This is to certify that the

thesis entitled

Text Document Classification

presented by

Yonghong Li

has been accepted towards fulfillment
of the requirements for

Master degree in Computer Science

Anil Kumar

Major professor

Date June 19, 1998

Text Document Classification

By

Yonghong Li

A Thesis

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

Computer Science

1998

Abstract

Text Document Classification

By

Yonghong Li

The exponential growth of the Internet has led to a great deal of interest in developing useful and efficient tools and software to assist users in searching the Web. Document retrieval, categorization, routing and filtering can all be formulated as classification problems. However, the complexity of natural languages and extremely high dimensionality of the feature space of documents have made this classification problem very difficult. We investigate five different methods for document classification: the naive Bayes classifier, nearest neighbor classifier, decision trees, subspace method, and multilayer feedforward neural network. These classifiers were applied to seven-class Yahoo news groups (Business, Entertainment, Health, International, Politics, Sports, and Technology) individually and in combination. Naive Bayes classifier and subspace classifier have also been tested on the enlarged Yahoo news group and Reuters-21578 benchmark. We studied three classifier combination approaches: simple voting, dynamic classifier selection, and adaptive classifier combination. These classifiers were also evaluated on different document representation schemes obtained

by best feature selection, principal component analysis and discriminant analysis. Our experimental results indicate that the naive Bayes classifier, subspace method, and multilayer feedforward neural network classifier outperform the other two classifiers on our data sets. Orthonormal discriminant analysis gives better classification results than other dimensionality reduction schemes. Combinations of multiple classifiers do not always improve the classification accuracy compared to the best individual classifier. Among the three different combination approaches, our adaptive classifier combination method introduced here performs the best. The best classification accuracy that we are able to achieve on this seven-class problem is approximately $83 - 85\%$, which is comparable to the performance of other similar studies. However, the classification problem considered here is more difficult because the pattern classes used in our experiments have a large overlap of words in their corresponding documents.

To My Husband **Lin Hong**

ACKNOWLEDGMENTS

I would like to acknowledge all the people who have assisted me during the years of my graduate study at Michigan State University. I am most grateful to my advisor, Dr. Anil Jain, for both his professional and personal advice, his continuous guidance, encouragement, support, and valuable time that he spent on projects with me.

My colleagues at the MSU PRIP lab and my friends at MSU have provided help and moral support throughout my study here. I would like to give my thanks to all of them. A special thanks to Aditya Vailaya, for the valuable discussions and help he gave me, not only as a colleague, but as a friend as well. I would like to specially acknowledge Shaoyun Chen for his great help in my projects and thesis, and for his assistance in using artificial neural network and discriminant analysis code. I would also like to thank Chitra Dorai for her valuable suggestion and her assistance to adjust to the new environments. Many thanks to Scott Connell, Salil Prabhakar, Nicolae Duta, and Marilyn Wulfekuhler for proofreading my papers, thesis and helpful discussions. My acknowledgment also goes to Jinhui Liu, Xiaojie Dong and Fan Du, for their help in manually classifying the data.

No words can express my thanks to my husband, for his love, understanding, and

support through all the difficulties, as well as for sharing the happiness. I must give my immense thanks to my parents. Their never-fading love, care, and encouragement are of immeasurable value to me.

I am also grateful to Dr. John Weng and Dr. Sridhar Mahadevan, for serving as my committee members, and for reviewing the thesis. I would also like to thank Cathy Davison, Mary Gebbia, Lora Mae Higbee, and Linda Moore for their administrative assistance.

TABLE OF CONTENTS

List of Tables

List of Figures

Chapter 1

Introduction

The World Wide Web (WWW) is a wide-area hypermedia information repository that aims to give world-wide access to the large repository of documents through the Internet. The WWW has inherent properties that makes it different from traditional digital libraries in the following ways:

(i) Unorganized, non-hierarchical: the information on the Web is distributed using a client/server model. The collection of documents resides on *servers*, and viewers access it from *clients*. Documents are connected by hyperlinks. In fact, the WWW was not really designed for organized information retrieval.

(ii) Dynamic: it is very easy for a user to post information, add new sites, update existing documents, as well as remove old files from the Web.

The WWW has exhibited exponential growth over the last few years, since it was created in 1989 at the European Laboratory for Particle Physics (CERN) in Geneva, Switzerland. A pessimistic estimate is that, in the middle of 1996, the WWW con-

sisted of more than 60 million documents on 12 million hosts and 600,000 servers, up from 9 million hosts and 250,000 servers at the beginning of the year [77], and it is growing everyday.

The rapid growth of Internet and the characteristics of the WWW have resulted in an information overload [41, 5, 46], which make it increasingly difficult for users to locate the relevant information quickly. This has led to a great deal of interest in developing useful and efficient tools and software to assist users in searching on the Web. In the following sections, we provide a brief introduction to various search engines and intelligent agents that are available to the users.

1.1 Search Engines

Search engines were designed to reduce the effort and information overload on the Web. Commercial search engines, such as Yahoo, HotBot, InfoSeek, WebCrawler, MetaCrawler, and Lycos, etc. are examples of tools that construct indices of Web resources and find information requested by a user. Basically, there are three different kinds of search engines: categorical, full-text, and meta[1]. They differ in the basic search logic. All search engines perform their function a bit differently, so their search results could be quite different.

[1] A Search Engine Primer: http://131.252.56.87/TandT/SENGINE.HTM

1.1.1 Categorical Engines

Categorical engines, like *Yahoo*[2] and *Magellan*[3], are more of a subject guide than a search engine. The guide is hierarchically organized into topic categories by human analysts, each category containing a list of manually indexed web pages. For example, *Yahoo* lists more than 200,000 Web sites under 20,000 different categories. The hierarchical organization provides a good structure for browsing. Starting with no specific indices, this kind of search engine can lead a user to quickly find information on various topics.

A nice property of categorical engines is that they extract high-level semantics (topics) and construct links between topics (directories and sub-directories). However, manual examination of each document is expensive, time-consuming and tedious. Thus, these engines are faced with a trade-off between the size and the quality of their directory. Moreover, the larger the number of analysts involved in examining the web documents, the larger the intra-class variability.

1.1.2 Full-Text Engines

All current search engines seem to follow *gathering-indexing-matching* strategy. Figure 1.1 which is adapted from [41] shows the structure of a search engine. They collect the Web resource by periodically dispatching programs, known as Web robots, crawlers, wanderers, spiders, worms and ants[4] to the Web. Indexing robots auto-

[2]http://www.yahoo.com
[3]http://www.mckinley.com
[4]The Web Robots Pages: http://info.webcrawler.com/mak/projects/robots/robots.html

matically traverse the Web's hypertext structure by starting from one document and recursively retrieving all documents that are referenced. Instead of organizing pages according to topics, these search engines work differently from categorical engines. They scan and analyze the internal text of a page, and perform keyword-based searches against an indexed database. The indexing and searching schemes of these search engines are similar, but they may differ in their database size, update frequency, search strategy and capability, and ways of representing the search results.

