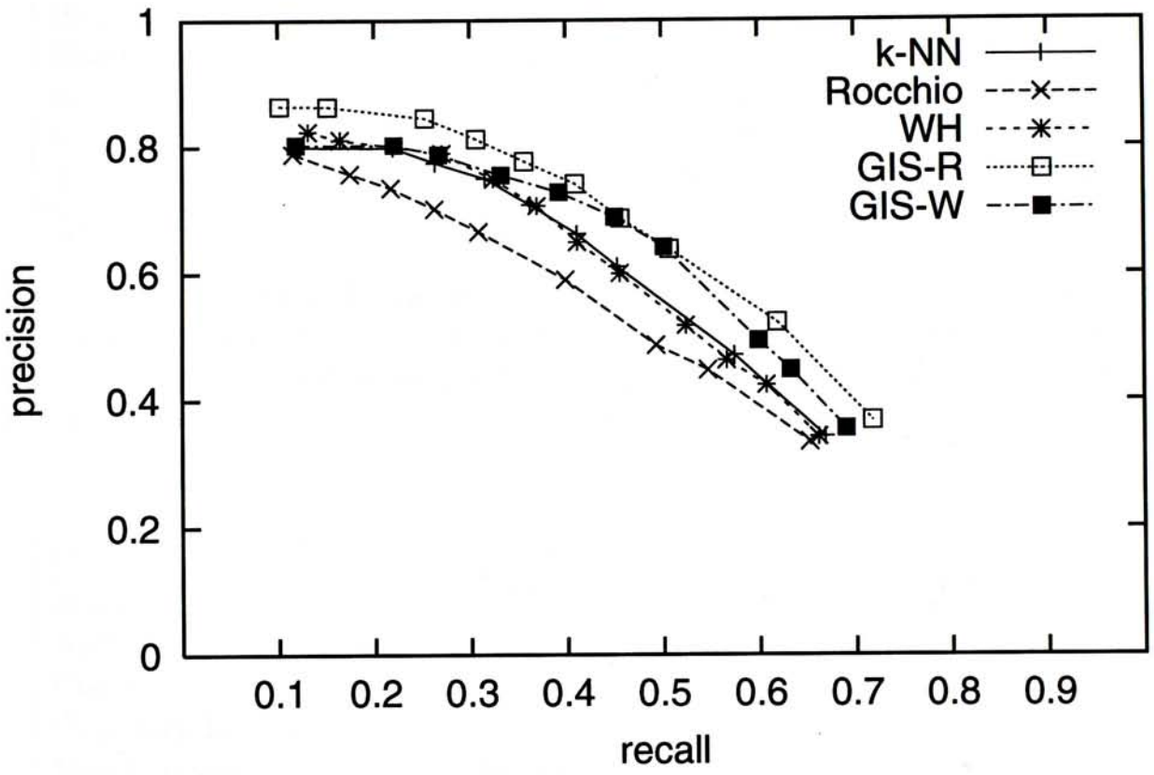

Figure 7.2: Micro-recall/micro-precision performance of 84 categories in the OHSUMED corpus.

Category	Rocchio	WH	k-NN	GIS-R	GIS-W
Angina Pectoris	0.383	0.519	0.452	0.494	**0.585**
Arrhythmia	0.405	**0.504**	0.407	**0.504**	0.491
Coronary Arteriosclerosis	0.263	0.329	**0.438**	0.427	0.426
Coronary Disease	0.543	0.554	0.580	0.599	**0.603**
Heart Arrest	0.611	0.626	0.587	0.621	**0.632**
Heart Defects, Congenital	0.465	0.501	0.450	0.498	**0.503**
Heart Diseases	0.144	0.116	0.161	**0.232**	0.200
Heart Failure, Congestive	0.523	0.613	0.583	0.622	**0.639**
Myocardial Infarction	0.645	0.673	0.660	0.704	**0.710**
Tachycardia/	0.589	0.629	0.667	0.662	**0.679**
Average	0.457	0.506	0.499	0.536	**0.547**

Table 7.13: Cross-validation recall and precision break-even points for the ten most frequent categories in OHSUMED corpus. The parameters, which correspond to the best results are: $\eta = 0.75$ for Rocchio, $\eta = 1/4$ for WH, $k = 100$ for k-NN.

Category	Rocchio	WH	k-NN	GIS-R	GIS-W
Angina Pectoris	0.345	0.442	0.458	**0.539**	0.538
Arrhythmia	0.516	0.417	0.521	**0.654**	0.581
Coronary Arteriosclerosis	0.304	0.342	0.133	0.258	**0.522**
Coronary Disease	0.487	0.475	0.530	0.564	**0.571**
Heart Arrest	**0.633**	0.595	0.480	0.588	0.557
Heart Defects, Congenital	0.364	**0.583**	0.500	0.542	0.577
Heart Diseases	**0.184**	0.180	0.049	0.174	0.049
Heart Failure, Congestive	0.349	0.519	0.487	**0.569**	0.525
Myocardial Infarction	0.759	0.764	0.772	0.785	**0.812**
Tachycardia	**0.667**	0.486	0.659	0.559	0.529
Average	0.461	0.480	0.459	0.523	**0.526**

Table 7.14: F_1 measures of the ten most frequent categories in the OHSUMED corpus. The parameters, which correspond to the best results are: $\eta = 1$ for Rocchio, $\eta = 1/4$ for WH, $k = 100$ for k-NN.

Category	Rocchio	WH	k-NN	GIS-R	GIS-W
Angina Pectoris	0.300	**0.454**	0.401	0.445	0.418
Arrhythmia	0.335	0.424	0.387	0.425	**0.484**
Coronary Arteriosclerosis	0.225	0.243	0.271	0.341	**0.422**
Coronary Disease	0.538	0.551	0.566	**0.600**	0.587
Heart Arrest	0.570	**0.600**	0.546	0.595	0.568
Heart Defects, Congenital	0.360	0.462	0.449	0.456	**0.478**
Heart Diseases	0.120	0.059	0.109	**0.138**	0.071
Heart Failure, Congestive	0.506	0.544	0.559	0.600	**0.619**
Myocardial Infarction	0.604	0.604	0.622	**0.650**	0.643
Tachycardia	0.511	0.499	0.528	**0.639**	0.611
Average	0.407	0.444	0.444	0.489	**0.490**

Table 7.15: Cross-validation F_1 measures for the ten most frequent categories in OHSUMED corpus. The parameters, which correspond to the best results are: $\eta = 75$ for Rocchio, $\eta = 1/4$ for WH, $k = 200$ for k-NN.

	GIS-RP	GIS-WP	GIS-R	GIS-W
MBE	0.473	0.477	0.572	0.550

Table 7.16: Micro-averaged recall and precision break-even point measures of our GIS algorithm and some of its variants for 84 categories in the OHSUMED corpus

Category	GIS-RP	GIS-WP	GIS-R	GIS-W
Angina Pectoris	0.310	0.341	0.543	0.574
Arrhythmia	0.446	0.446	0.547	0.443
Coronary Arteriosclerosis	0.218	0.218	0.364	0.327
Coronary Disease	0.526	0.512	0.579	0.581
Heart Arrest	0.602	0.602	0.641	0.563
Heart Defects, Congenital	0.435	0.484	0.484	0.527
Heart Diseases	0.130	0.114	0.228	0.194
Heart Failure, Congestive	0.453	0.431	0.598	0.621
Myocardial Infarction	0.711	0.690	0.810	0.806
Tachycardia	0.554	0.535	0.673	0.653
Average	0.439	0.437	0.547	0.529

Table 7.17: Recall and precision break-even point measures of our GIS algorithm and some of its variants for the ten most frequent categories in the OHSUMED corpus.

Chapter 8

A New Information Filtering Approach Based on GIS

In this chapter, we discuss information filtering (IF) systems in more details. A widely used similarity-based IF technique is described. We then introduce a new IF approach based on GIS, a newly developed method.

8.1 Information Filtering Systems

Filtering of information occurs in our daily lives. For example, we only buy certain newspapers since other newspapers may contain information that is redundant or irrelevant to our interests. In this way, we have filtered out some of the large amount of information to which we have access. Even within a newspaper, we also choose articles that appear relevant to our interests. With

the advent of electronic presentation of information, some manual filtering tasks can be eliminated by an information filtering (IF) system.

One of the earliest electronic document filtering came from the Selective Dissemination of Information (SDI). It was designed to keep scientists informed of new documents published in their areas of specialization. The scientists could create and modify a user profile of keywords that described his or her interests. The system then used the profile to match the keywords against new documents in order to predict which new documents would be relevant to the scientist's interests. Several other approaches have been developed such as the systems discussed in Chapter 2 [38, 28, 16, 20].

Techniques for automatic document categorization can sometimes be applied for information filtering with some modifications, and vice versa. Some examples are the ExpNet developed by Yang [41] and the Linear Text Classifier developed by Lewis [23].

There are many common characteristics for automatic document categorization and information filtering problems. Both of them mainly deal with documents with a set of category labels or user relevance judgments. In general, a document in automatic document categorization system may be compared to a number of category labels at once and the relevant category labels are assigned to the document. A category label here can be viewed as a user with interest of a particular topic in information filtering problem. The learned classifier can act as filter to decide whether or not accept a new incoming document. This is similar to binary classification.

However, some important characteristics distinguish information filtering systems from document classification systems. Both filtering systems and classification systems have a set of internal specifications used to make judgement on the relevance of new documents. The internal specifications of information filtering systems are called profiles, which are typically structures representing long-term needs or topics. For document categorization, the internal specifications are called classification schemes. Furthermore, in filtering systems, the speed of filtering is more critical because large number of new documents may need to be processed in real time.

Several existing information filtering approaches have been discussed in Chapter 2. All of them require a user to specify his or her interests by a user profile. This profile is not easy to create manually. This is a major weakness of these approaches. In our IF investigation, we use sample documents that have been judged to be relevant or not relevant to learn the topic profile automatically. There is no need for a user to worry about what keywords and their corresponding weights should be put into the topic profile.

In fact, many text document collections contain the relevance judgments specifying the set of documents relevant or not relevant to certain topics. For example, the documents corpus, Foreign Broadcast Information Service (FBIS), contains translated text documents or transcripts from various non-American broadcast and print publications [11]. This corpus also comes with a set of topics expressing various information needs. An example of a topic is like: "A relevant document should describe non-commercial satellite launches".

Associated with each topic, there is a set of sample documents which have been judged as relevant or not relevant to the topic. The aim of information filtering is to construct for each topic a topic profile or a filtering function which is able to make a binary decision to either accept or reject each new document as it arrives.

8.2 GIS-Based Information Filtering

Similar to automatic document categorization, the document and the user profile are mapped into an internal representation. In the similarity-based information filtering approach, the similarity of the document and the user profile is measured by a similarity function. The IF system determines whether the document is relevant to the user based on the similarity score generated by the similarity function. Figure 8.1 depicts a system architecture of a similarity-based information filtering system. There are three main components in the system, namely the Topic Profile Learning Module, the Filtering Function, and the Thresholding Module.

The purpose of the Topic Profile Learning Module is to learn the profile of a topic given a set of training document collection. Typically, the training document collection is further divided into two collections, namely the training document collection and the tuning document collection. The training document collection is used to learn the profile of a topic. Different learning algorithms can be used in this Topic Profile Learning Module such as linear

classifier learning algorithms, k-NN learning algorithms and our new GIS algorithm. The learned topic profile will then be used for determining the filtering threshold of the filtering function. The on-line filtering task is done by the Filtering Function which computes the similarity of an incoming document and the learned profile. The Thresholding Module uses the learned topic profile and the tuning document collection to determine the filtering threshold. This filtering threshold together with the similarity score from the Filtering Function are used to determine whether the incoming document is accepted or rejected.

An alternative setup is to use the whole training document collection for both building the filtering profile and threshold selection. However, the scores of relevant training documents will be biased upwards and this bias may be passed on to the selected threshold. We attempt to reduce this bias by splitting the training document collection into two parts. One is for building the filtering profile and the other part is for selecting the threshold.

We propose to use our new GIS approach as the Topic Profile Learning Module. For each topic, a topic profile is learned automatically from the training document collection with relevant judgments. The tuning document collection is then used to select the filtering threshold according to the evaluation utility used in the experiment. To select the filtering threshold, document in the tuning document collection are ranked by the similarity values with the topic profile using the cosine similarity coefficient. Then, we select a filtering threshold that optimizes the utility value of the tuning document collection.

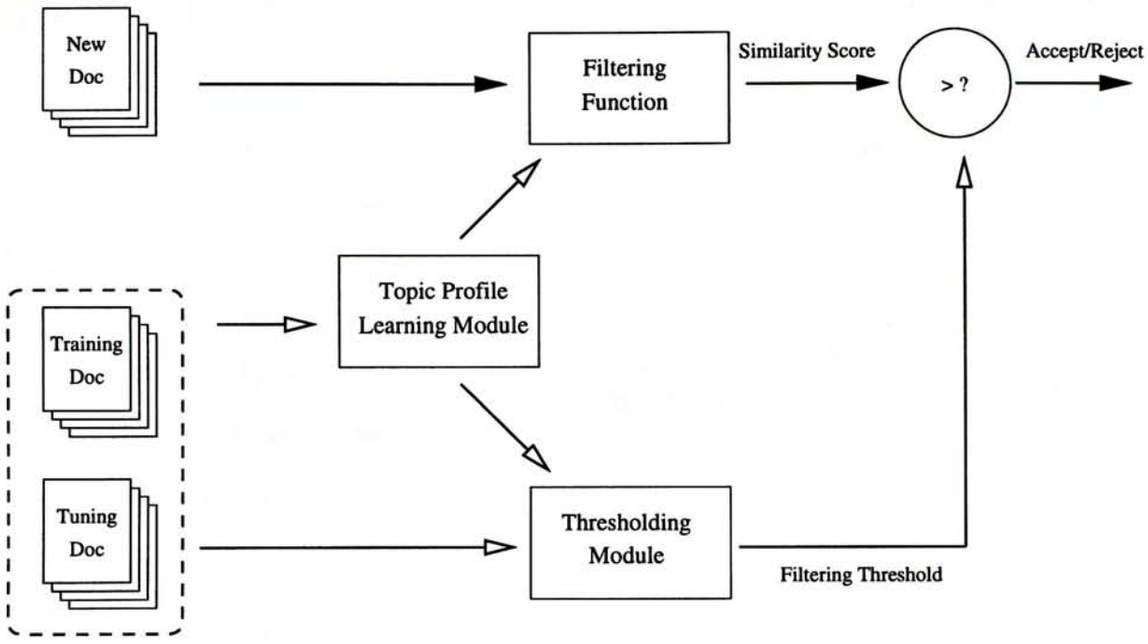

Figure 8.1: A similarity-based information filtering system.

Figure 8.2 depicts the tuning documents ranked by their similarity values with a topic. Each tuple is assumed to contain the document ID., the relevant judgment and the similarity value with the topic profile. The filtering threshold of a certain evaluation score is obtained by searching for a value, for instance 0.603, so that it maximizes the utility score in the tuning set for this topic. Documents in the testing document collection with similarity scores with the topic profile greater than 0.603 are judged as relevant to this topic under the evaluation scheme.