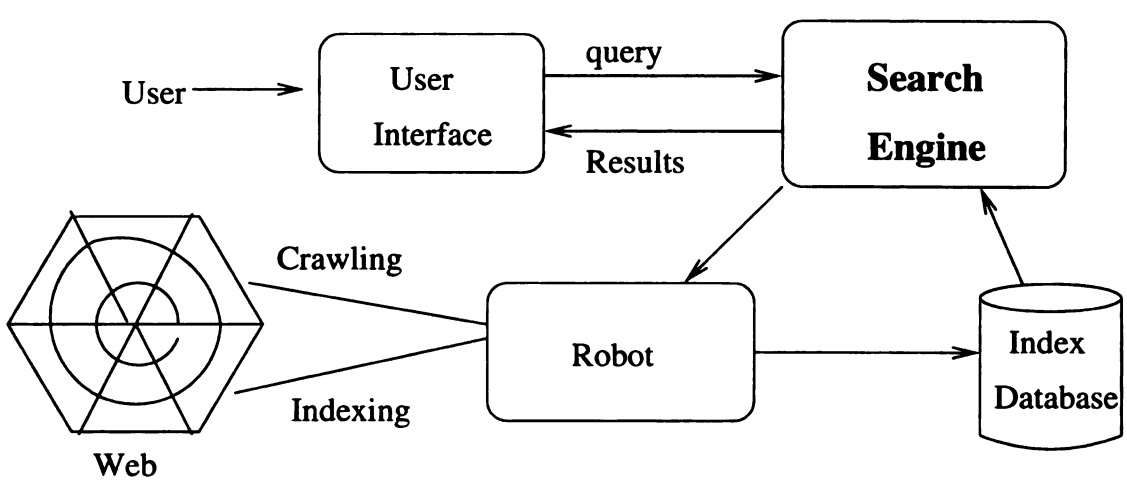

Figure 1.1: The structure of a search engine, adapted from [41].

Search engines mainly follow the so called *location/frequency* rule to determine the relevancy between the query keywords and retrieved documents. It is based on the assumption that any page relevant to the topic will mention those keywords right from the beginning, such as in the heading or in the first few lines of the page. Keyword frequency is the other major factor in determining the relevancy. Those pages with higher occurrence of keywords are more likely to be relevant than other Web documents. Some search engines add other strategies to "boost" the relevancy.

For example, WebCrawler uses link popularity as part of its ranking method. A search engine features chart[5] provides a summary of important features of the most commonly used search engines. We briefly summarize the properties in Table 1.1. Since each search engine has a different strategy for selecting pages to index, and

Search Engine	Size (no. of pages in millions)	Update Freq.	Ranking (in addition to location/ freq. method)	Comments
AltaVista	100	1 day to 1 month	-	one of the largest and most comprehensive search engines
Excite	55	3 weeks	3 or 4 star review	best for finding widely discussed, mainstream topics
HotBot	110	2 weeks	-	easy to scale up
InfoSeek	30	1 to 2 months	keywords in meta tag	one of the best tool for comprehensive search relevancy and speed
Lycos	30	1 to 2 weeks	-	Also offers to search a subject-based directory of the most popular pages from its database
WebCrawler	2	-	keywords in title, link popularity	good for beginners with keyword searching

Table 1.1: A brief summary of characteristics of the most commonly used search engines. These results are from footnote 5.

none of the search engines has a completely indexed database, the coverage of the individual engines is relatively low, i.e., searching with a second engine would often return several document that were not returned by the first engine. Also, because

[5]http://www.searchenginewatch.com/webmasters/features.html (the chart is as of Feb. 2, 1988)

search engines may have their down times, and because of network traffic congestion, an engine's response time is often dynamic. Meta engines are a new breed of search engines which combine the results of several engines for a given set of query keywords.

1.1.3 Meta Engines

The structure of a meta-search engine given in [41] is shown in Figure 1.2. This kind of engine provides a uniform query interface, accepts the query keywords entered by a user, and sends the query to many different search engines. It then collects the results from various search engines and analyzes them, eliminates duplicates, re-ranks pages on the basis of relevancy, and summarizes the information. MetaCrawler [72] and Fusion [74] are examples of such search engines.

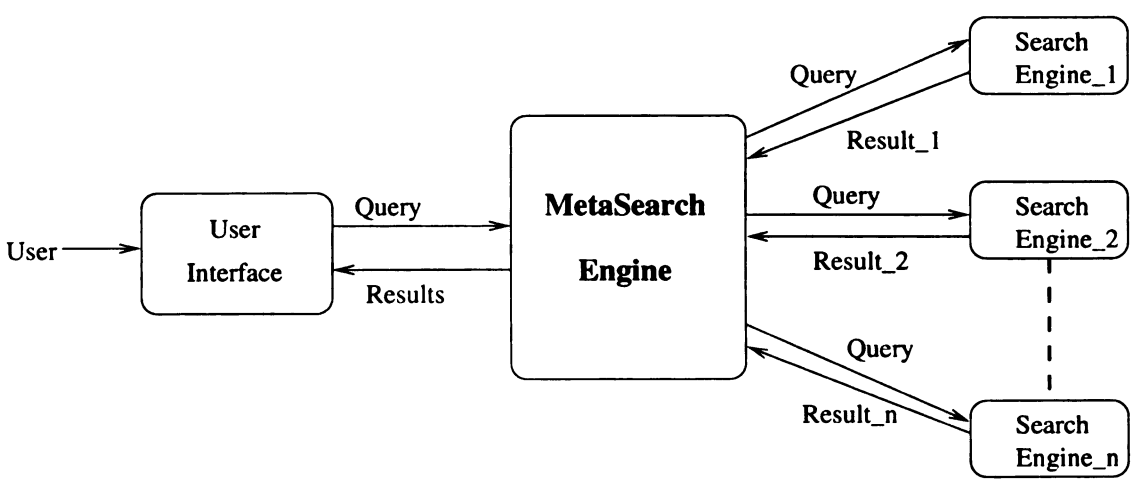

Figure 1.2: The structure of a meta-search engine, from [41]

Human indexing and categorization of Web pages are time-consuming and tedious. It is hard to keep pace with the explosive growth of Internet by analyzing each document manually. Moreover, keywords-based search is not always accurate enough. It is not uncommon that search engines, in response to a query, often return some

sites that have little to do with a user's interest. This has led to the development of *intelligent agents* which are playing an important role in making the Internet more usable [53, 6, 54].

1.2 Intelligent Agents

Intelligent agents can sense their environments, act on behalf of their owners, interact with other agents, and take actions in order to achieve their goals. They are being used in many applications, such as:

1. Electronic Commerce. As an example, *AuctionBot*[6] is a multi-purpose server used to create automated Internet auction according to user specifications, or bid in an existing AuctionBot auction. *Kasbah*[7] [9] is MIT's buying and selling agent in electronic marketplace; it is a multi-agent research project which aims at fundamentally transforming the way people make transactions. *Jango*[8] is a tool for shopping on the web. It is quoted by NetBot as the world's first intelligent shopping assistant.

2. Entertainment. As an example, *eGenie*[9] is the premier source for personalized entertainment information on the Web. It lets users explore categories of entertainment that include music, movies, books, events, and TV. eGenie is the showcase of Learning Sesame, which can learn from user's interests and dynam-

[6] http://auction.eecs.umich.edu
[7] http://ecommerce.media.mit.edu/Kasbah/
[8] http://jango.com
[9] http://egenie.opensesame.com

ically generate Web pages customized to the user's taste. *MORSE*[10] is a movie recommendation system that can suggest new movies based on user's previous rating record.

3. Internet. As an example, *Autonomy Intelligent Agents*[11] is a news filtering system that automatically compiles a personalized newspaper of interest to the user. A user can train an agent by first typing a short description, then the agent can enhance its knowledge of user's interests by using the new features learned from user's past experience. *LIRA* [4] developed at Stanford University stands for Learning Information Retrieval Agent. It can help users browse or "surf" the Web. The system observes how users rank various pages they visit. It then retrieves new pages on its own, and presents only those which it believes the user may want. *WebWatcher* [36] created at Carnegie Mellon University is a "tour guide" agent for the WWW. Given a description of a user's current interests (keywords), the WebWatcher accompanies the user from page to page as he/she browses the Web, and recommends the hyperlinks that it believes to be highly relevant to the user's interest. It learns the strategy for giving advice from the feedback it received during earlier tours.

Agents can work independently, known as *personal agents*, as well as work jointly together in an intellectual endeavor, known as *collaborative agents* [40].

[10]http://www.labs.bt.com/innovate/multimed/morse
[11]http://ultra.agentware.com/dailyme

1.2.1 Personal Agents

Personal agents are usually used to model users' long-term interests. They take user's interests, such as keywords, a brief description, bookmarks, and feedback, etc. as user's profile, and make predictions on user's behavior based on personal history. Figure 1.3 shows a typical personal agent. For example, *WBI*[12], IBM's Web Browser Intelligent agent, can personalize a web user's experience, and give a web server the ability to add or modify web content on the fly. *WebLearner* [59] maintains user's preferences as hotlist (for links that are interesting) and coldlist (for links that are not interesting), analyzes the document from each link, and recommends new pages that might interest the user. *Letizia* [45] is another Web browsing assistant. It keeps track of user's browsing process, explores links concurrently and automatically from the user's current position by using the knowledge inferred from the user's behavior, and attempts to suggest pages that are suited to user's interests.

1.2.2 Collaborative Agents

Collaborative agents interact and cooperate with each other to perform tasks on behalf of a user. Figure 1.4 shows a typical collaborative agent. The design of collaborative agents involves problem solving, communication, and coordination strategies for agents to maintain autonomy. They must also be able to efficiently use the available network bandwidth to communicate with other agents. Agent collaborations may involve heterogeneous or homogeneous groups of agents, and agents with similar

[12]http://www.networking.ibm.com/wbi/wbisoft.htm

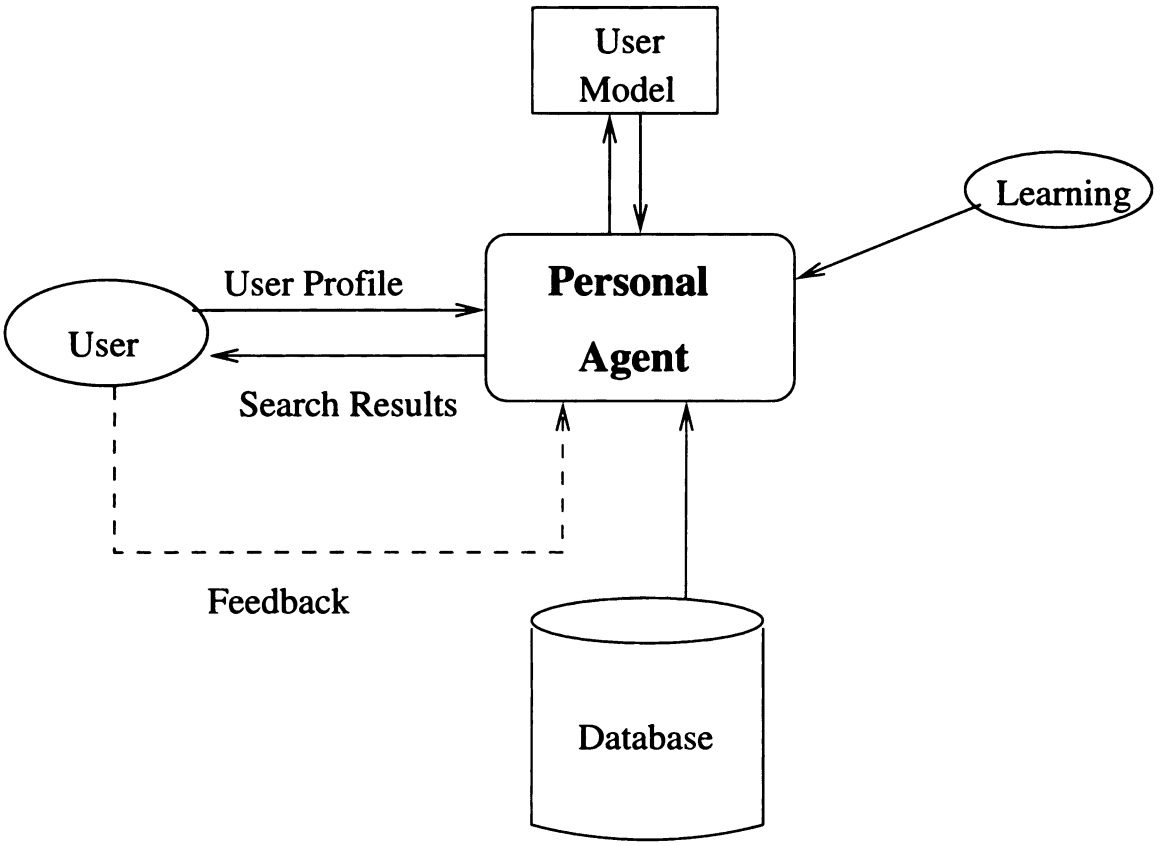

Figure 1.3: A typical personal agent

or differing goals, languages, and knowledge representation facilities. For example, *WiseWire*[13] uses a combination of content-filters, collaborative-filters, and learning agents to deliver interesting documents to users. It encodes the key conceptual material from documents contents, sifts out uninteresting pages, categorizes them, and recommends information rated highly by one user to other users with similar interests [38]. *Ginkgo*[14] developed at IBM is an agent-based learning system. It uses personal learning as well as collaborative learning to model a user's preferences, and predicts a user's behavior based on personal history, past experience, and on similar

[13]http://www.wisewire.com
[14]http://www.networking.ibm.com/iag/iaginkgo.htm

individuals' histories. It can learn from other agents and promote knowledge sharing. *Yenta*[15] under development at MIT, is a matchmaking system that aims at providing privacy-protected, distributed, automatic generation of clusters of users who are interested in similar topics [20].

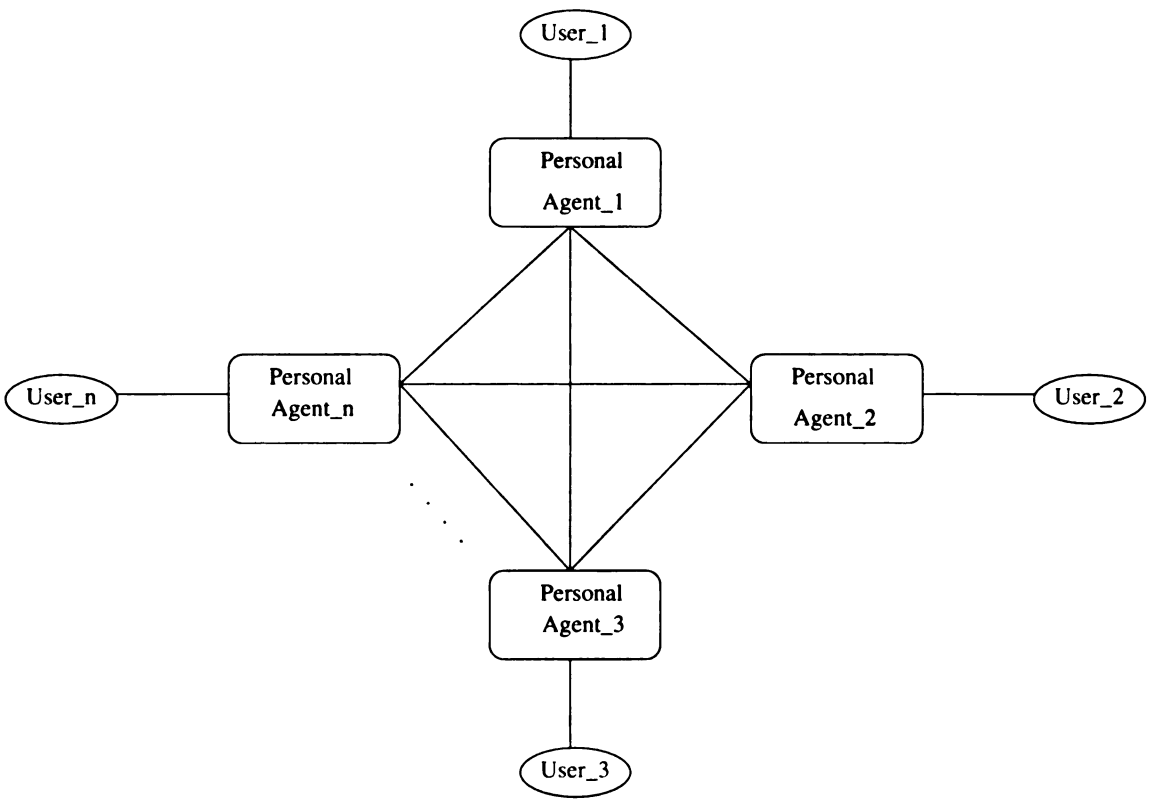

Figure 1.4: A typical collaborative agent

1.3 Motivation

Many tasks in information retrieval can be formulated as a document classification problem [43]. These tasks include:

[15]http://foner.www.media.mit.edu/people/foner/Yenta

1. Document Retrieval: Document retrieval is to select a set of documents from an indexed database, usually in response to some user queries. User queries could range from a few keywords to multi-sentence descriptions of an information item that is needed. A vast majority of retrieval systems currently in use range from simple Boolean systems (e.g., search engines) to systems using statistical or natural language processing (e.g., intelligent agents). A document retrieval system usually consists of four main phases: indexing, query formulation, comparison, and feedback [43].

2. Document Filtering and Routing: Also known as selective dissemination of information, is an information seeking system to sort through large volumes of dynamically generated information and present those documents to the user which satisfy a relatively stable and specific information need. It is similar to document retrieval, but emphasizes the dynamic environment and specific long-term interests [55]. It may involve indexing, user profiles modeling, adaptation of user profiles, matching, and social filtering etc. SIFT [82], FireFly[16] and WiseWire are some examples of available document filtering systems.

3. Document Categorization: It is the assignment of a document to pre-defined categories. A category label can be assigned to a document manually, such as in *Yahoo.* However, manual assignment of categories requires considerable human labor, time, and cost. An automatic document categorization system is desirable to reduce these costs substantially.

[16] http://www.firefly.com

4. Document Clustering: Classification is ambiguous in information retrieval. People often refer to clustering as a classification task. Clustering is to discover the underlying structure (categories) of a collection of documents based on some similarity measure. It is an unsupervised classification.

Text classification can also be incorporated with *push technology*, such as *PointCast Network*[17]. Push technology involves software that is loaded on the machines connected to the Internet. The software works in the background to compile Internet sources related to a user's selected areas of interest, and automatically delivers news and other specified information from the Internet to the user's desktop.

Text classification presents many challenges and difficulties. First, it is difficult to capture high-level semantics and abstract concepts of natural languages just from a few individual words. For instance, there are many ways to represent similar concepts (e.g., agent, softbot, robot, or bot), and the same word can represent different meanings (e.g., bank can either mean a financial institution or a river bank). Furthermore, semantic analysis, which is a major step in designing a natural language information retrieval system, is not well understood, although there are some techniques that have been successfully applied to limited domains [18]. Noise, high dimensionality (thousands of features), and variable length of content are some of the other undesirable characteristics of a huge number of documents on the Web. Due to these difficulties, there is a tradeoff between efficiency and accuracy of a classification system.

[17]http://www.pointcast.com

1.4 Problem Definition

We focus on supervised text classification in this thesis. A typical classification problem can be stated as follows: Given a set of labeled examples belonging to two or more classes (training data), classify a test sample to a class with the highest similarity. Document retrieval, routing and filtering systems, can often be viewed as two-class classifiers which label a document as relevant or non-relevant [59, 82, 38]. User feedback provides a set of training examples with positive and negative labels. A document is presented to the user if it is classified as the relevant class. In document categorization, we already have human indexed training data available, so a classifier is used to automatically assign a given previously unseen document to the appropriate class. Examples include deciding what newsgroup an article belongs to [79], what folder an email message should be directed to [13], or automatic coding of diagnoses in patient records [47].

In this thesis, we concentrate on document categorization, i.e., we deal with a multi-class classification problem. Figure 1.5 shows a block diagram of our text classification scheme. We report experimental results using news data from Yahoo web site, which are categorized into seven groups (business, entertainment, health, international, politics, sports, and technology), and Reuters-21578 newswire benchmark[18], which involves 135 topical categories. Each item of news is indexed manually by human experts (i.e, we have labeled training examples and "ground-truth" of test samples). The contributions of this thesis are as follows:

[18]http://www.research.att.com/~lewis

1. Our classification system is based on a multi-class classification system, rather than a two-class classification which is typical in the classification systems [44, 58, 71] reported in the literature.

2. We apply subspace method to text classification. It performs well on the Yahoo dataset, while poor on Reuters dataset.

3. We apply five different classification methods to our datasets: the naive Bayes classifier, nearest neighbor classifier, decision trees, subspace method, and artificial neural network method. These classifiers were also evaluated on different document representation schemes obtained by best feature selection, principal component analysis, linear discriminant analysis, and term-grouping in subspace.

4. We also investigate whether performance can be improved by a combination of different classifiers. Our experimental results indicate that a combination of multiple classifiers does not always improve classification accuracy compared to the best individual classifier. The adaptive classifier combination method introduced in this thesis outperforms simple voting and dynamic classifier selection.

1.5 Thesis Outline

The outline of this thesis is as follows. Chapter 2 briefly reviews some relevant work on text document classification. Chapter 3 describes the five text classification methods we have used in our experiments, and investigates combinations of multiple

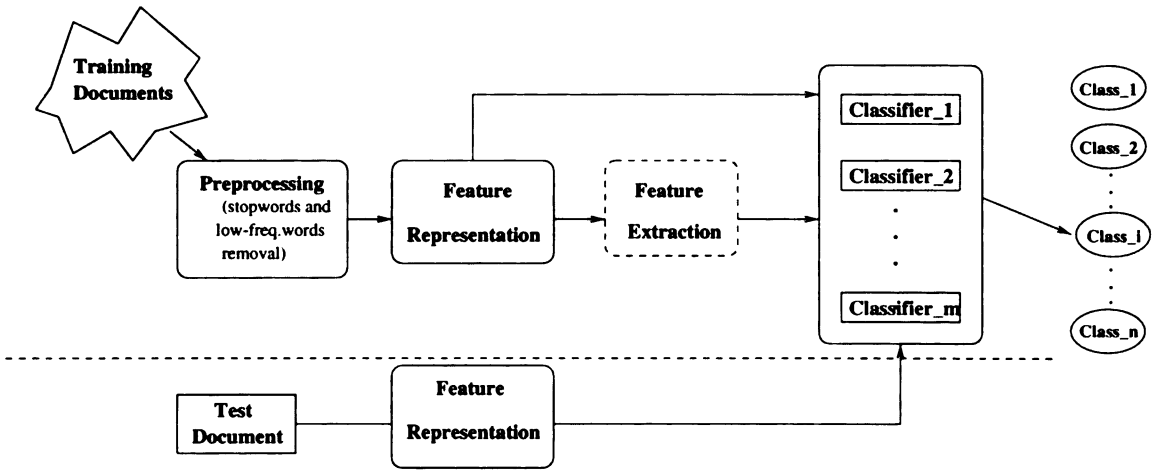

Figure 1.5: A block diagram of our text classification scheme

classifiers. Dimensionality reduction techniques are introduced in chapter 4. We present our experimental results in chapter 5. Finally, we conclude with discussions and future work in chapter 6.