For on-line document filtering, the similarity value of the incoming document and the learned topic profile is measured. An incoming document with similarity value larger than the filtering threshold is considered as relevant to

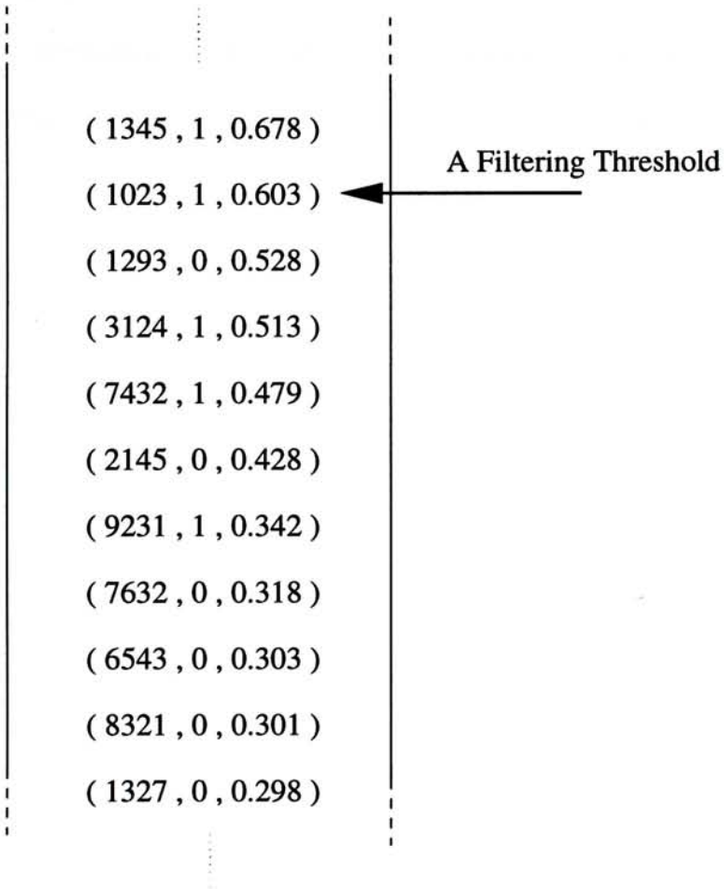

Figure 8.2: A set of tuning document ranked by the similarity score.

the topic, otherwise, it is considered as non-relevant to the topic.

To evaluate our GIS-based IF approach, we have conducted extensive experiments on our GIS approach as well as some existing techniques such as linear classifier algorithms and k-NN algorithms. The next chapter gives a comparison of the filtering performance of these algorithms on a large-scale, real-world document collection.

Chapter 9

Experiments on GIS-based Information Filtering

We have implemented our GIS-based information filtering approach. In order to compare with other approaches, we have also implemented linear classifiers approaches and k-NN approaches. Extensive experiments have been conducted on a large-scale corpus, namely the Foreign Broadcast Information Service (FBIS) collection. The results show that the filtering performance of our GIS approach, in general, outperforms others approaches used in the experiments.

9.1 Experimental Setup

The document corpus used in information filtering experiments is FBIS. It contains translated text documents or transcripts from various non-American

broadcast and print publications [11]. All documents have date stamps attached and have been ordered according to the date. We used 130,000 documents in our experiments. The first 70,000 documents were used as training documents. The remaining 60,000 documents were used as testing documents. In order to have a fair evaluation, the topics with too few relevant documents will not be considered in our experiments. We used 15 topics which have the most relevant documents in our experiments. An example of a topic is:

> A relevant document will discuss a current debt rescheduling agreement reached, proposed, or being negotiated between a debtor developing country and one or more of its creditors, commercial and official. It will identify the debtor country and the creditor(s), the repayment time period requested or granted, the monetary amount requested or covered by the accord, and the interest rate, proposed or set.

The last 10,000 documents in the training documents were used as tuning set to determine the filtering threshold. In other words, only the first 60,000 documents were used in constructing the filtering function.

In a filtering system, an incoming document needs to be determined if it is accepted or rejected. The output of the filtering system is treated as an unordered set of documents. Therefore, the evaluation measures based on a ranked set of documents, such as the precision-recall curve, are not appropriate. Instead, we use two set-based evaluation metrics, namely, utility metrics and Average Set Precision (ASP) similar to the metrics used in Text REtrieval

Conference (TREC) [10].

	Relevant	Not Relevant
Retrieved	R_+	N_+
Not Retrieved	R_-	N_-

Figure 9.1: A contingency table showing the result of a topic

Utility metric assigns a value or cost to each document based on whether it is retrieved or not and whether it is relevant or not. Figure 9.1 shows a contingency table summarizing the filtering result of a topic. The utility metric is defined as:

$$Utility = AR_+ + BN_+ + CR_- + DN_-$$

where the R_+ is the number of documents relevant to the topic and being retrieved, N_+ is the number of documents not relevant to the topic and being retrieved, R_- is the number of documents relevant to the topic and not being retrieved, and N_- is the number of documents not relevant to the topic and not being retrieved. The coefficients A, B, C and D are used to determine the relative value of each possible assignment. The larger the utility score, the better the filtering system is performing for a topic. For our experiments, we used two different settings of the utility coefficients and Average Set Precision (ASP) similar to TREC [10]. The utility metrics are:

$$
\begin{aligned}
U1 &= 3R_+ - 2N_+ \\
U2 &= 3R_+ - N_+ - R_-
\end{aligned}
\tag{9.1}
$$

ASP is defined as the product of precision and recall as follows:

$$
ASP = \left(\frac{R_+}{R_+ + N_+}\right)\left(\frac{R_+}{R_+ + R_-}\right)
$$

In evaluating the performance of a system, the utility score will vary widely from topic to topic, and there is no valid way to normalize them. Therefore, the utility score cannot easily be averaged or compared across topics. For the ASP metric, when there is no relevant documents retrieved, the system returns a score of zero. This means that retrieving no document is equivalent to retrieving an arbitrary number of non-relevant documents.

Utility	S1	S2	S3	S4	Rank	S1	S2	S3	S4
T1	1	1	-18	4	T1	2.5	2.5	1	4
T2	100	123	89	10	T2	3	4	2	1
T3	-12	-100	-1	0	T3	2	1	3	4
					Average	2.5	2.5	2	3

Figure 9.2: An example showing the conversion from utility to ranking score.

In order to compare the performance across topics, simple averaging of the utility score across topics gives each retrieved document equal weight. The

result will be dominated by the topics with large retrieved sets. Therefore, in addition to the above metrics, a comparative evaluation method will be employed based on a conversion from utility scores. This evaluation is based on the ranking of the utility score in each category. The conversion steps are given as follows:

1. For each topic, systems are ranked according to their performance. The higher the utility score, the higher is the ranking score. Systems having the same utility score will have the same ranking score. The ranking scores of these systems are computed as the mean of their lowest and highest ranking scores.

2. Average the ranks by the system over all topics.

Figure 9.2 shows an example of this ranking. Let S1, S2, S3 and S4 be the systems to be compared. T1, T2 and T3 are the topics used in the evaluation. The average ranking score of S1, S2, S3 and S4 across the topics T1, T2 and T3 are 2.5, 2.5, 2 and 3 respectively. S4 has the best performance among these four systems since it has the highest average ranking score. Systems S1 and S2 have the same utility score in topic T1. The ranking of S1, S2, S3 and S4 are 2.5, 2.5, 1 and 4 respectively. Since the lowest and highest ranking scores of S1 and S2 are 2 and 3. Thus, the mean of them is 2.5.

In this comparison, all topics are treated to be equal. The larger the average ranking score, the better is the system performing with respect to its competitors. One advantage of this comparative evaluation is that it provides

a global comparative evaluation in situations where it would be difficult when using utility scores. The average ranking scores generated by the same set of systems are directly comparable. Besides, the average ranking score depends on the systems being compared. Adding or removing a system will change the scores.

Similar to the last two experiments, different value of parameters have been tried on each algorithm to ensure that the experimental results can reflect the best performance. The values of η tried for Rocchio algorithm include 0.0, 0.5, 0.65, 0.75, 1.0. The value tried for WH algorithm is $\eta = 1/(4d^2)$ where $d = 1$ since all the documents have been normalized by the cosine normalization. The values of k tried for the k-NN algorithm include 50, 100, 200, 300, 500, 800, 1000. Then, we chose the parameter with the best filtering performance to compare different algorithms. The threshold is determined by optimizing the performance, namely the highest utility score or ASP score, on the tuning document collection.

9.2 Results

Table 9.1 to Table 9.6 show the utility scores and ranking scores of 15 topics of FBIS using different algorithms. Table 9.1 shows the ASP score of the 15 topics and their average. Table 9.2 shows that GIS gets the best average ranking of ASP with 9 topics out of the 15 topics. Table 9.3 shows the U1 utility of the 15 topics and their averages. Table 9.4 shows that GIS gets the

best average ranking of U1 utility with 10 topics out of the 15 topics. Table 9.5 shows the U2 utility of the 15 topics. Table 9.6 shows that GIS gets the best average ranking of U2 utility with 9 topics out of the 15 topics. The results demonstrate that our GIS algorithm, in general, achieves better performance than other approaches used in the experiments.

Table 9.4 and Table 9.6 use the utility scores to evaluate the performance. Most approaches of the average U2 values are negative while GIS achieves positive scores. The difference between U1 and U2 is that U2 does not penalize those documents belonging to the topic but not being retrieved. Most approaches perform quite well under U1 utility score, but most approaches have negative average U2 utility value. Approaches fail to retrieve relevant documents result in low U2 utility values.

Topic	Rocchio	WH	k-NN	GIS-W	GIS-R
3	0.267	0.371	0.296	0.323	0.354
5	0.289	0.237	0.286	0.223	0.326
7	0.080	0.174	0.094	0.078	0.101
18	0.007	0.005	0.003	0.003	0.004
19	0.065	0.231	0.117	0.195	0.107
20	0.053	0.027	0.079	0.040	0.090
21	0.202	0.226	0.202	0.212	0.237
23	0.063	0.088	0.062	0.029	0.136
24	0.067	0.021	0.027	0.000	0.073
29	0.138	0.197	0.167	0.160	0.211
32	0.231	0.311	0.226	0.303	0.295
36	0.116	0.094	0.097	0.120	0.177
37	0.084	0.080	0.029	0.080	0.119
40	0.138	0.168	0.134	0.150	0.194
42	0.034	0.007	0.049	0.002	0.038
Avg. ASP	0.122	0.149	0.124	0.128	0.164

Table 9.1: Filtering performance based on ASP score. The parameters, which correspond to the best results, are: $\eta = 0.75$ for Rocchio, $\eta = 0.25$ for WH, $k = 300$ for k-NN.

Topic	Rocchio	WH	k-NN	GIS-W	GIS-R
3	1.000	**5.000**	2.000	3.000	4.000
5	4.000	2.000	3.000	1.000	**5.000**
7	2.000	**5.000**	3.000	1.000	4.000
18	**5.000**	4.000	1.000	2.000	3.000
19	1.000	**5.000**	3.000	4.000	2.000
20	3.000	1.000	4.000	2.000	**5.000**
21	1.000	4.000	2.000	3.000	**5.000**
23	3.000	4.000	2.000	1.000	**5.000**
24	4.000	2.000	3.000	1.000	**5.000**
29	1.000	4.000	3.000	2.000	**5.000**
32	2.000	**5.000**	1.000	4.000	3.000
36	3.000	1.000	2.000	4.000	**5.000**
37	4.000	3.000	1.000	2.000	**5.000**
40	2.000	4.000	1.000	3.000	**5.000**
42	3.000	2.000	**5.000**	1.000	4.000
Avg. ranking score	2.600	3.400	2.400	2.267	**4.333**

Table 9.2: Filtering performance based on ASP ranking score. The parameters, which correspond to the best results, are: $\eta = 0.75$ for Rocchio, $\eta = 0.25$ for WH, $k = 300$ for k-NN.

Topic	Rocchio	WH	k-NN	GIS-W	GIS-R
3	74.000	110.000	86.000	96.000	74.000
5	33.000	46.000	30.000	51.000	80.000
7	-57.000	-40.000	6.000	-45.000	-91.000
18	-14.000	-10.000	0.000	-6.000	-6.000
19	-91.000	44.000	21.000	54.000	21.000
20	-2.000	-3.000	6.000	-22.000	14.000
21	73.000	-51.000	-15.000	76.000	140.000
23	-2.000	42.000	9.000	-3.000	16.000
24	-17.000	-39.000	-13.000	-2.000	-11.000
29	53.000	100.000	80.000	112.000	143.000
32	34.000	46.000	31.000	44.000	48.000
36	6.000	-9.000	-11.000	11.000	23.000
37	29.000	4.000	3.000	31.000	12.000
40	17.000	13.000	68.000	13.000	93.000
42	8.000	-46.000	6.000	-100.000	3.000
Avg. U1	9.600	13.800	20.467	20.667	37.267

Table 9.3: Filtering performance based on U1 score. The parameters, which correspond to the best results, are: $\eta = 0.75$ for Rocchio, $\eta = 0.25$ for WH, $k = 200$ for k-NN.

Topic	Rocchio	WH	k-NN	GIS-W	GIS-R
3	1.500	**5.000**	3.000	4.000	1.500
5	2.000	3.000	1.000	4.000	**5.000**
7	2.000	4.000	**5.000**	3.000	1.000
18	1.000	2.000	**5.000**	3.500	3.500
19	1.000	4.000	2.500	**5.000**	2.500
20	3.000	2.000	4.000	1.000	**5.000**
21	3.000	1.000	2.000	4.000	**5.000**
23	2.000	**5.000**	3.000	1.000	4.000
24	2.000	1.000	3.000	**5.000**	4.000
29	1.000	3.000	2.000	4.000	**5.000**
32	2.000	4.000	1.000	3.000	**5.000**
36	3.000	2.000	1.000	4.000	**5.000**
37	4.000	2.000	1.000	**5.000**	3.000
40	3.000	1.500	4.000	1.500	**5.000**
42	**5.000**	2.000	4.000	1.000	3.000
Avg. ranking score	2.367	2.767	2.767	3.267	**3.833**

Table 9.4: Filtering performance based on U1 ranking score. The parameters, which correspond to the best results, are: $\eta = 0.75$ for Rocchio, $\eta = 0.25$ for WH, $k = 200$ for k-NN.

Topic	Rocchio	WH	k-NN	GIS-W	GIS-R
3	38.000	75.000	21.000	69.000	79.000
5	67.000	25.000	42.000	6.000	66.000
7	-138.000	-29.000	-80.000	-102.000	-37.000
18	-42.000	-42.000	-38.000	-40.000	-40.000
19	-49.000	6.000	-21.000	11.000	-3.000
20	-61.000	-61.000	-86.000	-75.000	-56.000
21	44.000	160.000	21.000	187.000	200.000
23	-114.000	-139.000	-61.000	-120.000	-120.000
24	-55.000	-100.000	-65.000	-94.000	-62.000
29	-37.000	53.000	5.000	24.000	128.000
32	12.000	11.000	12.000	24.000	54.000
36	-28.000	-40.000	-27.000	-36.000	-16.000
37	-215.000	-139.000	-175.000	-109.000	-105.000
40	-41.000	-105.000	-42.000	-147.000	40.000
42	-41.000	-70.000	-39.000	-97.000	-40.000
Avg. U2	-44.000	-26.333	-35.533	-33.267	5.867

Table 9.5: Filtering performance based on U2 score. The parameters, which correspond to the best results, are: $\eta = 0.75$ for Rocchio, $\eta = 0.25$ for WH, $k = 300$ for k-NN.

Topic	Rocchio	WH	k-NN	GIS-W	GIS-R
3	2.000	4.000	1.000	3.000	**5.000**
5	**5.000**	2.000	3.000	1.000	4.000
7	1.000	**5.000**	3.000	2.000	4.000
18	1.500	1.500	**5.000**	3.500	3.500
19	1.000	4.000	2.000	**5.000**	3.000
20	3.500	3.500	1.000	2.000	**5.000**
21	2.000	3.000	1.000	4.000	**5.000**
23	4.000	1.000	**5.000**	2.500	2.500
24	**5.000**	1.000	3.000	2.000	4.000
29	1.000	4.000	2.000	3.000	**5.000**
32	2.500	1.000	2.500	4.000	**5.000**
36	3.000	1.000	4.000	2.000	**5.000**
37	1.000	3.000	2.000	4.000	**5.000**
40	4.000	2.000	3.000	1.000	**5.000**
42	3.000	2.000	**5.000**	1.000	4.000
Avg. ranking score	2.633	2.533	2.833	2.667	**4.333**

Table 9.6: Filtering performance based on U2 ranking score. The parameters, which correspond to the best results, are: $\eta = 0.75$ for Rocchio, $\eta = 0.25$ for WH, $k = 300$ for k-NN.

Chapter 10

Conclusions and Future Work

10.1 Conclusions

We have conducted research on learning document classification scheme and investigate its application to automatic text categorization and text filtering problems. We have studied several machine learning approaches for automatic document classification. Two new techniques for automatic document classification have proposed, namely IBRI and GIS. The GIS algorithm is further refined to solve the text filtering problem.

We investigate some existing approaches such as rule-based techniques. After identifying the shortcomings of rule-based and instance-based approaches, we propose a new technique known as the IBRI algorithm by combining the strengths of them. We have implemented our approach and extensive experiments have been conducted on a large-scale, real-world document corpus,

namely the Reuters-21578 collection. The results show that our new approach outperforms rule-based and instance-based approaches.

Based on the idea of IBRI, we further investigate several recent approaches for document classification under the framework of similarity-based learning. They include two families of techniques, namely the k-nearest neighbor (k-NN) algorithm and linear classifiers. After identifying the weakness and strengths of each technique, we propose another new technique known as the generalized instance set (GIS) algorithm by unifying the strengths of k-NN and linear classifiers and adapting to characteristics of document classification problems. We have implemented our GIS algorithm, the ExpNet algorithm, and some linear classifiers. Extensive experiments have been conducted on two common benchmark document corpora, namely the OHSUMED collection and the Reuters-21578 collection. The results show that our new approach outperforms the latest k-NN approach and linear classifiers.

Our GIS approach have been refined to solve the text filtering problem. We have compared the filtering performance of GIS, linear classifiers, and k-NN. Extensive experiments have been conducted on a benchmark document filtering corpus, namely the FBIS document collection. The results also show that our new approach outperforms the latest k-NN approach and linear classifiers in our experiments.

10.2 Future Work

The performance of our GIS approach for automatic document categorization and information filtering has already been shown to be useful. More research can be done to further explore the potential benefits of it. It includes the followings:

- The effect of different generalization functions in the GIS approach can be investigated. Experimental results show that the better the generalization function, the better is the performance of GIS.

- Advanced feature selection such as mutual information gain for reducing the dimension of the document vector can be employed. This may increase the learning rate of the system and speed up the on-line document categorization as well as the information filtering task.

Appendix A

Sample Documents in the corpora

This Appendix gives the sample documents of our experiments.

- Table A.1 shows an sample document of the Reuters-21578 corpus.
- Table A.2 shows an sample document of the OHSUMED corpus.
- Table A.3 shows an sample document of the FBIS corpus.

```
<REUTERS TOPICS="YES" LEWISSPLIT="TRAIN" CGISPLIT="TRAINING-SET"
OLDID="5549" NEWID="6">
<DATE>26-FEB-1987 15:14:36.41</DATE>
<TOPICS><D>veg-oil</D><D>linseed</D><D>lin-oil</D><D>soy-oil</D>
<D>sun-oil</D><D>soybean</D><D>oilseed</D><D>corn</D><D>sunseed</D>
<D>grain</D><D>sorghum</D><D>wheat</D></TOPICS>
<PLACES><D>argentina</D></PLACES>
<PEOPLE></PEOPLE>
<ORGS></ORGS>
<EXCHANGES></EXCHANGES>
<COMPANIES></COMPANIES>
<TEXT>
<TITLE>ARGENTINE 1986/87 GRAIN/OILSEED REGISTRATIONS</TITLE>
<DATELINE>    BUENOS AIRES, Feb 26 - </DATELINE>
<BODY>
Argentine grain board figures show crop registrations of grains,
oilseeds and their products to February 11, in thousands of tonnes,
showing those for future shipments month, 1986/87 total and 1985/86
total to February 12, 1986, in brackets:

  Bread wheat prev 1,655.8, Feb 872.0, March 164.6, total 2,692.4 (4,161.0).
  Maize Mar 48.0, total 48.0 (nil).
  Sorghum nil (nil)

  Oilseed export registrations were:
  Sunflowerseed total 15.0 (7.9)
  Soybean May 20.0, total 20.0 (nil)

  The board also detailed export registrations for subproducts, as follows,
  SUBPRODUCTS

  Wheat prev 39.9, Feb 48.7, March 13.2, Apr 10.0, total 111.8 (82.7).
  Linseed prev 34.8, Feb 32.9, Mar 6.8, Apr 6.3, total 80.8 (87.4).
  Soybean prev 100.9, Feb 45.1, MAr nil, Apr nil, May 20.0,
  total 166.1 (218.5).
  Sunflowerseed prev 48.6, Feb 61.5, Mar 25.1, Apr 14.5,
  total 149.8 (145.3).

  Vegetable oil registrations were :
  Sunoil prev 37.4, Feb 107.3, Mar 24.5, Apr 3.2, May nil,
    Jun 10.0, total 182.4 (117.6).
  Linoil prev 15.9, Feb 23.6, Mar 20.4, Apr 2.0, total 61.8, (76.1).
  Soybean oil prev 3.7, Feb 21.1, Mar nil, Apr 2.0, May 9.0,
    Jun 13.0, Jul 7.0, total 55.8 (33.7).

</BODY>
</TEXT>
</REUTERS>
```


Table A.1: A sample document in the Reuters-21578 corpus.

.I 12 274249
.T
Haemophilus influenzae meningitis with prolonged hospital course.
.W
A retrospective evaluation of Haemophilus influenzae type b meningitis observed over a 2-year period documented 86 cases. Eight of these patients demonstrated an unusual clinical course characterized by persistent fever (duration: greater than 10 days), cerebrospinal fluid pleocytosis, profound meningeal enhancement on computed tomography, significant morbidity, and a prolonged hospital course. The mean age of these 8 patients was 6 months, in contrast to a mean age of 14 months for the entire group. Two patients had clinical evidence of relapse. Four of the 8 patients
tested for latex particle agglutination in the cerebrospinal fluid remained positive after 10 days. All patients received antimicrobial therapy until they were afebrile for a minimum of 5 days. Subsequent neurologic examination revealed a persistent seizure disorder in 5 patients (62.5%), moderate-to-profound hearing loss in 2 (25%), mild ataxia in 1 (12.5%), and developmental delay with hydrocephalus which required shunting in 1 (12.5%). One patient had no sequelae.

Table A.2: A sample document in the OHSUMED corpus.

<DOC>
<DOCNO> FBIS3-49 </DOCNO>
<HT>   "cr00000015894001" </HT>
<HEADER>
<DATE1>   22 March 1994 </DATE1>
Article Type:FBIS
DUE TO COPYRIGHT OR OTHER RESTRICTIONS THE FOLLOWING
ITEM IS INTENDED FOR USE ONLY BY U.S. GOVERNMENT
CONSUMERS. IT IS BASED ON FOREIGN MEDIA CONTENT AND
BEHAVIOR AND IS ISSUED WITHOUT COORDINATION WITH OTHER
U.S. GOVERNMENT COMPONENTS.
Document Type:FBIS TRENDS-21MAR94-DPRK-ROK-U.S.-IAEA
<H3> <TI>   DPRK-ROK-U. S.-IAEA </TI></H3>
</HEADER>
<TEXT>
SUMMARY

In a string of confrontational actions and pronouncements,

Pyongyang has raised the stakes further in its high-stakes nuclear issues game by threatening to withdraw from the Nuclear Nonproliferation Treaty (NPT). Although apparently unwilling to publicly concede that its own recent actions have damaged the chances for resumption of high-level U.S.-North Korea dialogue, Pyongyang appears to be trying to somehow salvage bilateral talks with Washington.

END SUMMARY

Pyongyang Threatens NPT Withdrawal, Calls for U.S. Talks

Pyongyang's threat to withdraw from the NPT came in an authoritative Foreign Ministry spokesman's statement issued on 21 March (Pyongyang radio, 21 March). In the statement Pyongyang said it will "carry into practice" its declaration of NPT withdrawal announced on 12 March 1993 under certain conditions:

+ If the United States refuses bilateral talks with North Korea and resumes Team Spirit, thereby "increasing its nuclear threat" against the DPRK.

+ If the IAEA "further expands it unfairness" by "distorting" the results of its recent inspections in North Korea and resorts to "forcible measures and pressures."

The 21 March statement, which for the first time acknowledged that the DPRK-U.S. talks scheduled for the same day in Geneva did not take place, also repeated familiar North Korean charges of U.S. violations of previous agreements with North Korea and declared that Pyongyang will "no longer" be duty-bound to ensure continuity of NPT safeguards.

Hint of Moderation Possibly seeking to soften its threat--and thus somehow leave the door ajar for dialogue with Washington--the statement seemed to imply that as long as there is any hope of resuming bilateral talks with Washington, Pyongyang may refrain from taking the final step of NPT withdrawal. North Korean withdrawal from the NPT, the statement said, would occur only if the United States "persists in avoiding" bilateral talks "to the end." Underscoring Pyongyang's reluctance to definitively rule out future talks, the statement asserted that the DPRK is "in no hurry at all"--presumably, to foreclose avenues of dialogue with Washington--"even if the DPRK-U.S. talks are not held." In addition, in contrast to North Korean pronouncements on the inter-Korean talks (see following article), the statement, like previous pronouncements on Washington's policy, refrained from extreme polemical attacks on Washington.

Pyongyang media also appeared to react with similar circumspection to recent statements by high U.S. officials criticizing the North. For example, monitored North Korean media so far have not reported or commented on CIA Director Woolsey's charges that Pyongyang has been exporting weapons of mass destruction, is supporting terrorism, and may already possess nuclear weapons (Yonhap, 18 March).

Implications Pyongyang clearly prefers to deal only with Washington on the nuclear issue. However, it seems unwilling or unable to acknowledge that its own dilatory and confrontational tactics are damaging the prospects for the dialogue it is seeking.

Pyongyang Breaks Off Talks With Seoul, Threatens War

SUMMARY

Accusing Seoul of engaging in pressure tactics, Pyongyang has abruptly broken off bilateral talks and issued threats of war--threats that it subsequently appeared to moderate. North Korea apparently hopes to gain from what it sees as differences within the ROK Government and between Seoul and Washington on the nuclear issue.

END SUMMARY

The North came to Panmunjom on 19 March clearly prepared to break off the ongoing negotiations with the South on the exchange of presidential envoys. The North Korean chief delegate--identified in Seoul media as Pak Yong-su--not only reinstated demands that he had previously indicated would be retracted, but also insisted that the ROK Government "apologize" to all Koreans--a formulation Pyongyang had typically used in the past to signal its noninterest in dialogue with Seoul and Tokyo--this time for allegedly obstructing DPRK-U.S. high-level talks (Pyongyang radio, 19 March).

The North's main complaint against the South focused on what Pak described as
Seoul's "sudden change of attitude" toward adoption of "tough measures" against Pyongyang. In particular, Pak said that North Korea's pique was based on reports of the 17 March "high-level strategic meeting" in Seoul, which had reportedly discussed holding Team Spirit, the deployment of Patriot missiles, and international sanctions. Pak equated this alleged change in the South's attitude to "a grave crime" and claimed that it necessitated the reimposition of the four demands made earlier by the North--demands that Pak had previously admitted were "barriers" to envoy exchanges erected by Pyongyang itself. (The North had demanded that the South refrain from 1) staging large-scale nuclear exercises, 2) introducing new weapons, including the Patriot missile, and 3) using threats of international sanctions. The North had also demanded previously that 4) the South retract ROK President Kim Yong-nam's remarks about not shaking hands with North Korean officials.)

Demand for Apology Signaling that Pyongyang is no longer interested in continuing dialogue with Seoul, Pak accused the South of having used the envoy exchange talks "solely" for the purpose of derailing DPRK-U.S. high-level talks. He went on to "strongly insist" that the South "frankly admit" this "dark intention" and "apologize before the nation." Attempting to place the onus of any breakdown of dialogue on the South, Pak further accused Seoul of having in effect adopted "a declaration of abandonment of the special envoys exchange, a declaration of an all-out confrontation with us, and a declaration of war."

Pyongyang radio on 19 March depicted Pak as making his most provocative remarks only in response to remarks the South made at the 19 March session. According to the radio, the South side had said that should the envoy exchange talks be discontinued, "there is no knowing what danger would materialize." To this Pak reportedly replied, "Don't worry about it. What do you think the South is? If the North suffers damage, do you think the South will go unscathed?" Pak went on to pledge unspecified "immediate and decisive countermeasures of self-defense" in case "some powers impose sanctions on us or otherwise provoke us." The radio said Pak coupled his pledge with a "stern warning" not to take the North Korean threats lightly, quoting him as saying, "we do not know how to engage in idle talk. "(SEE NOTE)

(NOTE: Seoul's government-run KBS-1 television on 19 March broadcast video recording of even more inflammatory remarks by Pak that were captured by closed-circuit television coverage of the meeting. In the video, Pak told his Southern counterpart, Song Yong-tae, to "give a careful consideration to the consequences of a war," and warned that "Seoul is not far from here. If war breaks out, it will turn into a sea of fire. Mr. Song, it will probably difficult for you, too, to survive." Pyongyang media have not been observed to report on this portion of Pak's remarks.)