Chapter 2

Literature Review

In the past few decades, the availability of cheap and effective storage devices and information systems, and the rapid growth of Internet has created an immense information bottleneck. Most commercial information retrieval systems (e.g., search engines) still rely on conventional inverted index and Boolean querying techniques. An inverted index catalogs a collection of objects in their textual representation. Given a set of documents, keywords and other attributes (possibly including relevance ranking) are assigned to each document. Inverted index is the list of keywords and links to the corresponding document. Index is sorted on the key values to allow rapid searching for a particular key value. But they often produce less than satisfactory results. Probabilistic models have been used to improve the performance of information retrieval, filtering, routing and categorization. Since the late 1980s, knowledge-based techniques have been used extensively by information science researchers. These techniques have attempted to capture users and information specialists' domain knowledge, classification scheme knowledge, effective search strategies, and query refinement heuristics in

the design of information systems [11]. More recently, machine learning approaches, such as symbolic, inductive learning methods, artificial neural networks, and genetic algorithms have also been utilized [10].

In this chapter, we briefly review several popular techniques used in document retrieval, filtering, routing, and categorization. We pay more attention to classification methods. We also review some feature selection and extraction techniques that are commonly used in a text document classification system, because of the high dimensionality of the feature vector (usually in the order of several thousand) often encountered in this problem.

2.1 Classification Techniques

There are several ways to design a classifier, from typical techniques in information retrieval, such as TF-IDF, to pattern recognition and machine learning techniques, such as decision trees, naive Bayes classifier, nearest neighbor, artificial neural network and genetic algorithms. They all have the same goal: given a set of training samples along with category labels, classify a test sample to one or several most appropriate categories.

2.1.1 TF-IDF

TF-IDF is one of the most successful techniques in information retrieval. A document is represented as a vector of weighted terms. Terms that appear frequently in one document (TF=term frequency), but rarely occur outside the document (IDF=inverse

document frequency) have larger weights. The weights can be normalized according to the length of the documents. In Syskill & Webert [58] and NewsWeedeer [38] systems, the TF-IDF vectors of all examples of one class are averaged to get a prototype vector for the class. A test document is assigned to the class that has the highest cosine similarity measure, i.e., smallest angle between the TF-IDF vector of the test pattern and the class prototype vector.

2.1.2 Probabilistic Methods

Probabilistic Methods have been widely used in information retrieval. Naive Bayes classifier has been demonstrated to be a powerful classifier in several domains [16, 44, 58]. Domingos et al. [16] conducted an empirical study on 28 domains, a varied selection of databases from the UCI repository[1]. They showed that naive Bayes classifier was more accurate than a decision tree (C4.5) in 16 domains. Lewis et al. [44] compared their ProbBayes method with a decision tree classifier (DT-min10) on two data sets. One of the data sets was Reuters-22173 newswire stories. These stories have been manually indexed using 135 financial topic categories. The second data set consisted of 1,500 documents from the U.S. foreign Broadcast Information Service that had been used in the MUC-3. ProbBayes essentially uses Bayesian rule to estimate $P(C_j = 1|D)$, the a *posteriori* probability of category c_j given the document. A 2-class classification was applied on all the categories for one test pattern. For Reuters-22173 data set, both Bayes method and the decision tree method gave similar

[1] http://www.ics.uci.edu/ mlearn/MLRepository.html

classification performance. The best break-even value (where the recall is equal to the precision) is about 65%; for MUC-3 data set, the DT-min10 performed slightly well, the best break-even value is about 50%.

Bayesian networks, also known as inference networks, are generalizations of naive Bayes classifier. A Bayesian network is a directed acyclic graph representation of the joint probability distribution for a set of variables, where each node in the graph represents a variable, and each edge represents the correlation between the two variables [32]. Bayesian networks were originally designed to encode the uncertain knowledge of an expert [60], and in recent years, they have been applied to a variety of fields, such as expert systems, diagnosis engines, information retrieval and decision support systems [25]. In [75, 76, 7], Turtle and Croft et al. used Bayesian network for information retrieval. The Bayesian network-based retrieval systems consist of two parts: a document network and a query network. In document net, nodes representing documents are connected to nodes representing terms. The query net is constructed where terms in the query are connected to nodes representing how the terms should be combined. To perform a retrieval, the system connects these two networks together. Given the prior probabilities associated with the documents and the conditional probabilities with internally connected nodes, the a *posteriori* probability of each document in the collection, given the query, can be computed using Bayesian rule. The system then ranks the documents by this probability.

The underlying assumption of the naive Bayes approach is that, for a given class, the probabilities of terms occurring in a document are independent of each other. On the other hand, Bayesian inference networks reflect the dependence between the

terms. Several studies have attempted to relax the strong independence assumption in naive Bayes classifier by incorporating Bayesian inference networks [66, 24].

2.1.3 Nearest Neighbor Classifier

For a given test document, the nearest neighbor classifier can be viewed as an attempt to estimate the a *posteriori* probability of a category from the training patterns. This rule assigns a test document to the category of its nearest training pattern. A K-nearest neighbor classifier assigns a test document to the category which is most frequently present among its K nearest neighbors. Its simplicity and ease of implementation has made it popular in information retrieval systems. For example, Pazzani et al. [58] applied the nearest neighbor rule on their Syskill & Webert system, an interesting web page identification agent that learns from a user's profile. Weiss [79] et al. showed that a classification-based information retrieval method using the nearest neighbor classifier performed better that the TF-IDF prototype method on USENET newsgroups datasets.

2.1.4 Symbolic Learning Methods

Symbolic learning techniques have been extensively studied during the last few decades [8, 10]. Among these methods, decision trees, such as ID3 [61], and C4.5 [63] are the most popular classifiers that have been used in information retrieval fields, such as medical record classification [3], Web page selection [58], and news stories categorization [44]. Decision trees classify patterns by starting from the root of the

tree, and then traversing down to the leaves. Leaf nodes represent categories, while interior nodes represent attributes. When constructing a tree, attributes are selected based on their discrimination abilities.

Another type of symbolic learning is rule-based method. Rules can be automatically induced from decision trees [63], from neural networks [14], and they can also be learned from examples [12, 13, 50, 2].

2.1.5 Artificial Neural Networks

Artificial neural networks, e.g., multilayer perceptrons and radial basis networks, seem to fit well with conventional retrieval methods in information science [10], such as the vector space model [67]. A typical multilayer feedforward network [31, 48] learns the weights for its interconnections by using a gradient descent method to minimize the squared error between the network output values and the target values for these outputs. Schutze et al. [71] used neural networks trained by backpropagation for document routing problem. Two types of neural networks were used in their work: one is a linear neural network which consists of only input and output units, and another is a non-linear neural network which has one hidden layer with three units. They demonstrated that neural networks, linear discriminant analysis, and logistic regression classifiers perform $10 - 15\%$ better than relevance feedback via Rocchio expansion for TREC-2 and TREC-3 routing tasks. In another similar study [15], LSI (Latent Semantic Indexing)-based neural networks were employed for classification of 10 categories of TREC-1 news wire data. They did their experiments in two

configurations: (i) single sensor neural net, where the inputs are based on LSI alone, and (ii) two sensor neural net, which uses additional 10 keywords as inputs. Their experimental results showed that neural network methods perform 4% to 20% better than single LSI method. Artificial neural network techniques have also been employed in commercial intelligent agents, such as the wisewire collaborative filtering engine[2].

2.1.6 Genetic Algorithms

Another type of learning methods, called genetic algorithms (GA), are also popular in information science. Chen [10] surveyed several implementations of genetic algorithms in information retrieval. Sheth [73] used a genetic algorithm to build a personal information filtering system, called Newt. During the learning phase in Newt, a genetic algorithm was used to model the population characteristic and behavior in response to changing user interests. By using crossover and mutation, new members were added into the population, while unfit members were removed during each generation.

2.2 Dimensionality Reduction

Document classification is often characterized by the high dimensionality (often thousands of features) of the associated feature space and a relatively small number of training samples. The feature set is often "noisy" and may contain redundancy [43], which may lead to the problem of "curse of dimensionality" [33]. Moreover, large

[2]http://www.wisewire-corp.com/indexprod.html

number of features result in larger computational and storage complexity. There are several ways to reduce the feature set size. Traditional feature selection approaches in pattern recognition, such as sequential forward/backward selection and "plus l-take away r" selection [35] may perform better, but they are often computationally demanding. In this thesis, we briefly review two most popular feature reduction methods in information retrieval: best feature selection and latent semantic indexing (LSI).