Moderation of Threats Pyongyang subsequently appeared to backpedal a bit on its threats of war. In a 21 March statement issued in the name of the North Korean delegation to the inter-Korean contacts, which again chastised the South for "suddenly" assuming what it described as a hardline stance, the North made

only passing references to the possibility of war, implying instead that the downfall of the ROK government will come from revolt within the South (Pyongyang radio, 21 March). The statement warned the ROK to realize that "no dictators" in the South had "survived committing the antinational act of betraying fellow countrymen in collusion with outside forces" and that "these flunkeyist, nation-sellers have met a bitter end."

North Korean Motives Pyongyang may calculate that its brinksmanship will sufficiently split opinions within the South Korean government to make a firm stand against the North difficult. For the past few months, Seoul media reporting on relations with the North has indicated that there are serious divisions within the Seoul government over how to deal with Pyongyang. The latest of these indications came shortly after a 12 March inter-Korean meeting. The Seoul daily Hanguk Ilbo on 13 March cited an unnamed ROK "government official" as saying that the South and the United States "believe" that by failing to agree to the exchange of envoys with Seoul, Pyongyang had "unilaterally invalidated" an agreement reached with Washington on resuming high-level talks in Geneva later in the month. Apparently alarmed by the possible effects such remarks could have on the inter-Korean talks, other "government officials" were cited the next day by the South Korean news agency Yonhap as advocating a different approach (14 March). One of them was quoted as saying that Seoul should not consider it "a violation of the North Korea-U.S. agreement" even if North Korea "refused" the envoy exchange outright. All that would happen, he reportedly said, would be that "the date of the third round [U.S.-North Korea] meeting would be delayed" until the envoy exchange was realized.

In addition, there has been intermittent South Korean reporting of a division of views between Seoul and Washington that could have emboldened Pyongyang. Most recently, for instance, Yonhap on 16 March cited Kim Tae-chung, a "retired" opposition leader and former presidential candidate, as "lashing out" at U.S. "hardliners" for allegedly jeopardizing lives of Koreans in their pursuit of confrontation with North Korea on the nuclear issue. Similarly, in an article datelined Washington, the Seoul daily Choson Ilbo on 13 March claimed that the U.S. Congress has

concluded that differences between Washington and Seoul over how to deal with the North Korean nuclear issue are serving as a "stumbling block" in the U.S. Government's formulation of its policy toward Pyongyang--reporting that could have encouraged Pyongyang to reinforce its constant attempt to drive a wedge between Washington and Seoul.

Outlook Pyongyang media treatment of the 19 March meeting seems aimed at portraying the North Korean leadership as fully prepared to face the worst case scenario. By declaring Pyongyang's willingness to destroy Seoul in a conflict, the North Korean leadership may hope to frighten officials in the South into advocating concessions and thus further aggravate a perceived division between the United States and South Korea.

(AUTHOR: YIM. QUESTIONS AND/OR COMMENTS, PLEASE CALL CHIEF, ASIA DIVISION ANALYSIS TEAM, (703) 733-6534.)
EAG/HEBBEL/sdj 23/0017Z MAR

</TEXT>
</DOC>

Table A.3: A sample document in the FBIS corpus.

Appendix B

Details of Experimental Results of GIS

This Appendix give the detailed experimental results of GIS.

- Table B.1 shows the cross-validation microaveraged break-even point measures of Reuters-21578 corpus.
- Table B.2 shows the cross-validation microaveraged break-even point measures of OHSUMED corpus.
- Table B.3 shows the cross-validation F_1 measures of Reuters-21578 corpus.
- Table B.4 shows the cross-validation F_1 measures of OHSUMED corpus.
- Table B.5 shows the cross-validation microaveraged break-even point measures of Reuters-21578 corpus.

- Table B.6 shows the microaveraged break-even point measures of OHSUMED corpus.
- Table B.7 shows the F_1 measures of Reuters-21578 corpus.
- Table B.8 shows the F_1 measures of OHSUMED corpus.

Category	Rocchio	WH	KNN(100)	GIS-R	GIS-W
acq	0.757	0.771	0.819	0.887	0.874
alum	0.865	0.865	0.855	0.842	0.889
austdlr	0.500	0.500	0.500	0.500	0.500
barley	0.597	0.829	0.604	0.779	0.786
bop	0.634	0.684	0.634	0.622	0.649
can	0.500	0.500	0.500	0.500	0.500
carcass	0.696	0.674	0.668	0.683	0.709
castor-oil	0.500	0.500	0.500	0.500	0.500
cocoa	0.825	0.935	0.825	0.852	0.937
coconut	0.750	0.750	0.750	0.750	0.750
coconut-oil	0.533	0.400	0.533	0.533	0.200
coffee	0.899	0.930	0.878	0.904	0.938
copper	0.717	0.849	0.701	0.801	0.849
copra-cake	0.500	0.500	0.500	0.500	0.500
corn	0.639	0.808	0.717	0.800	0.813
cotton	0.740	0.686	0.628	0.767	0.826
cotton-oil	0.500	0.500	0.500	0.500	0.500
cpi	0.572	0.721	0.607	0.658	0.700
cpu	0.500	0.500	0.500	0.500	0.500
crude	0.774	0.816	0.792	0.807	0.841
dfl	0.500	0.500	0.500	0.500	0.500
dlr	0.545	0.594	0.557	0.652	0.606
dmk	0.365	0.292	0.411	0.365	0.421
earn	0.927	0.891	0.944	0.942	0.964
fuel	0.427	0.427	0.477	0.457	0.457
gas	0.513	0.735	0.555	0.640	0.676
gnp	0.763	0.819	0.791	0.806	0.841
gold	0.837	0.834	0.800	0.852	0.859
grain	0.751	0.899	0.799	0.836	0.925
groundnut	0.167	0.417	0.167	0.167	0.417
groundnut-oil	0.500	0.500	0.500	0.500	0.500
heat	0.407	0.337	0.556	0.407	0.527
hog	0.626	0.766	0.619	0.626	0.798
housing	0.686	0.573	0.740	0.686	0.723
income	0.658	0.658	0.658	0.658	0.658
instal-debt	0.667	0.667	0.667	0.667	0.333
interest	0.663	0.641	0.671	0.661	0.706
inventories	0.500	0.500	0.500	0.500	0.500
ipi	0.530	0.681	0.690	0.802	0.776
iron-steel	0.659	0.649	0.626	0.678	0.572
jet	0.375	0.375	0.375	0.375	0.375
jobs	0.747	0.828	0.759	0.828	0.840
l-cattle	0.333	0.583	0.333	0.333	0.583

lead	0.548	0.560	0.450	0.450	0.499
lei	0.768	0.786	0.768	0.749	0.768
lin-oil	0.500	0.500	0.500	0.500	0.500
linseed	0.500	0.500	0.500	0.500	0.500
livestock	0.661	0.685	0.691	0.704	0.756
lumber	0.282	0.365	0.223	0.582	0.465
meal-feed	0.330	0.624	0.519	0.550	0.679
money-fx	0.688	0.707	0.685	0.739	0.772
money-supply	0.599	0.585	0.689	0.657	0.712
naphtha	0.500	0.500	0.167	0.500	0.167
nat-gas	0.607	0.547	0.607	0.620	0.574
nickel	0.583	0.583	0.583	0.583	0.771
nkr	0.500	0.500	0.500	0.500	0.500
nzdlr	0.500	0.500	0.500	0.500	0.500
oat	0.490	0.146	0.490	0.469	0.219
oilseed	0.500	0.704	0.565	0.677	0.643
orange	0.705	0.697	0.667	0.705	0.705
palladium	0.500	0.500	0.500	0.500	0.500
palm-oil	0.761	0.823	0.761	0.761	0.823
palmkernel	0.500	0.500	0.500	0.500	0.500
pet-chem	0.411	0.428	0.426	0.475	0.475
platinum	0.677	0.656	0.656	0.708	0.656
plywood	0.500	0.500	0.500	0.500	0.500
potato	0.333	0.333	0.333	0.333	0.667
propane	0.625	0.625	0.625	0.375	0.563
rand	0.500	0.500	0.500	0.500	0.500
rape-oil	0.333	0.000	0.333	0.333	0.000
rapeseed	0.382	0.672	0.537	0.622	0.672
reserves	0.641	0.622	0.734	0.724	0.750
retail	0.405	0.370	0.455	0.461	0.484
rice	0.634	0.732	0.645	0.626	0.787
rubber	0.793	0.864	0.774	0.840	0.896
rye	0.500	0.500	0.500	0.500	0.500
saudriyal	0.500	0.500	0.500	0.500	0.500
ship	0.699	0.784	0.710	0.707	0.778
silver	0.583	0.699	0.513	0.572	0.709
sorghum	0.591	0.681	0.554	0.608	0.737
soy-meal	0.495	0.564	0.515	0.560	0.731
soy-oil	0.317	0.337	0.360	0.317	0.195
soybean	0.620	0.729	0.625	0.736	0.732
stg	0.624	0.593	0.624	0.624	0.568
strategic-metal	0.054	0.258	0.054	0.052	0.161
sugar	0.767	0.898	0.811	0.854	0.908

sun-oil	0.350	0.700	0.300	0.350	0.350
sunseed	0.420	0.517	0.417	0.420	0.478
tea	0.458	0.458	0.438	0.458	0.458
tin	0.856	0.878	0.836	0.856	0.878
trade	0.704	0.674	0.735	0.731	0.754
veg-oil	0.633	0.736	0.648	0.636	0.722
wheat	0.731	0.847	0.742	0.829	0.885
wool	0.500	0.500	0.500	0.500	0.500
wpi	0.598	0.716	0.598	0.598	0.574
yen	0.370	0.337	0.392	0.337	0.414
zinc	0.740	0.848	0.703	0.740	0.830
MBE	0.765	0.787	0.793	0.820	0.843

Table B.1: Cross-validation microaveraged break-even point measures of Reuters-21578 corpus.

Category	Rocchio	WH	k-NN	GIS-R	GIS-W
Angina Pectoris	0.383	0.519	0.452	0.494	0.585
Angina Pectoris, Variant	0.367	0.350	0.367	0.433	0.450
Angina, Unstable	0.550	0.651	0.585	0.612	0.615
Aortic Coarctation	0.714	0.780	0.752	0.794	0.780
Aortic Subvalvular Stenosis	0.333	0.333	0.333	0.333	0.333
Aortic Valve Insufficiency	0.393	0.364	0.518	0.506	0.536
Aortic Valve Stenosis	0.517	0.443	0.579	0.593	0.572
Arrhythmia	0.405	0.504	0.407	0.504	0.491
Arrhythmia, Sinus	0.125	0.125	0.125	0.125	0.125
Atrial Fibrillation	0.586	0.611	0.601	0.647	0.698
Atrial Flutter	0.510	0.718	0.601	0.631	0.749
Bradycardia	0.419	0.457	0.322	0.416	0.455
Bundle-Branch Block	0.598	0.517	0.586	0.623	0.578
Carcinoid Heart Disease	0.500	0.500	0.500	0.500	0.500
Cardiac Output, Low	0.066	0.041	0.024	0.053	0.024
Cardiac Tamponade	0.659	0.707	0.569	0.611	0.697
Cardiomyopathy, Alcoholic	0.333	0.333	0.333	0.333	0.333
Cardiomyopathy, Congestive	0.400	0.468	0.432	0.508	0.512
Cardiomyopathy, Hypertrophic	0.428	0.470	0.444	0.497	0.548
Cardiomyopathy, Restrictive	0.500	0.500	0.500	0.500	0.500
Chagas Cardiomyopathy	0.300	0.300	0.300	0.300	0.300
Cor Triatriatum	0.792	0.792	0.792	0.792	0.792
Coronary Aneurysm	0.543	0.162	0.595	0.595	0.297
Coronary Arteriosclerosis	0.263	0.329	0.438	0.427	0.426
Coronary Disease	0.543	0.554	0.580	0.599	0.603
Coronary Thrombosis	0.300	0.360	0.360	0.297	0.361
Coronary Vasospasm	0.429	0.418	0.495	0.550	0.491
Coronary Vessel Anomalies	0.361	0.319	0.584	0.584	0.500
Double Outlet Right Ventricle	0.125	0.375	0.625	0.125	0.375
Ductus Arteriosus, Patent	0.695	0.695	0.695	0.695	0.695
Ebstein's Anomaly	0.542	0.646	0.542	0.646	0.646
Eisenmenger Complex	0.500	0.500	0.500	0.500	0.500
Endocardial Cushion Defects	0.500	0.500	0.500	0.500	0.500
Endocardial Fibroelastosis	0.500	0.500	0.500	0.500	0.500
Endocarditis	0.357	0.221	0.287	0.329	0.280
Endocarditis, Bacterial	0.574	0.637	0.584	0.609	0.694
Endocarditis, Subacute Bacterial	0.500	0.500	0.500	0.500	0.500
Endomyocardial Fibrosis	0.375	0.000	0.500	0.375	0.000
Extrasystole	0.173	0.432	0.236	0.365	0.461
Heart Aneurysm	0.192	0.090	0.367	0.257	0.327
Heart Arrest	0.611	0.626	0.587	0.621	0.632
Heart Block	0.276	0.295	0.243	0.336	0.249

Heart Defects, Congenital	0.465	0.501	0.450	0.498	0.503
Heart Diseases	0.144	0.116	0.161	0.232	0.200
Heart Failure, Congestive	0.523	0.613	0.583	0.622	0.639
Heart Murmurs	0.250	0.250	0.250	0.472	0.250
Heart Neoplasms	0.388	0.390	0.372	0.420	0.374
Heart Rupture	0.287	0.220	0.265	0.310	0.265
Heart Rupture, Post-Infarction	0.583	0.367	0.583	0.583	0.500
Heart Septal Defects	0.423	0.000	0.403	0.403	0.403
Heart Septal Defects, Atrial	0.423	0.481	0.455	0.508	0.522
Heart Septal Defects, Ventricular	0.391	0.349	0.367	0.391	0.422
Heart Valve Diseases	0.209	0.311	0.215	0.250	0.293
Kearns Syndrome	0.625	0.625	0.625	0.625	0.625
Long QT Syndrome	0.467	0.533	0.533	0.467	0.467
Mitral Valve Insufficiency	0.555	0.547	0.532	0.551	0.595
Mitral Valve Prolapse	0.339	0.413	0.473	0.549	0.489
Mitral Valve Stenosis	0.549	0.607	0.631	0.622	0.673
Myocardial Diseases	0.116	0.177	0.169	0.215	0.207
Myocardial Infarction	0.645	0.673	0.660	0.704	0.710
Myocarditis	0.409	0.217	0.463	0.595	0.484
Pericardial Effusion	0.561	0.476	0.502	0.537	0.534
Pericarditis	0.267	0.327	0.350	0.368	0.368
Pericarditis, Constrictive	0.217	0.333	0.217	0.217	0.267
Pre-Excitation Syndromes	0.500	0.500	0.500	0.500	0.500
Pulmonary Heart Disease	0.854	0.854	0.854	0.854	0.854
Pulmonary Valve Insufficiency	0.500	0.500	0.500	0.500	0.500
Pulmonary Valve Stenosis	0.338	0.000	0.338	0.338	0.338
Rheumatic Heart Disease	0.142	0.000	0.058	0.058	0.200
Shock, Cardiogenic	0.333	0.465	0.438	0.355	0.423
Sick Sinus Syndrome	0.000	0.000	0.000	0.000	0.067
Sinoatrial Block	0.500	0.500	0.500	0.500	0.500
Tachycardia	0.589	0.629	0.667	0.662	0.679
Tachycardia, Atrioventricular Nodal Reentry	0.444	0.240	0.398	0.444	0.402
Tachycardia, Ectopic Atrial	0.500	0.500	0.500	0.500	0.500
Tachycardia, Ectopic Junctional	0.500	0.500	0.500	0.500	0.500
Tachycardia, Paroxysmal	0.495	0.475	0.500	0.442	0.442
Tachycardia, Supraventricular	0.475	0.616	0.577	0.619	0.631
Tetralogy of Fallot	0.648	0.804	0.627	0.648	0.804
Transposition of Great Vessels	0.448	0.410	0.448	0.556	0.547
Tricuspid Valve Insufficiency	0.444	0.479	0.607	0.549	0.561
Tricuspid Valve Stenosis	0.156	0.281	0.156	0.156	0.531
Truncus Arteriosus, Persistent	0.500	0.500	0.500	0.500	0.500
Ventricular Fibrillation	0.346	0.397	0.389	0.418	0.470
Ventricular Outflow Obstruction	0.317	0.142	0.325	0.383	0.392

Wolff-Parkinson-White Syndrome	0.669	0.652	0.669	0.669	0.686
Myocardial Reperfusion Injury	0.492	0.443	0.476	0.485	0.449
Torsades de Pointes	0.875	0.875	0.875	0.875	0.875
MBE	0.477	0.502	0.507	0.540	0.548

Table B.2: Cross-validation microaveraged break-even point measures of OHSUMED corpus.

Category	Rocchio	WH	k-NN	GIS-R	GIS-W
acq	0.749	0.705	0.816	0.882	0.895
alum	0.616	0.573	0.681	0.667	0.684
barley	0.430	0.712	0.342	0.662	0.631
bop	0.544	0.371	0.543	0.598	0.504
carcass	0.615	0.672	0.661	0.637	0.570
cocoa	0.763	0.863	0.752	0.776	0.864
coffee	0.865	0.877	0.881	0.876	0.888
copper	0.427	0.696	0.450	0.671	0.750
corn	0.623	0.802	0.685	0.787	0.838
cotton	0.561	0.434	0.464	0.669	0.805
cpi	0.480	0.558	0.532	0.604	0.589
crude	0.746	0.782	0.778	0.807	0.799
dlr	0.419	0.432	0.437	0.355	0.446
dmk	0.400	0.357	0.000	0.444	0.364
earn	0.925	0.848	0.940	0.946	0.964
fuel	0.204	0.512	0.431	0.215	0.431
gas	0.384	0.532	0.336	0.418	0.606
gnp	0.641	0.703	0.688	0.659	0.761
gold	0.641	0.653	0.669	0.651	0.670
grain	0.742	0.861	0.759	0.834	0.926
groundnut	0.065	0.222	0.022	0.091	0.286
heat	0.275	0.143	0.239	0.231	0.004
hog	0.457	0.182	0.424	0.472	0.222
housing	0.883	0.816	0.862	0.955	0.900
income	0.667	0.667	0.667	0.833	0.857
interest	0.653	0.561	0.653	0.678	0.720
ipi	0.395	0.556	0.473	0.599	0.520
iron-steel	0.511	0.482	0.570	0.588	0.520
jet	1.000	1.000	0.000	1.000	1.000
jobs	0.459	0.557	0.459	0.698	0.555
l-cattle	0.400	0.000	0.222	0.400	0.667
lead	0.474	0.300	0.588	0.582	0.286
lei	0.000	1.000	0.000	0.000	1.000
livestock	0.601	0.530	0.593	0.613	0.647
lumber	0.000	0.000	0.051	0.000	0.000
meal-feed	0.404	0.428	0.470	0.356	0.425
money-fx	0.640	0.660	0.594	0.739	0.771
money-supply	0.563	0.453	0.522	0.672	0.721

nat-gas	0.458	0.375	0.442	0.454	0.378
oat	0.333	0.000	0.167	0.000	0.000
oilseed	0.403	0.667	0.462	0.596	0.674
orange	0.394	0.325	0.308	0.299	0.319
palm-oil	0.532	0.521	0.532	0.341	0.502
pet-chem	0.300	0.306	0.276	0.300	0.310
platinum	0.722	0.686	0.682	0.700	0.733
rape-oil	0.000	0.000	0.000	0.000	0.000
rapeseed	0.161	0.421	0.192	0.345	0.405
reserves	0.516	0.433	0.514	0.617	0.534
retail	0.141	0.132	0.299	0.540	0.232
rice	0.462	0.622	0.452	0.491	0.712
rubber	0.640	0.698	0.668	0.726	0.685
ship	0.654	0.595	0.664	0.734	0.625
silver	0.405	0.429	0.349	0.263	0.364
sorghum	0.454	0.527	0.283	0.445	0.649
soy-meal	0.354	0.458	0.356	0.433	0.679
soy-oil	0.100	0.056	0.000	0.250	0.000
soybean	0.550	0.628	0.497	0.635	0.721
stg	0.800	0.400	0.800	0.800	0.800
strategic-metal	0.115	0.266	0.000	0.095	0.286
sugar	0.719	0.830	0.706	0.796	0.851
sun-oil	0.041	0.003	0.091	0.047	0.003
sunseed	0.417	0.500	0.378	0.241	0.278
tea	0.291	0.500	0.314	0.318	0.533
tin	0.608	0.542	0.492	0.608	0.564
trade	0.655	0.557	0.704	0.739	0.758
veg-oil	0.596	0.672	0.603	0.658	0.645
wheat	0.710	0.822	0.748	0.838	0.875
wpi	0.364	0.147	0.333	0.417	0.123
yen	0.546	0.482	0.619	0.611	0.625
zinc	0.352	0.437	0.308	0.288	0.437
AFM	0.486	0.508	0.464	0.533	0.563

Table B.3: Cross-validation F_1 measures of Reuters-21578 corpus.

Category	Rocchio	WH	k-NN	GIS-R	GIS-W
Angina Pectoris	0.300	0.454	0.401	0.445	0.418
Angina Pectoris, Variant	0.054	0.016	0.035	0.077	0.000
Angina, Unstable	0.331	0.588	0.492	0.529	0.514
Aortic Coarctation	0.619	0.000	0.833	0.700	0.533
Aortic Valve Insufficiency	0.189	0.199	0.309	0.171	0.291
Aortic Valve Stenosis	0.440	0.377	0.436	0.485	0.487
Arrhythmia	0.335	0.424	0.387	0.425	0.484
Atrial Fibrillation	0.381	0.484	0.465	0.542	0.461
Atrial Flutter	0.599	0.829	0.663	0.833	0.873
Bradycardia	0.209	0.317	0.261	0.324	0.368
Bundle-Branch Block	0.800	0.333	0.333	0.333	0.727
Cardiac Output, Low	0.040	0.022	0.047	0.093	0.033
Cardiac Tamponade	0.444	0.663	0.349	0.509	0.657
Cardiomyopathy, Congestive	0.210	0.346	0.227	0.341	0.260
Cardiomyopathy, Hypertrophic	0.170	0.296	0.140	0.169	0.167
Chagas Cardiomyopathy	0.500	0.000	0.000	0.500	0.500
Coronary Aneurysm	0.286	0.000	0.333	0.400	0.400
Coronary Arteriosclerosis	0.225	0.243	0.271	0.341	0.422
Coronary Disease	0.538	0.551	0.566	0.600	0.587
Coronary Thrombosis	0.267	0.285	0.334	0.202	0.064
Coronary Vasospasm	0.069	0.072	0.002	0.060	0.023
Coronary Vessel Anomalies	0.192	0.049	0.099	0.080	0.219
Ductus Arteriosus, Patent	0.015	0.003	0.002	0.013	0.004
Endocardial Cushion Defects	0.286	0.000	0.500	0.500	0.000
Endocardial Fibroelastosis	1.000	0.500	1.000	1.000	1.000
Endocarditis	0.062	0.003	0.061	0.002	0.001
Endocarditis, Bacterial	0.482	0.488	0.468	0.544	0.537
Extrasystole	0.149	0.068	0.223	0.144	0.177
Heart Aneurysm	0.168	0.111	0.268	0.195	0.100
Heart Arrest	0.570	0.600	0.546	0.595	0.568
Heart Block	0.086	0.079	0.126	0.151	0.135
Heart Defects, Congenital	0.360	0.462	0.449	0.456	0.478
Heart Diseases	0.120	0.059	0.109	0.138	0.071
Heart Failure, Congestive	0.506	0.544	0.559	0.600	0.619
Heart Murmurs	0.000	0.000	0.000	0.000	0.000
Heart Neoplasms	0.259	0.130	0.188	0.152	0.126
Heart Rupture	0.133	0.022	0.048	0.400	0.222
Heart Septal Defects	0.000	0.105	0.138	0.000	0.000
Heart Septal Defects, Atrial	0.211	0.375	0.250	0.508	0.583

Heart Septal Defects, Ventricular	0.275	0.264	0.313	0.400	0.369
Heart Valve Diseases	0.184	0.225	0.164	0.251	0.139
Kearns Syndrome	0.333	1.000	0.333	0.333	0.400
Long QT Syndrome	0.650	0.667	0.733	0.733	0.733
Mitral Valve Insufficiency	0.404	0.372	0.432	0.429	0.462
Mitral Valve Prolapse	0.182	0.330	0.215	0.394	0.466
Mitral Valve Stenosis	0.400	0.484	0.439	0.392	0.421
Myocardial Diseases	0.102	0.072	0.040	0.049	0.065
Myocardial Infarction	0.604	0.604	0.622	0.650	0.643
Myocarditis	0.398	0.310	0.383	0.530	0.579
Pericardial Effusion	0.273	0.300	0.300	0.273	0.333
Pericarditis	0.331	0.157	0.267	0.302	0.058
Pericarditis, Constrictive	0.571	0.667	0.750	0.571	0.667
Pre-Excitation Syndromes	0.043	0.053	0.000	0.044	0.044
Pulmonary Valve Stenosis	0.133	0.018	0.222	0.182	0.200
Rheumatic Heart Disease	0.068	0.041	0.070	0.083	0.048
Shock, Cardiogenic	0.221	0.407	0.234	0.218	0.265
Sick Sinus Syndrome	0.051	0.103	0.068	0.032	0.012
Sinoatrial Block	0.028	0.100	0.036	0.028	0.028
Tachycardia	0.511	0.499	0.528	0.639	0.611
Tachycardia, Atrioventricular Nodal Reentry	0.500	0.002	0.286	0.444	0.353
Tachycardia, Paroxysmal	0.400	0.000	0.400	0.400	0.750
Tachycardia, Supraventricular	0.338	0.508	0.415	0.415	0.527
Tetralogy of Fallot	0.000	0.400	0.000	0.400	0.400
Transposition of Great Vessels	0.607	0.402	0.540	0.523	0.695
Tricuspid Valve Insufficiency	0.450	0.192	0.550	0.571	0.550
Truncus Arteriosus, Persistent	0.000	0.000	0.000	0.000	0.000
Ventricular Fibrillation	0.191	0.257	0.246	0.388	0.329
Ventricular Outflow Obstruction	0.364	0.029	0.211	0.286	0.018
Wolff-Parkinson-White Syndrome	0.454	0.366	0.469	0.451	0.539
Myocardial Reperfusion Injury	0.393	0.341	0.369	0.423	0.428
Torsades de Pointes	0.000	0.000	0.000	0.000	0.000
AFM	0.297	0.271	0.304	0.343	0.341

Table B.4: Cross-validation F_1 measures of OHSUMED corpus

Category	Rocchio	WH	k-NN	GIS-R	GIS-W
acq	0.828	0.830	0.875	0.930	0.922
alum	0.766	0.826	0.739	0.783	0.766
barley	0.483	0.829	0.552	0.760	0.829
bop	0.525	0.558	0.590	0.590	0.722
carcass	0.703	0.703	0.703	0.595	0.703
castor-oil	0.500	0.500	0.500	0.500	0.500
cocoa	0.920	0.974	0.920	0.920	0.974
coconut	0.750	0.750	0.750	0.750	0.750
coconut-oil	0.667	0.417	0.667	0.000	0.000
coffee	0.913	0.913	0.877	0.913	0.913
copper	0.865	0.865	0.844	0.865	0.825
copra-cake	0.500	0.500	0.500	0.500	0.500
corn	0.620	0.850	0.702	0.842	0.850
cotton	0.732	0.781	0.586	0.732	0.635
cotton-oil	0.500	0.500	0.500	0.500	0.500
cpi	0.386	0.597	0.456	0.491	0.597
cpu	1.000	1.000	1.000	1.000	1.000
crude	0.795	0.839	0.818	0.839	0.850
dfl	0.500	0.500	0.500	0.500	0.500
dlr	0.652	0.607	0.652	0.719	0.719
dmk	0.225	0.000	0.225	0.225	0.225
earn	0.957	0.954	0.963	0.972	0.977
fuel	0.364	0.121	0.364	0.486	0.607
gas	0.458	0.858	0.458	0.572	0.686
gnp	0.733	0.862	0.845	0.873	0.873
gold	0.853	0.775	0.787	0.820	0.820
grain	0.803	0.903	0.816	0.829	0.913
groundnut	0.000	0.000	0.000	0.000	0.625
groundnut-oil	0.500	0.500	0.500	0.500	0.500
heat	0.550	0.550	0.733	0.550	0.550
hog	0.774	0.774	0.583	0.774	0.774
housing	0.675	0.675	0.675	0.675	0.675
income	0.686	0.514	0.857	0.857	0.514
instal-debt	1.000	1.000	1.000	1.000	0.000
interest	0.692	0.659	0.707	0.738	0.712
ipi	0.481	0.721	0.721	0.881	0.962
iron-steel	0.690	0.690	0.760	0.690	0.690
jet	0.000	0.000	0.000	0.000	0.000
jobs	0.605	0.838	0.745	0.931	0.884
l-cattle	0.000	0.417	0.000	0.000	0.417
lead	0.639	0.456	0.548	0.639	0.548
lei	0.875	0.875	0.875	0.875	0.875
lin-oil	0.500	0.500	0.500	0.500	0.500