2.2.1 Best Feature Selection

In Best feature selection, the goal is to find the best subset of features from the entire feature space. Salton et al. [69] provided a weighting scheme, such as TF-IDF, for terms in the document collection. Schutze et al. [71] applied a χ^2-measure of dependence to a relevant and non-relevant contingency table to indicate the importance of the features. Mladenic [49] suggested the use of mutual information to assign weights to the terms in the documents. Koller et al. [37] developed an efficient algorithm for feature selection based on information theory. Their empirical results indicated that the algorithm can effectively handle datasets with a large number of features (up to 1675 features).

2.2.2 Latent Semantic Indexing

Latent semantic indexing (LSI) method developed at Bellcore [51] is a vector space information retrieval method which has been widely adopted in information filtering and retrieval systems [18]. In LSI method, introduced in [51, 18], the singular value

decomposition (SVD) is performed on a term-document matrix, which is formed from a set of training patterns. Recall that each entry in the term-document matrix indicates the term-frequency in the corresponding document. Only large singular values are preserved. The resulting singular vector and singular value matrices are used to project term frequency vectors for documents and queries onto a reduced feature subspace where semantic relationships in the term-document matrix are preserved. In a query system, documents can be ranked according to their similarity measure to a query by using these projected subspace vectors. This technique has been successfully used in Salton's SMART system [68]. LSI has also been used for feature reduction in document classification systems [71, 15].

Chapter 3

Classification Algorithms

In this chapter, we introduce different classifiers used in our study of text document classification. We formulate our text classification as a multi-class classification problem, i.e., assign a test pattern to one of the pre-defined categories (see Figure 1.5). The vector space model [67] is used to represent document features. During the preprocessing stage, a simple lexical analysis is performed to convert an input stream of characters into a stream of "useful" words.

3.1 Preprocessing

Web documents are written in HTML language. We use the following preprocessing steps to construct the so called *indexing file*, which contains those words that may be helpful in text classification.

1. Use an HTML parser to filter out HTML tags.

2. Convert all characters to lower case.

3. Remove digits and punctuations.

4. Screen out words in a *stoplist.*

5. Remove low frequency words (which only occur once in the training examples).

Words in the stoplist are known as *stopwords*, which are the most frequently occurring words in English, such as "the", "some", "of", etc. These words are not considered helpful in retrieval. The stoplist we choose is the one given in [21].

Other preprocessing steps, like *stemming* [22] can be applied to further reduce the size of indexing files. An effective stemming is helpful in retrieval and classification, because the words with the same stem should have similar meaning. However, over-stemming [22] can cause original unrelated words to be represented as the same stem, in which case it may actually degrade the performance of a classification system. For example, "international" and "internal" are both stemmed as "intern", but they originally have different meanings.

3.2 Feature Representation

The feature space of documents can be represented by the indexing files generated above. We adopt the commonly used "bag-of-words" document representation scheme (vector space model), in which we ignore the structure of a document and the order of words in the document. The *word-list* $\mathcal{W} = (w_1, ..., w_d)$ in the training set consists of all the distinct words (terms) that appear in the training indexing files. Typically, there can be several thousand features in document classification (the number of

commonly used English words is between 20,000 to 50,000). Given a document $\mathcal{D}$, its feature (term) vector is represented by $\mathcal{T} = (t_1, ..., t_d)$ constructed from $\mathcal{W}$. The value of each component of $\mathcal{T}$ could be either binary (a value of 1 indicates that the corresponding word appeared in the document) or an integer representing the number of times the corresponding word was observed. Some training examples and the corresponding indexing files of each Yahoo news group are shown in Appendix A.

3.3 Classification Algorithms

In the following sections, we describe the naive Bayes classifier, nearest neighbor classifier, decision tree classifier, and artificial neural network classifier used in our study. We also introduce our use of the subspace method for text document classification.

3.3.1 Naive Bayes Classifier

The naive Bayes classifier [48], also known as simple Bayes classifier, has been successfully used in text classification [1, 58]. Let $\mathcal{C} = (c_1, ..., c_m)$ be the set of m document classes. Given a new unlabeled document $\mathcal{D}$, its word-list $\bar{\mathcal{W}} = (w_1, ..., w_{d'})$ (defined in the same way as the word-list for the training set), and its corresponding binary feature (term) vector $\bar{\mathcal{T}} = (t_1, ..., t_{d'})$, $\mathcal{D}$ is described by the conjunction of a set of attribute values t_i, $i = 1, ..., d'$. Using Bayes rule, the a *posteriori* density of class c_j, given document $\mathcal{D}$ can be written as:

$$P(c_j|\mathcal{D}) = \mathcal{P}(\lfloor_1|\sqcup_\infty, ..., \sqcup_{\lceil'}), \tag{3.1}$$

where

$$P(c_j|t_1,...,t_{d'}) = \frac{P(t_1,...,t_{d'}|c_j)P(c_j)}{P(t_1,...,t_{d'})}. \tag{3.2}$$

The underlying assumption of the naive Bayes approach is that, for a given class c_j, the probabilities of words occurring in a document are independent of each other, i.e.,

$$P(t_1,...,t_{d'}|c_j) = \prod_{i=1}^{d'} P(t_i|c_j). \tag{3.3}$$

The naive Bayes approach assigns $\mathcal{D}$ to a class c^*_{NB} as follows:

$$c^*_{NB} = argmax_{c_j \in \mathcal{C}} P(c_j) \prod_{i=1}^{d'} P(t_i|c_j), \tag{3.4}$$

where $P(c_j)$ is the a *priori* probability of class c_j and $P(t_i|c_j)$ is the conditional probability of word w_i, given class c_j. Both $P(c_j)$ and $P(t_i|c_j)$ are estimated from the training data.

When the size of the training set is small, the relative frequency estimates of probabilities, $P(t_i|c_j)$, will not be reasonable. For example, if a word never appeared in the given training data, its relative frequency estimate will be zero. So, we applied the Laplace's law of succession [64] to estimate $P(t_i|c_j)$. It is essentially a Bayesian estimate of the multinomial parameters. The Bayesian estimate of $P(t_i|c_j)$ is given as:

$$P(t_i|c_j) = \frac{n_{ij} + 1}{n_j + k}, \tag{3.5}$$

where n_j is the total number of word occurrences in class c_j, n_{ij} is the number of

occurrences of word w_i in class c_j, and k is the vocabulary size of the training set. This Bayesian estimation is based on a uniform prior assumption, i.e., probabilities of various word occurrences in class c_j are equally likely. A detailed derivation of this estimate can be found in Appendix B.

3.3.2 Nearest Neighbor Classifier

The nearest neighbor (NN) decision rule assigns an unlabeled document $\mathcal{D}$ to the document class c_j if the training pattern closest to $\mathcal{D}$ is from class c_j. A variation of nearest neighbor rule is the K-nearest neighbor (KNN) rule, which first finds the K nearest neighbors of $\mathcal{D}$ among the training patterns, and then uses a voting scheme to assign a category label to $\mathcal{D}$. The NN/KNN rule can be viewed as an attempt to estimate the a posteriori probabilities $p(c_j|\mathcal{D})$ from training patterns. The number of nearest neighbors (K) should be small compared to the total number of training samples. A rule of thumb is that K should be proportional to $\sqrt{n}$. The NN rule is used in our text classification for simplicity. A number of algorithms can be adopted to reduce the complexity of nearest neighbor search, such as nearest neighbor editing [17], reduced nearest neighbor rule [26], and an effective algorithm for nearest neighbor search in high dimensions [52].