Angina Pectoris	0.323	0.496	0.496	0.543	0.585
Angina Pectoris, Variant	0.000	0.292	0.267	0.292	0.292
Angina, Unstable	0.609	0.737	0.769	0.833	0.769
livestock	0.694	0.490	0.694	0.776	0.735
lumber	0.550	0.733	0.550	0.550	0.550
meal-feed	0.368	0.789	0.579	0.737	0.842
money-fx	0.585	0.680	0.652	0.708	0.763
money-supply	0.522	0.638	0.725	0.667	0.783
naphtha	0.500	0.500	0.500	0.500	0.500
nat-gas	0.525	0.656	0.656	0.722	0.656
nickel	0.750	0.750	0.750	0.750	0.750
nkr	0.500	0.500	0.500	0.500	0.500
nzdlr	0.500	0.500	0.500	0.500	0.500
oat	0.500	0.250	0.500	0.250	0.250
oilseed	0.526	0.653	0.611	0.653	0.737
orange	0.909	0.764	0.909	0.909	0.909
palladium	0.500	0.500	0.500	0.500	0.500
palm-oil	0.764	0.764	0.764	0.668	0.764
palmkernel	0.500	0.500	0.500	0.500	0.500
pet-chem	0.436	0.348	0.348	0.436	0.436
platinum	0.571	0.571	0.571	0.571	0.571
potato	0.667	0.667	0.667	0.667	0.667
propane	0.667	0.667	0.667	0.667	0.667
rand	0.500	0.500	0.500	0.500	0.500
rape-oil	0.000	0.000	0.000	0.000	0.000
rapeseed	0.528	0.739	0.633	0.633	0.844
reserves	0.703	0.757	0.757	0.703	0.757
retail	0.375	0.417	0.417	0.417	0.417
rice	0.639	0.809	0.667	0.766	0.792
rubber	0.721	0.881	0.721	0.721	0.801
rye	0.500	0.500	0.500	0.500	0.500
ship	0.800	0.816	0.793	0.827	0.793
silver	0.590	0.826	0.450	0.590	0.708
sorghum	0.477	0.764	0.477	0.573	0.764
soy-meal	0.659	0.659	0.659	0.659	0.659
soy-oil	0.234	0.351	0.234	0.468	0.117
soybean	0.537	0.717	0.597	0.717	0.687
strategic-metal	0.101	0.202	0.101	0.101	0.101
sugar	0.685	0.893	0.795	0.767	0.849
sun-meal	0.500	0.500	0.500	0.500	0.500
sun-oil	0.750	0.750	0.000	0.750	0.000
sunseed	0.400	0.200	0.400	0.400	0.400

tea	0.675	0.675	0.450	0.675	0.675
tin	0.958	0.871	0.958	0.958	0.871
trade	0.732	0.672	0.754	0.766	0.737
veg-oil	0.613	0.613	0.613	0.560	0.632
wheat	0.713	0.839	0.713	0.825	0.839
wpi	0.573	0.477	0.642	0.764	0.573
yen	0.276	0.345	0.414	0.414	0.552
zinc	0.923	0.769	1.000	0.846	0.769
MBE	0.781	0.812	0.815	0.842	0.853

Table B.5: Cross-validation microaveraged break-even point measures of Reuters-21578 corpus.

Category	Rocchio	WH	k-NN	GIS-R	GIS-W
Angina Pectoris	0.323	0.496	0.496	0.543	0.585
Angina Pectoris, Variant	0.000	0.292	0.267	0.292	0.292
Angina, Unstable	0.609	0.737	0.769	0.833	0.769
Aortic Coarctation	0.710	0.840	0.775	0.904	0.904
Aortic Subvalvular Stenosis	0.000	0.000	0.000	0.000	0.000
Aortic Valve Insufficiency	0.458	0.515	0.572	0.629	0.572
Aortic Valve Stenosis	0.572	0.531	0.490	0.572	0.531
Arrhythmia	0.475	0.486	0.460	0.518	0.532
Arrhythmia, Sinus	0.000	0.000	0.000	0.000	0.000
Atrial Fibrillation	0.604	0.679	0.604	0.679	0.679
Atrial Flutter	0.487	0.893	0.649	0.649	0.812
Bradycardia	0.483	0.483	0.276	0.483	0.552
Bundle-Branch Block	0.633	0.422	0.528	0.528	0.739
Carcinoid Heart Disease	0.500	0.500	0.500	0.500	0.500
Cardiac Output, Low	0.073	0.000	0.000	0.000	0.000
Cardiac Tamponade	0.561	0.721	0.481	0.641	0.721
Cardiomyopathy, Congestive	0.423	0.563	0.479	0.563	0.479
Cardiomyopathy, Hypertrophic	0.483	0.345	0.552	0.552	0.414
Cardiomyopathy, Restrictive	0.000	0.000	0.000	0.000	0.000
Chagas Cardiomyopathy	0.000	0.000	0.000	0.000	0.000
Cor Triatriatum	0.833	0.833	0.833	0.833	0.833
Coronary Aneurysm	0.325	0.162	0.650	0.650	0.162
Coronary Arteriosclerosis	0.218	0.364	0.291	0.327	0.509
Coronary Disease	0.525	0.560	0.550	0.595	0.581
Coronary Thrombosis	0.329	0.362	0.460	0.387	0.394
Coronary Vasospasm	0.194	0.292	0.292	0.389	0.292
Coronary Vessel Anomalies	0.436	0.261	0.610	0.610	0.523
Double Outlet Right Ventricle	0.000	0.000	0.750	0.750	0.000
Ductus Arteriosus, Patent	0.675	0.675	0.675	0.675	0.675
Ebstein's Anomaly	0.350	0.833	0.417	0.417	0.833
Eisenmenger Complex	0.500	0.500	0.500	0.500	0.500
Endocardial Fibroelastosis	0.500	0.500	0.500	0.500	0.500
Endocarditis	0.236	0.236	0.118	0.354	0.000
Endocarditis, Bacterial	0.553	0.553	0.553	0.639	0.639
Endomyocardial Fibrosis	0.000	0.000	0.000	0.667	0.000
Extrasystole	0.317	0.317	0.211	0.211	0.317
Heart Aneurysm	0.236	0.000	0.236	0.236	0.118
Heart Arrest	0.641	0.602	0.583	0.641	0.596
Heart Block	0.477	0.095	0.286	0.382	0.382
Heart Defects, Congenital	0.440	0.550	0.484	0.484	0.609
Heart Diseases	0.195	0.195	0.194	0.211	0.114
Heart Failure, Congestive	0.473	0.562	0.552	0.591	0.581

Heart Murmurs	0.000	0.000	0.000	0.000	0.000
Heart Neoplasms	0.401	0.401	0.481	0.401	0.561
Heart Rupture	0.310	0.155	0.310	0.310	0.155
Heart Rupture, Post-Infarction	0.583	0.292	0.583	0.583	0.292
Heart Septal Defects	0.450	0.000	0.450	0.450	0.450
Heart Septal Defects, Atrial	0.367	0.550	0.183	0.550	0.550
Heart Septal Defects, Ventricular	0.593	0.445	0.519	0.593	0.519
Heart Valve Diseases	0.133	0.400	0.133	0.356	0.267
Kearns Syndrome	0.000	0.750	0.000	0.000	0.000
Long QT Syndrome	0.800	0.800	0.800	0.800	0.800
Mitral Valve Insufficiency	0.642	0.679	0.642	0.642	0.679
Mitral Valve Prolapse	0.310	0.464	0.464	0.619	0.464
Mitral Valve Stenosis	0.479	0.489	0.489	0.697	0.489
Myocardial Diseases	0.118	0.157	0.118	0.314	0.116
Myocardial Infarction	0.734	0.760	0.759	0.799	0.786
Myocarditis	0.118	0.236	0.354	0.354	0.590
Pericardial Effusion	0.528	0.528	0.422	0.528	0.633
Pericarditis	0.183	0.162	0.171	0.183	0.145
Pericarditis, Constrictive	0.733	0.550	0.733	0.550	0.550
Pulmonary Heart Disease	0.667	0.667	0.667	0.667	0.667
Pulmonary Valve Insufficiency	0.500	0.500	0.500	0.500	0.500
Pulmonary Valve Stenosis	0.238	0.292	0.292	0.292	0.000
Rheumatic Heart Disease	0.000	0.292	0.000	0.267	0.000
Shock, Cardiogenic	0.528	0.528	0.528	0.422	0.528
Sick Sinus Syndrome	0.000	0.000	0.000	0.000	0.000
Sinoatrial Block	0.500	0.500	0.500	0.500	0.500
Tachycardia	0.673	0.555	0.673	0.654	0.654
Tachycardia, Atrioventricular Nodal Reentry	0.225	0.167	0.225	0.225	0.181
Tachycardia, Ectopic Atrial	0.500	0.500	0.500	0.500	0.500
Tachycardia, Ectopic Junctional	0.500	0.500	0.500	0.500	0.500
Tachycardia, Paroxysmal	0.444	0.222	0.444	0.667	0.222
Tachycardia, Supraventricular	0.566	0.679	0.566	0.679	0.717
Tetralogy of Fallot	0.583	0.583	0.583	0.583	0.583
Transposition of Great Vessels	0.367	0.183	0.550	0.550	0.550
Tricuspid Valve Insufficiency	0.183	0.183	0.183	0.183	0.367
Tricuspid Valve Stenosis	0.000	0.600	0.000	0.600	0.600
Truncus Arteriosus, Persistent	0.500	0.500	0.500	0.500	0.500
Ventricular Fibrillation	0.308	0.154	0.308	0.257	0.205
Ventricular Outflow Obstruction	0.183	0.000	0.367	0.367	0.000
Wolff-Parkinson-White Syndrome	0.733	0.733	0.733	0.733	0.733
Myocardial Reperfusion Injury	0.473	0.437	0.509	0.473	0.437
Torsades de Pointes	0.750	0.750	0.750	0.750	0.750
MBE	0.492	0.521	0.521	0.568	0.547

Table B.6: Microaveraged break-even point of OHSUMED corpus.

Category	Rocchio	WH	k-NN	GIS-R	GIS-W
acq	0.856	0.844	0.894	0.919	0.874
alum	0.385	0.385	0.385	0.385	0.444
barley	0.250	0.923	0.444	0.833	0.875
bop	0.537	0.414	0.684	0.647	0.791
carcass	0.545	0.333	0.526	0.526	0.609
cocoa	0.917	1.000	0.880	0.929	1.000
coffee	0.839	0.800	0.875	0.914	0.909
copper	0.857	0.857	0.900	0.900	0.900
corn	0.600	0.894	0.735	0.923	0.863
cotton	0.727	0.667	0.571	0.632	0.823
cotton-oil	0.002	0.002	0.001	0.002	0.002
cpi	0.492	0.472	0.528	0.548	0.536
crude	0.850	0.789	0.799	0.859	0.776
dlr	0.514	0.489	0.345	0.649	0.595
earn	0.964	0.952	0.965	0.962	0.987
fuel	0.000	0.133	0.000	0.000	0.000
gas	0.267	0.857	0.267	0.267	0.471
gnp	0.721	0.880	0.875	0.742	0.898
gold	0.788	0.722	0.811	0.765	0.811
grain	0.722	0.877	0.800	0.790	0.914
groundnut	0.051	1.000	0.000	0.111	0.000
heat	0.000	0.000	0.000	0.000	0.000
hog	0.000	0.000	0.000	0.000	0.000
income	0.625	0.600	0.667	0.833	0.667
interest	0.703	0.667	0.759	0.748	0.725
ipi	0.533	0.000	0.333	0.461	0.750
iron-steel	0.667	0.700	0.737	0.632	0.700
jobs	0.526	0.800	0.593	0.857	0.812
l-cattle	0.500	0.500	0.500	0.500	0.500
lead	0.947	0.286	0.900	0.842	0.857
lei	1.000	1.000	0.000	0.000	1.000
livestock	0.518	0.454	0.621	0.621	0.621
lumber	0.088	0.303	0.004	0.189	0.333
meal-feed	0.471	0.857	0.615	0.714	0.923
money-fx	0.369	0.608	0.664	0.676	0.768
money-supply	0.539	0.448	0.619	0.688	0.744
nat-gas	0.623	0.682	0.613	0.826	0.760
oat	0.000	0.000	0.000	0.000	0.000
oilseed	0.489	0.539	0.526	0.545	0.656
orange	0.667	0.500	0.667	0.667	0.667
palm-oil	0.667	0.800	0.800	0.667	0.667
pet-chem	0.000	0.286	0.308	0.308	0.400

platinum	1.000	0.889	1.000	1.000	0.889
rape-oil	0.000	0.000	0.000	0.286	0.000
rapeseed	0.250	0.250	0.250	0.250	0.250
reserves	0.500	0.774	0.727	0.759	0.118
rice	0.588	0.857	0.629	0.870	0.828
rubber	0.696	0.842	0.667	0.778	0.889
ship	0.763	0.330	0.816	0.827	0.519
silver	0.667	0.706	0.625	0.750	0.706
sorghum	0.667	0.800	0.667	0.667	0.667
soy-meal	0.667	0.400	0.667	0.667	0.889
soy-oil	0.286	0.000	0.286	0.286	0.038
soybean	0.552	0.710	0.667	0.703	0.667
strategic-metal	0.046	0.027	0.090	0.098	0.055
sugar	0.500	0.833	0.500	0.690	0.811
sun-oil	0.000	0.000	0.000	0.000	0.000
sunseed	0.000	0.000	0.000	0.000	0.000
tea	0.400	0.400	0.500	0.400	0.400
tin	1.000	0.909	1.000	1.000	0.909
trade	0.736	0.691	0.760	0.777	0.772
veg-oil	0.684	0.683	0.655	0.609	0.791
wheat	0.696	0.818	0.680	0.812	0.853
wpi	0.667	0.400	0.800	0.833	0.526
yen	0.429	0.167	0.182	0.200	0.333
zinc	0.429	0.000	0.533	0.429	0.000
AFM	0.516	0.543	0.529	0.572	0.584

Table B.7: F_1 measures of Reuters-21578 corpus.