We use the *TF-IDF* (*TF* is the term frequency in a document, and *IDF* is the inverse document frequency) weighting scheme and use the cosine similarity [69] instead of Euclidean distance to measure the similarity of the documents. Given two documents $\mathcal{D}_1$ and $\mathcal{D}_2$, their corresponding weighted feature vectors are represented

as $\mathcal{T}_1 = (t_{1i}\delta_i)_{i=1}^d$ and $\mathcal{T}_2 = (t_{2i}\delta_i)_{i=1}^d$, where δ_i is the weight of word w_i (based on *TF-IDF*). The similarity between $\mathcal{D}_1$ and $\mathcal{D}_2$ is then defined as:

$$S(\mathcal{D}_1, \mathcal{D}_2) = \frac{\mathcal{T}_1^T \mathcal{T}_2}{\|\mathcal{T}_1\| \|\mathcal{T}_2\|}, \tag{3.6}$$

where $\|\cdot\|$ denotes the norm of the vector.

3.3.3 Decision Tree Classifier

Decision trees are one of the most widely used inductive learning methods. Their robustness to noisy data and their capability to learn disjunctive expressions seem suitable for document classification. One of the most well known decision tree algorithm is ID3 [61], and its successor C4.5 [63] and C5. It is a top-down method which recursively constructs a decision tree classifier. The interior nodes of the tree are associated with specific attributes (e.g., terms in document collection), and leaves of the tree represent specific categories. At each level of the tree, ID3 selects the attribute that has the highest *information gain.* Information gain is simply the expected reduction in entropy caused by partitioning the training examples according to this attribute [48]. Given a collection of training samples Tr, information gain for an attribute W is defined as [48]:

$$Gain(Tr, W) = Entropy(Tr) - \sum_{v \in Values(W)} \frac{|Tr\_v|}{|Tr|} Entropy(Tr\_v) \tag{3.7}$$

$$Entropy(Tr) = \sum_{i=1}^{m} -p_i log_2 p_i \tag{3.8}$$

where $Values(W)$ is a set of values which W can take, Tr_v is the set of training samples for which $Value(W) = v$, and p_i is the proportion of samples that belong to class i.

In summary, ID3 searches a complete hypothesis space, and uses the statistical properties (e.g., information gain) of all training examples at each step in the search. This makes it more robust to noisy training data. Furthermore, it uses *reduced-error pruning* [62] to avoid overfitting in decision tree learning. Mitchell gives an analysis of ID3 decision tree in more detail in [48].

For our experiments, we chose the C5 decision tree package, since it has many nice features over its predecessor, ID3 and C4.5. For example, the *rulesets* used in C5 are more accurate, faster, and require less memory [1]. Furthermore, *adaptive boosting* [23] is incorporated into the software. The basic idea of boosting is to generate n ($n > 1$, n is specified by the user) classifiers (either decision trees or rule sets) instead of one. The ith classifier is constructed by examining the errors made by the $(i-1)th$ classifier. When a new document is to be classified, a voting scheme based on the n classifiers is used to determine the final class of the document. We use the adaptive boosting option in the decision tree classifier in our experiments.

3.3.4 Multilayer Feed-Forward Network Classifier

An important class of neural networks, namely, adaptive layered networks (e.g., multilayer perceptrons and radial basis networks) have been widely applied to diverse pat-

[1] http://www.rulequest.com/see5-comparison.html

tern classification domains with some success. They also seem to fit well with conventional retrieval models, such as vector space model in information science [10, 71, 15]. A three-layer feed-forward network with two hidden layers and an output layer is used in our text classification. Figure 3.1 shows the architecture of our feed-forward neural network. Text document classification usually involves thousands of features. It is

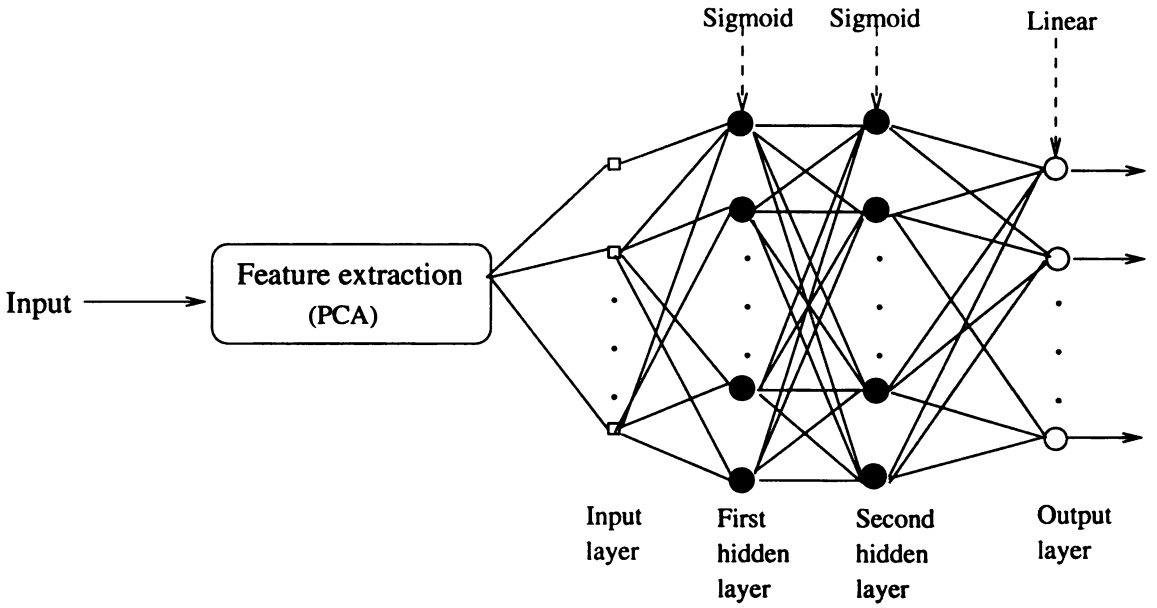

Figure 3.1: The architecture of our adaptive feed-forward neural network.

difficult for neural networks to handle such a large input dimension. The principal component analysis method (see section 4.2) is used on the vector space model to reduce the feature set size. The projected features in the subspace are used as input to the three-layer feed-forward network.

Webb et al. [78] illustrated why a nonlinear adaptive feed-forward layered network with linear output is capable of performing classification tasks well. They demonstrated that, if the weights are adjusted to minimize the total mean square output error, then the nonlinear transformation of the hidden layers also maximizes

the *network discriminant function* (C) [78]:

$$C = Tr\{S_B S_T^+\} \tag{3.9}$$

where S_B is the (weighted) between-class covariance matrix, and S_T^+ is the pseudo-inverse [29] of the total-class covariance matrix S_T.

For a one-to-m target coding scheme,

$$t_{cp} = \begin{cases} 1, & \text{if input pattern } p \text{ is in class } c \\ 0, & \text{otherwise.} \end{cases} \tag{3.10}$$

This coding scheme results in a weighted between-class covariance matrix S_B. We choose to use an alternative target coding scheme [78],

$$t_{cp} = \begin{cases} \frac{1}{\sqrt{n_c}}, & \text{if input pattern } p \text{ is in class } c, \text{ where} \\ & n_c \text{ is the number of patterns in class } c \\ 0, & \text{otherwise} \end{cases} \tag{3.11}$$

in which case S_B is the conventional between-class covariance matrix, and it is argued in [78] that, for multi-class classification problems, this target coding scheme compensates for un-balanced class memberships in training data. By performing a nonlinear transformation of the data into a space where the classes maybe more separated as determined by the between-class and total-class covariance matrices, the subsequent linear transformation to minimize the mean square target error could perform better than on the original data.

The neural network uses backpropagation algorithm [31, 48] for training. Basically, the algorithm performs stochastic gradient descent to minimize the mean square error between the network output and their corresponding target values. The activations of input pattern are propagated forward through the net, while the errors are backpropagated to update the weights in order to minimize the mean square target error.