Category	Rocchio	WH	k-NN	GIS-R	GIS-W
Angina Pectoris	0.345	0.442	0.458	0.539	0.538
Angina, Unstable	0.566	0.600	0.612	0.786	0.454
Aortic Coarctation	0.786	0.818	0.762	0.818	0.818
Aortic Valve Insufficiency	0.364	0.500	0.400	0.556	0.571
Aortic Valve Stenosis	0.516	0.222	0.235	0.467	0.348
Arrhythmia	0.516	0.417	0.521	0.654	0.581
Atrial Fibrillation	0.409	0.632	0.485	0.593	0.690
Atrial Flutter	0.375	0.909	0.375	0.375	0.737
Bradycardia	0.429	0.381	0.267	0.333	0.526
Bundle-Branch Block	0.438	0.500	0.128	0.636	0.769
Cardiac Output, Low	0.000	0.016	0.027	0.059	0.000
Cardiac Tamponade	0.333	0.533	0.429	0.667	0.625
Cardiomyopathy, Congestive	0.357	0.432	0.389	0.490	0.454
Cardiomyopathy, Hypertrophic	0.000	0.145	0.429	0.000	0.000
Chagas Cardiomyopathy	0.000	0.333	0.000	0.000	0.333
Cor Triatriatum	1.000	1.000	1.000	0.000	0.000
Coronary Aneurysm	0.103	0.002	0.081	0.030	0.006
Coronary Arteriosclerosis	0.304	0.342	0.133	0.258	0.522
Coronary Disease	0.487	0.475	0.530	0.564	0.571
Coronary Thrombosis	0.273	0.200	0.312	0.312	0.294
Coronary Vasospasm	0.167	0.267	0.364	0.300	0.182
Coronary Vessel Anomalies	0.429	0.286	0.500	0.714	0.500
Double Outlet Right Ventricle	0.000	0.000	0.000	0.000	0.000
Ductus Arteriosus, Patent	0.062	0.001	0.167	0.091	0.080
Ebstein's Anomaly	0.333	0.000	0.222	0.000	0.000
Endocarditis	0.154	0.114	0.167	0.300	0.235
Endocarditis, Bacterial	0.562	0.564	0.516	0.643	0.625
Extrasystole	0.308	0.250	0.222	0.000	0.222
Heart Aneurysm	0.235	0.017	0.296	0.211	0.050
Heart Arrest	0.633	0.595	0.480	0.588	0.557
Heart Block	0.200	0.093	0.000	0.200	0.200
Heart Defects, Congenital	0.364	0.583	0.500	0.542	0.577
Heart Diseases	0.184	0.180	0.049	0.174	0.049
Heart Failure, Congestive	0.349	0.519	0.487	0.569	0.525
Heart Neoplasms	0.571	0.222	0.600	0.600	0.600
Heart Rupture	0.286	0.005	0.250	0.094	0.014
Heart Septal Defects	0.000	0.000	0.000	0.000	0.000
Heart Septal Defects, Atrial	0.500	0.500	0.500	0.500	0.500
Heart Septal Defects, Ventricular	0.600	0.375	0.667	0.667	0.600
Heart Valve Diseases	0.279	0.421	0.329	0.190	0.349
Long QT Syndrome	0.667	0.667	0.667	0.667	0.667
Mitral Valve Insufficiency	0.429	0.480	0.414	0.348	0.461

Mitral Valve Prolapse	0.500	0.000	0.500	0.667	0.500
Mitral Valve Stenosis	0.545	0.392	0.588	0.560	0.647
Myocardial Diseases	0.056	0.103	0.143	0.357	0.095
Myocardial Infarction	0.759	0.764	0.772	0.785	0.812
Myocarditis	0.000	0.000	0.250	0.000	0.000
Pericardial Effusion	0.000	0.000	0.000	0.286	0.500
Pericarditis, Constrictive	0.000	0.500	0.000	0.333	0.444
Pulmonary Valve Stenosis	0.000	0.000	0.000	0.000	0.000
Shock, Cardiogenic	0.545	0.015	0.476	0.471	0.060
Tachycardia	0.667	0.486	0.659	0.559	0.529
Tachycardia, Paroxysmal	0.625	0.303	0.667	0.263	0.417
Tachycardia, Supraventricular	0.500	0.552	0.596	0.638	0.667
Transposition of Great Vessels	0.000	0.000	0.000	0.000	0.200
Ventricular Fibrillation	0.111	0.204	0.389	0.286	0.364
Ventricular Outflow Obstruction	0.222	0.133	0.000	0.286	0.400
Wolff-Parkinson-White Syndrome	0.400	0.667	0.000	0.400	0.857
Myocardial Reperfusion Injury	0.380	0.500	0.481	0.411	0.387
Torsades de Pointes	1.000	1.000	1.000	1.000	1.000
AFM	0.354	0.344	0.358	0.381	0.395

Table B.8: F_1 measures of OHSUMED corpus.

Appendix C

Computational Time of Reuters-21578 Experiments

This Appendix gives the detailed computational time (in seconds) of Rocchio, WH, k-NN and GIS algorithms on Reuters-21578 experiments.

- Table C.1 shows the computational time (in seconds) of each category on Reuters-21578 document corpus for Rocchio Algorithm.
- Table C.2 shows the computational time (in seconds) of each category on Reuters-21578 document corpus for WH Algorithm.
- Table C.3 shows the computational time (in seconds) of each category on Reuters-21578 document corpus for k-NN Algorithm.
- Table C.4 shows the computational time (in seconds) of each category on Reuters-21578 document corpus for GIS-W Algorithm.
- Table C.5 shows the computational time (in seconds) of each category on Reuters-21578 document corpus for GIS-R Algorithm.

Category	Training Time	Testing Time	Total Time
acq	2.47	2.71	5.18
alum	2.48	2.43	4.91
barley	2.47	2.40	4.87
bop	2.45	2.44	4.89
carcass	2.48	2.42	4.90
castor-oil	2.45	2.44	4.89
cocoa	2.47	2.49	4.96
coconut	2.51	2.39	4.90
coconut-oil	2.45	2.45	4.90
coffee	2.52	2.44	4.96
copper	2.46	2.49	4.95
copra-cake	2.46	2.40	4.86
corn	2.49	2.40	4.89
cotton	2.46	2.48	4.94
cotton-oil	2.45	2.48	4.93
cpi	2.48	2.45	4.93
cpu	2.49	2.44	4.93
crude	2.48	2.41	4.89
dfl	2.51	2.40	4.91
dlr	2.44	2.45	4.89
dmk	2.49	2.43	4.92
earn	2.46	2.51	4.97
fuel	2.49	2.42	4.91
gas	2.49	2.42	4.91
gnp	2.48	2.44	4.92
gold	2.52	2.50	5.02
grain	2.46	2.49	4.95
groundnut	2.47	2.50	4.97
groundnut-oil	2.49	2.44	4.93
heat	2.51	2.47	4.98
hog	2.46	2.39	4.85
housing	2.50	2.45	4.95
income	2.49	2.40	4.89
instal-debt	2.48	2.44	4.92
interest	2.46	2.42	4.88
ipi	2.48	2.47	4.95
iron-steel	2.49	2.44	4.93
jet	2.48	2.42	4.90
jobs	2.47	2.48	4.95
l-cattle	2.51	2.45	4.96
lead	2.47	2.44	4.91

lei	2.49	2.47	4.96
lin-oil	2.48	2.44	4.92
livestock	2.54	2.45	4.99
lumber	2.49	2.46	4.95
meal-feed	2.47	2.44	4.91
money-fx	2.52	2.46	4.98
money-supply	2.48	2.51	4.99
naphtha	2.47	2.45	4.92
nat-gas	2.50	2.44	4.94
nickel	2.47	2.49	4.96
nkr	2.45	2.50	4.95
nzdlr	2.51	2.44	4.95
oat	2.47	2.47	4.94
oilseed	2.52	2.49	5.01
orange	2.49	2.47	4.96
palladium	2.51	2.41	4.92
palm-oil	2.48	2.47	4.95
palmkernel	2.50	2.43	4.93
pet-chem	2.50	2.45	4.95
platinum	2.51	2.46	4.97
potato	2.50	2.46	4.96
propane	2.48	2.44	4.92
rand	2.50	2.42	4.92
rape-oil	2.52	2.43	4.95
rapeseed	2.48	2.48	4.96
reserves	2.52	2.43	4.95
retail	2.52	2.42	4.94
rice	2.50	2.46	4.96
rubber	2.48	2.41	4.89
rye	2.50	2.47	4.97
ship	2.53	2.47	5.00
silver	2.54	2.40	4.94
sorghum	2.51	2.45	4.96
soy-meal	2.48	2.55	5.03
soy-oil	2.49	2.48	4.97
soybean	2.50	2.46	4.96
strategic-metal	2.48	2.48	4.96
sugar	2.47	2.45	4.92
sun-meal	2.48	2.44	4.92
sun-oil	2.46	2.45	4.91
sunseed	2.57	2.42	4.99
tea	2.49	2.42	4.91
tin	2.52	2.41	4.93

trade	2.49	2.44	4.93
veg-oil	2.48	2.46	4.94
wheat	2.50	2.41	4.91
wpi	2.51	2.41	4.92
yen	2.50	2.41	4.91
zinc	2.48	2.46	4.94

Table C.1: Computational time (in seconds) of Rocchio algorithm of all categories in Reuters-21578 corpus.

Category	Training Time	Testing Time	Total Time
acq	14.58	2.54	17.12
alum	14.56	2.47	17.03
barley	14.55	2.49	17.04
bop	14.58	2.5	17.08
carcass	14.55	2.52	17.07
castor-oil	14.63	2.43	17.06
cocoa	14.68	2.5	17.18
coconut	14.68	2.41	17.09
coconut-oil	14.61	2.41	17.02
coffee	14.65	2.43	17.08
copper	14.69	2.48	17.17
copra-cake	14.74	2.48	17.22
corn	14.65	2.46	17.11
cotton	14.62	2.48	17.1
cotton-oil	14.62	2.44	17.06
cpi	14.58	2.55	17.13
cpu	14.56	2.46	17.02
crude	14.62	2.45	17.07
dfl	14.61	2.42	17.03
dlr	14.72	2.49	17.21
dmk	14.48	2.44	16.92
earn	14.61	2.55	17.16
fuel	14.53	2.46	16.99
gas	14.61	2.45	17.06
gnp	14.63	2.52	17.15
gold	14.58	2.48	17.06
grain	14.64	2.48	17.12
groundnut	14.56	2.50	17.06
groundnut-oil	14.58	2.47	17.05
heat	14.68	2.49	17.17
hog	14.64	2.47	17.11
housing	14.63	2.51	17.14
income	14.54	2.45	16.99
instal-debt	14.58	2.55	17.13
interest	14.72	2.53	17.25
ipi	14.69	2.47	17.16
iron-steel	14.57	2.51	17.08
jet	14.71	2.50	17.21
jobs	14.67	2.52	17.19
l-cattle	14.62	2.48	17.10
lead	14.67	2.49	17.16

lei	14.71	2.50	17.21
lin-oil	14.69	2.53	17.22
livestock	14.67	2.51	17.18
lumber	14.70	2.53	17.23
meal-feed	14.67	2.49	17.16
money-fx	14.65	2.46	17.11
money-supply	14.58	2.46	17.04
naphtha	14.58	2.46	17.04
nat-gas	14.65	2.45	17.10
nickel	14.59	2.51	17.10
nkr	14.52	2.40	16.92
nzdlr	14.59	2.46	17.05
oat	14.65	2.53	17.18
oilseed	14.60	2.47	17.07
orange	14.59	2.48	17.07
palladium	14.69	2.48	17.17
palm-oil	14.63	2.46	17.09
palmkernel	14.69	2.49	17.18
pet-chem	14.65	2.58	17.23
platinum	14.67	2.48	17.15
potato	14.66	2.45	17.11
propane	14.74	2.53	17.27
rand	14.61	2.45	17.06
rape-oil	14.69	2.53	17.22
rapeseed	14.63	2.50	17.13
reserves	14.65	2.48	17.13
retail	14.66	2.44	17.10
rice	14.54	2.51	17.05
rubber	14.61	2.46	17.07
rye	14.55	2.47	17.02
ship	14.63	2.51	17.14
silver	14.61	2.48	17.09
sorghum	14.62	2.52	17.14
soy-meal	14.65	2.49	17.14
soy-oil	14.59	2.49	17.08
soybean	14.61	2.52	17.13
strategic-metal	14.61	2.45	17.06
sugar	14.63	2.50	17.13
sun-meal	14.53	2.43	16.96
sun-oil	14.65	2.50	17.15
sunseed	14.61	2.47	17.08
tea	14.70	2.49	17.19
tin	14.60	2.43	17.03

trade	14.68	2.46	17.14
veg-oil	14.56	2.53	17.09
wheat	14.70	2.51	17.21
wpi	14.63	2.45	17.08
yen	14.71	2.50	17.21
zinc	14.65	2.49	17.14

Table C.2: Computational time (in seconds) of WH algorithm of all categories in Reuters-21578 corpus.

Category	Training Time	Testing Time	Total Time
acq	0.00	494.57	494.57
alum	0.00	495.12	495.12
barley	0.00	492.46	492.46
bop	0.00	499.07	499.07
carcass	0.00	492.28	492.28
castor-oil	0.00	499.00	499.00
cocoa	0.00	490.75	490.75
coconut	0.00	504.11	504.11
coconut-oil	0.00	487.12	487.12
coffee	0.00	496.06	496.06
copper	0.00	490.65	490.65
copra-cake	0.00	497.30	497.30
corn	0.00	489.26	489.26
cotton	0.00	493.91	493.91
cotton-oil	0.00	489.45	489.45
cpi	0.00	494.71	494.71
cpu	0.00	484.77	484.77
crude	0.00	495.49	495.49
dfl	0.00	483.60	483.60
dlr	0.00	492.47	492.47
dmk	0.00	487.94	487.94
earn	0.00	494.67	494.67
fuel	0.00	489.66	489.66
gas	0.00	494.32	494.32
gnp	0.00	488.24	488.24
gold	0.00	495.33	495.33
grain	0.00	486.43	486.43
groundnut	0.00	494.47	494.47
groundnut-oil	0.00	488.40	488.40
heat	0.00	493.98	493.98
hog	0.00	490.74	490.74
housing	0.00	490.88	490.88
income	0.00	489.95	489.95
instal-debt	0.00	490.66	490.66
interest	0.00	490.05	490.05
ipi	0.00	490.92	490.92
iron-steel	0.00	490.20	490.20
jet	0.00	490.94	490.94
jobs	0.00	489.96	489.96
l-cattle	0.00	490.95	490.95
lead	0.00	490.01	490.01

lei	0.00	491.07	491.07
lin-oil	0.00	490.01	490.01
livestock	0.00	487.56	487.56
lumber	0.00	485.96	485.96
meal-feed	0.00	488.39	488.39
money-fx	0.00	486.84	486.84
money-supply	0.00	488.37	488.37
naphtha	0.00	486.96	486.96
nat-gas	0.00	488.64	488.64
nickel	0.00	486.86	486.86
nkr	0.00	488.55	488.55
nzdlr	0.00	486.96	486.96
oat	0.00	488.41	488.41
oilseed	0.00	486.92	486.92
orange	0.00	488.56	488.56
palladium	0.00	487.06	487.06
palm-oil	0.00	487.09	487.09
palmkernel	0.00	485.85	485.85
pet-chem	0.00	487.26	487.26
platinum	0.00	485.88	485.88
potato	0.00	487.23	487.23
propane	0.00	485.97	485.97
rand	0.00	487.33	487.33
rape-oil	0.00	485.78	485.78
rapeseed	0.00	487.31	487.31
reserves	0.00	485.89	485.89
retail	0.00	487.07	487.07
rice	0.00	485.82	485.82
rubber	0.00	487.12	487.12
rye	0.00	485.75	485.75
ship	0.00	487.38	487.38
silver	0.00	486.29	486.29
sorghum	0.00	487.45	487.45
soy-meal	0.00	486.39	486.39
soy-oil	0.00	487.32	487.32
soybean	0.00	486.19	486.19
strategic-metal	0.00	487.39	487.39
sugar	0.00	486.33	486.33
sun-meal	0.00	487.32	487.32
sun-oil	0.00	486.47	486.47
sunseed	0.00	487.33	487.33
tea	0.00	486.53	486.53
tin	0.00	487.26	487.26

trade	0.00	486.37	486.37
veg-oil	0.00	485.86	485.86
wheat	0.00	485.40	485.40
wpi	0.00	486.13	486.13
yen	0.00	485.40	485.40
zinc	0.00	486.08	486.08

Table C.3: Computational time (in seconds) of k-NN algorithm of all categories in Reuters-21 578 corpus.

Category	Training Time	Testing Time	Total Time
acq	60.61	2.30	62.91
alum	4.54	0.87	5.41
barley	4.04	0.70	4.74
bop	3.70	0.87	4.57
carcass	6.51	0.82	7.33
castor-oil	0.14	0.14	0.28
cocoa	5.02	0.94	5.96
coconut	2.77	0.69	3.46
coconut-oil	4.68	0.85	5.53
coffee	6.59	0.85	7.44
copper	4.55	0.77	5.32
copra-cake	3.13	0.81	3.94
corn	5.35	0.74	6.09
cotton	6.34	0.81	7.15
cotton-oil	0.14	0.14	0.28
cpi	5.21	0.84	6.05
cpu	2.81	0.75	3.56
crude	31.74	1.50	33.24
dfl	3.29	0.78	4.07
dlr	4.97	0.78	5.75
dmk	3.27	0.78	4.05
earn	44.44	2.12	46.56
fuel	2.92	0.85	3.77
gas	5.91	0.87	6.78
gnp	8.54	1.09	9.63
gold	4.53	0.74	5.27
grain	17.84	1.27	19.11
groundnut	3.37	0.86	4.23
groundnut-oil	0.14	0.10	0.24
heat	4.42	0.74	5.16
hog	4.73	0.88	5.61
housing	3.90	0.75	4.65
income	4.25	0.73	4.98
instal-debt	4.53	0.66	5.19
interest	22.48	1.36	23.84
ipi	3.07	0.74	3.81
iron-steel	8.55	0.84	9.39
jet	4.43	0.68	5.11
jobs	3.01	0.71	3.72
l-cattle	3.54	0.96	4.50
lead	5.23	0.80	6.03

lei	3.01	0.78	3.79
lin-oil	0.12	0.12	0.24
livestock	8.02	0.82	8.84
lumber	7.86	1.12	8.98
meal-feed	7.03	0.81	7.84
money-fx	24.93	1.38	26.31
money-supply	11.16	1.06	12.22
naphtha	3.11	0.89	4.00
nat-gas	7.01	1.20	8.21
nickel	4.20	0.80	5.00
nkr	0.14	0.12	0.26
nzdlr	3.35	0.82	4.17
oat	4.27	0.82	5.09
oilseed	14.59	1.24	15.83
orange	4.57	0.88	5.45
palladium	1.55	0.17	1.72
palm-oil	4.36	0.81	5.17
palmkernel	2.99	0.82	3.81
pet-chem	6.22	0.88	7.10
platinum	3.17	0.88	4.05
potato	3.94	0.77	4.71
propane	1.79	0.20	1.99
rand	3.82	0.94	4.76
rape-oil	2.76	0.74	3.50
rapeseed	5.61	0.83	6.44
reserves	6.68	1.02	7.70
retail	2.56	0.66	3.22
rice	4.49	0.89	5.38
rubber	5.06	0.92	5.98
rye	0.15	0.11	0.26
ship	9.09	1.05	10.14
silver	2.81	0.74	3.55
sorghum	4.31	0.82	5.13
soy-meal	2.93	0.73	3.66
soy-oil	2.83	0.69	3.52
soybean	4.68	0.84	5.52
strategic-metal	10.75	1.13	11.88
sugar	6.67	0.78	7.45
sun-meal	0.14	0.14	0.28
sun-oil	4.05	0.87	4.92
sunseed	4.07	0.86	4.93
tea	6.18	0.92	7.10
tin	3.47	0.82	4.29

trade	29.18	1.42	30.60
veg-oil	7.99	1.15	9.14
wheat	4.80	0.75	5.55
wpi	2.72	0.73	3.45
yen	3.41	0.80	4.21
zinc	5.80	0.84	6.64

Table C.4: Computational time (in seconds) of GIS-W algorithm of all categories in Reuters-21578 corpus.

Category	Training Time	Testing Time	Total Time
acq	42.85	2.34	45.19
alum	3.88	0.73	4.61
barley	2.78	0.73	3.51
bop	3.40	0.72	4.12
carcass	3.16	0.87	4.03
castor-oil	0.14	0.13	0.27
cocoa	3.23	0.85	4.08
coconut	2.00	0.70	2.70
coconut-oil	3.14	0.87	4.01
coffee	3.32	0.82	4.14
copper	2.01	0.83	2.84
copra-cake	2.16	0.81	2.97
corn	5.94	0.96	6.90
cotton	2.19	0.84	3.03
cotton-oil	0.16	0.10	0.26
cpi	3.59	0.71	4.30
cpu	2.00	0.77	2.77
crude	17.94	1.44	19.38
dfl	2.28	0.83	3.11
dlr	6.28	1.03	7.31
dmk	3.96	0.80	4.76
earn	38.02	2.53	40.55
fuel	2.88	0.78	3.66
gas	5.72	0.68	6.40
gnp	4.48	1.10	5.58
gold	3.07	0.76	3.83
grain	12.81	1.38	14.19
groundnut	2.36	0.89	3.25
groundnut-oil	0.16	0.11	0.27
heat	3.12	0.82	3.94
hog	3.12	0.84	3.96
housing	2.77	0.70	3.47
income	2.88	0.67	3.55
instal-debt	2.62	0.64	3.26
interest	9.11	1.11	10.22
ipi	4.05	0.74	4.79
iron-steel	3.53	0.83	4.36
jet	3.14	0.77	3.91
jobs	4.14	0.89	5.03
l-cattle	2.32	0.95	3.27
lead	2.86	0.79	3.65

lei	2.04	0.77	2.81
lin-oil	0.14	0.17	0.31
livestock	3.23	0.81	4.04
lumber	5.31	1.06	6.37
meal-feed	4.91	0.93	5.84
money-fx	18.80	1.40	20.20
money-supply	11.84	1.13	12.97
naphtha	2.14	0.89	3.03
nat-gas	4.64	0.85	5.49
nickel	3.05	0.83	3.88
nkr	0.11	0.13	0.24
nzdlr	2.10	0.84	2.94
oat	3.04	0.88	3.92
oilseed	9.20	1.11	10.31
orange	2.91	0.86	3.77
palladium	1.11	0.13	1.24
palm-oil	3.01	0.87	3.88
palmkernel	2.15	0.82	2.97
pet-chem	6.91	0.85	7.76
platinum	2.17	0.87	3.04
potato	2.86	0.84	3.70
propane	1.28	0.18	1.46
rand	2.32	0.92	3.24
rape-oil	2.06	0.74	2.80
rapeseed	3.78	0.80	4.58
reserves	4.09	0.96	5.05
retail	1.91	0.67	2.58
rice	2.91	0.89	3.80
rubber	3.12	0.86	3.98
rye	0.15	0.17	0.32
ship	10.06	1.31	11.37
silver	2.89	0.79	3.68
sorghum	2.91	0.80	3.71
soy-meal	4.61	0.73	5.34
soy-oil	2.85	0.69	3.54
soybean	5.25	1.01	6.26
strategic-metal	6.81	1.30	8.11
sugar	8.50	1.18	9.68
sun-meal	0.14	0.14	0.28
sun-oil	2.87	0.80	3.67
sunseed	3.81	0.87	4.68
tea	4.15	0.92	5.07
tin	2.22	0.80	3.02

trade	14.38	1.37	15.75
veg-oil	7.06	1.09	8.15
wheat	7.86	1.16	9.02
wpi	1.93	0.72	2.65
yen	2.26	0.81	3.07
zinc	3.05	0.69	3.74

Table C.5: Computational time (in seconds) of GIS-R algorithm of all categories in Reuters-21578 corpus.

Bibliography

[1] D. Aha, D. Kibler, and M. K. Albert. Instance-based learning algorithms. In *Machine Learning,6*, pages 37–66, 1991.

[2] J. Allan, Callan J., W.B. Croft, L. Ballesteros, D. Byrd, R. Swan, and J. Xu. Inquery does battle with trec-6. In *In the 6th Text REtrieval Conference (TREC-6)*, page 169, 1997.

[3] C. Apte, F. Damerau, and S. Weiss. Automated learning of decision rules for text categorization. *ACM Transactions on Information Systems*, 12(3):233–251, 1994.

[4] P. Clark and T. Niblett. The cn2 induction algorithm. In *Machine Learning,3*, pages 261–283, 1989.

[5] W. W. Cohen. Fast effective rule induction. In *Machine Learning: Proceedings of the Twelfth International Conference*, 1995.

[6] W. W. Cohen. Learning with set-valued features. In *In Proceedings of*

the Thirtheenth National Conference on Artificial Intelligence, Portland, Oregon, 1996.

[7] S. Cost and S. Salzbery. A weighted nearest neighbor algorithm for learning with symbolic features. In *Machine Learning,10*, pages 57–78, 1993.

[8] T. M. Cover and P. E. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13:21–27, 1967.

[9] B. V. Dasarathy. Nosing around and neighborhood: A new system structure and classification rule for recognition in partially exposed environments,2. In *Pattern Analysis and Machine Intelligence*, pages 21–27, 1980.

[10] A. H. David. The trec-6 filtering track: Description and analysis. In *In the 6th Text REtrieval Conference (TREC-6)*, 1997.

[11] M. V. Eleen and Donna H. Overview of the 5th text retrieval conference(trec-5). In *In the 5th Text Tetrieval Conference (TREC-5), NIST SP 500-238*, pages 1–28, 1997.

[12] W. B. Frakes. Term conflation for information retrieval. In *The Third Joint BCS and ACM symposium on Research and Development in Information Retrieval, Cambridge, England.*, 1984.

[13] W. B. Frakes. Lattis: A corporate library and information system for the unix environment. In *Proceeding of the National Online Meeting, Medford, N.J., Learned Information Inc.*, pages 137–142, 1986.

[14] P. E. Hart. The condensed nearest neighbor rule. In *Institute of Electrical and Electronics Engineers and Transactions on Information Theory, 14*, pages 515–516, 1968.

[15] C. Y. Ho and W. Lam. Automatic discovery of document classification knowledges from text databases. *INFORMATION SYSTEMS Incorporating DATABASE TECHNOLOGY*, 1997. Submitted.

[16] A. Jennings and H. Higuchi. A personal news service based on a user model neural newtork. In *IEICE Transactions on Information and Systems*, 1992.

[17] E. Krol. The whole internet user's guide and catalog. In *O'Reilly and Associates, Sebastopol, California*, 1992.

[18] W. Lam and C. Y. Ho. Using a generalized instance set for automatic text categorization. In *Proceedings of the Twenty-First International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1998. To appear.

[19] W. Lam, K. F. Low, and C. Y. Ho. Using a bayesian network induction approach for text categorization. In *In Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence, (IJCAI), Nagoya, Japan*, pages 745–750, 1997.

[20] K. Lang. An adaptive multi-user text filter. Technical report, Technical report, School of Computer Science, Carnegie Mellon university, 1994.

[21] D. D. Lewis. Evaluating text categorization. In *The Speech and Natural Languare Workshop, Asilomar.*, 1991.

[22] D. D. Lewis, R. E. Schapore, J. P. Call, and R. Papka. Training algorithms for linear text classifiers. In *Proceedings of the Nineteenth International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 298–306, 1996.

[23] D. D. Lewis, R. E. Schapore, J. P. Callan, and R. Papka. Training algorithms for linear text classifiers. In *Proceedings of the Nineteenth International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 298–306, 1996.

[24] H. P. Luhn. A statistical approach to the mechanized encoding and searching of literary information. In *IBM Journal of Research and Development,1:4*, pages 309–317, 1957.

[25] B. Masand, G. Linoff, and D. Waltz. Classifying news stories using memory based reasoning. In *In Proceedings of the ACM SIGIR*, 1992.

[26] R. Mehnert. Federal agency and federal library reports: National library of medicine. In *Bowker Annual: Library and Book Trade Almanec, 2nd ed.*, pages 110–115, 1997.

[27] J. R. Quinlan. C4.5: Programs for machine learning. In *San Mateo, CA: Margan Kaufmann*, 1993.

[28] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. An open architecture for collaborative filtering of netnews. In *In Proceedings of ACM 1994 Conference on Computer Supported Cooperative Work*, pages 175–186, 1994.

[29] R. L. Rivest. Learning decision lists. In *Machine Learning,2*, pages 229–246, 1987.

[30] J. J. Rocchio Jr. *Relevance feedback in information retrieval.* The SMART Retrieval System: Experiments in Automatic Document Processing, editor: Gerard Salton, Prentice-Hall, Inc., Englewood Cliffs, News Jersey, 1971.

[31] G. Salton. A theory of indexing. In *Regional Conference Series in Applied Mathematics No. 18, Society for Industrial and Applied Mathematics, Philadelphia, PA*, 1975.

[32] G. Salton. *Automatic Text Processing: The Transformation , Analysis, and Retrieval of Information by Computer.* Addison-Wesley, Reading, Pennsylvania, 1989.

[33] G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval.* McGraw-Hill, 1983.

[34] G. Salton and C. S. Yang. On the specification of term values in automatic indexing. In *Journal of Documentation*, pages 351–372, 1973.

[35] M. W. Sholom and A. K. Casimir. *Computer Systems That Learn: Classification and Prediction Methods from Statistics, Neural Nets, Machine Learning, and Expert Systems.* Morgan Kaufmann Publishers, Inc., 1991.

[36] A. Singhal. At&t at trec-6. In *In the 6th Text REtrieval Conference (TREC-6)*, page 215, 1997.

[37] J. K. Sparck. A statistical interpretation of term specificity and its application in retrieval. In *Journal of Documentation, 28:1*, pages 11–21, 1972.

[38] Y. Y. Tak and Hector G. M. Sift: a tool for wide-area information dissemination. In *Proceedings of the 1995 USENIX Technical Conference*, pages 177–186, 1995.

[39] S. Weiss and N. Indurkhya. Optimized rule induction. *IEEE Exp. 8*, 6:61–69, 1993.

[40] B. Widrow and S.D. Stearns. *Adaptive Signal Processing.* Prentice-Hall, Englewood Cliffs, NJ, 1985.

[41] Y. Yang. Expert network: Effective and efficient learning from human decisions in text categorization and retrieval. In *Proceedings of the Seventeenth International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 13–22, 1994.

[42] Y. Yang and J. Pedersen. A comparative study on feature selection in

text categorization. In *In International Conference on Machine Learning (ICML)*, 1997.