Traditionally, the weights in the feed-forward neural network trained by a backpropagation algorithm are initialized randomly. This may involve large training epochs and the weights may not converge in some cases. Instead, we initialize the weights based on the eigenvalues of the covariance matrix of the input pattern matrix, which helps to speed up the convergence [70].

3.3.5 Linear Subspace Method

The subspace model [56] decomposes a given feature space into m sub-regions of lower dimensionality (subspaces), where each region is a representative feature space for its corresponding pattern class $c_i, i = 1, ..., m$. A test document is classified based on a comparison of its compressed representation in each feature space with that of different classes.

We apply this model to document classification as follows. Suppose we have m document classes $\mathcal{C} = (c_k)_{k=1}^{m}$. Class c_k is represented by a subspace $\mathcal{L}_k$ of cardinality d_k. Let $\mathcal{T} = (t_i)_{i=1}^{d}$ denote the term-vector in the original d-dimensional feature space, corresponding to the word-list of the training set $\mathcal{W} = (w_i)_{i=1}^{d}$. Let the word-list of

the subspace $\mathcal{L}_k$ be denoted as $\mathcal{W}_k = (w_i^k)_{i=1}^{d_k}$, where w_i^k are the words observed in class c_k. Given a vector $\mathcal{T}$ in the original feature space, the weighted projection Π_k of vector $\mathcal{T}$ on the subspace $\mathcal{L}_k$ is defined as:

$$\mathcal{T}_\mathbf{k} = \mathbf{\Pi_k}(\mathcal{T}) = \mathcal{H}_\mathbf{k}\mathcal{T}, \tag{3.12}$$

where $\mathcal{H}_k = (h_{ij})_{d_k \times d}$ is a $d_k \times d$ matrix, and the ith row corresponds to the ith component of the word-list $\mathcal{W}_k$ in the subspace $\mathcal{L}_k$, while the jth column is the jth component of the word-list $\mathcal{W}$ in the original feature space. The elements h_{ij} are calculated as follows:

$$h_{ij} = \begin{cases} \delta_j^k, & \text{when the term } w_i^k \text{ is the same as the term } w_j \\ 0, & \text{otherwise,} \end{cases} \tag{3.13}$$

where δ_j^k is the weight of term w_j^k in subspace $\mathcal{L}_k$. We define δ_j^k as:

$$\delta_j^k = \frac{CLASSFREQ_{jk}}{log_2(DOCFREQ_j + 1)}, \tag{3.14}$$

where $CLASSFREQ_{jk}$ denotes the ratio of the number of documents in which the term w_j occurred in class c_k to the number of documents in c_k, and $DOCFREQ_j$ represents the ratio of the number of documents in all those classes in which the term w_j occurred to the size of the training set.

The Euclidean vector norm of $\mathcal{T}_k$ is $\|\mathcal{T}_k\| = \sqrt{\mathcal{T}_k{}^T \mathcal{T}_k}$. For a new document $\mathcal{D}$, the subspace decision rule classifies $\mathcal{D}$ to the class on whose subspace its term-vector $\mathcal{T}$

has the largest projection in terms of the Euclidean vector norm.

3.4 Combination of Different Classifiers

A number of studies have shown that combining different classifiers can improve the classification accuracy [80, 39, 28]. Larkey et al. [39] applied weighted linear combinations of different classifiers to the medical document domain, where the weights were assigned by the user. Another CMC approach is dynamic classifier selection (DCS) [80, 28], where a single classifier is selected which has the highest local accuracy in small regions of feature space surrounding the test sample presented to the system. We investigated three different combination approaches: simple voting, DCS, and our own approach of adaptive classifier combination (ACC).

3.4.1 Simple Voting

For each test document, classify it to class c_i, where a majority of the classifiers individually assign the test document to class c_i.

3.4.2 Dynamic Classifier Selection (DCS)

We have implemented a version of DCS described in [80, 28]. For a test document $\mathcal{D}$, we use the k-nearest neighbor approach to find the neighborhood of $\mathcal{D}$ and the "leave-one-out" method [17] is applied on the training data to find the local accuracy in the neighborhood of $\mathcal{D}$. We used the "soft" measure [28] of the local accuracy, where the weight of each neighbor of $\mathcal{D}$ is the cosine similarity measure between $\mathcal{D}$

and that neighbor.

3.4.3 Adaptive Classifier Combination (ACC)

Instead of selecting the best classifier with the highest local accuracy for a test document, we assign the document to class c_i, which is the class identified by the classifier that has the highest local accuracy among all the classifiers. The outline of our ACC algorithm, given n classifiers is described as follows:

1. For a test document $\mathcal{D}$, find the neighborhood of $\mathcal{D}$, $\mathcal{NB}(\mathcal{D}) = (x_1, ..., x_K)$, $x_i \in Trainning\_Set$, using the K-nearest neighbor algorithm.

2. Denote the classification results for $\mathcal{D}$ by n classifiers as $\bar{\mathcal{C}} = (\bar{c_1}, ..., \bar{c_n})$, $\bar{c_j} \in \{c_1, ...c_m\}$.

3. For each class $c_j \in \bar{\mathcal{C}}$, calculate $Acc^j_{loc} = \sum_{s=1}^{n} \sum_{i=1}^{k} W_i P_s(c_j | x_i \in c_j)$, where $P_s(c_j | x_i \in c_j)$ is the local accuracy of a neighborhood pattern x_i, i.e. the *a posteriori* probability that x_i belongs to class c_j. The local accuracy of each x_i can be obtained by using the "leave-one-out" method on the training data, and W_i is the cosine similarity measure between pattern x_i and $\mathcal{D}$, which is defined as:

$$W_i = Cosine(x_i, \mathcal{D}) = \frac{\sum_{k=1}^{t}(Term_{ik} \times Term_k)}{\sqrt{\sum_{k=1}^{t}(Term_{ik})^2 \times \sum_{k=1}^{t}(Term_k)^2}}, \tag{3.15}$$

where t is the dimensionality of the feature vector, $Term_{ik}$ is the value of term k in document x_i, and $Term_k$ is the value of term k in the given document $\mathcal{D}$.

4. Classify $\mathcal{D}$ to class c_κ, where $\kappa = argmax_j(Acc_{loc}^j)$.

Chapter 4

Dimensionality Reduction

Document classification is often characterized by high dimensionality of the associated feature space and a relatively small number of training samples. The increase in dimensionality results in an increase in both computational and storage complexities. Furthermore, we must also guard against the potential problems of "curse of dimensionality" [33]. We study four feature dimensionality reduction approaches: feature selection, feature extraction (principal component analysis and linear discriminant analysis), and term grouping in subspace.

4.1 Best Feature Selection

One way to reduce the number of features in document classification is to select a subset of the best terms from the entire feature space. In this paper, we use the *individual best features* approach, where terms are sorted by their weights in a descending order, and the top n terms with the highest weights are selected. We

use mutual information (i.e., information gain, see section 3.3.3) as suggested in [49] to assign weights to the terms. Traditional feature selection approaches in pattern recognition, such as sequential forward/backward selection and "plus l-take away r" selection [35] may perform better, but they are often much more expensive in terms of computational cost. This is a major consideration in high dimensional feature spaces encountered in document classification problems.

Another method of dimensionality reduction is to map original measurements into a more effective lower dimensional subspace. Each new feature is a combination of the original features. This is the so called feature extraction method, which includes principal component analysis and linear discriminant analysis which we have used in our study.

4.2 Principal Component Analysis

Principal component analysis (PCA) (also called K-L transform) is a commonly used linear projection method [34]. It projects the original data vector (with dimension d) on the coordinate axes having the dimension p (usually $p < d$). This minimizes the mean-square error between the original data and the new representation with p eigenvectors of the covariance matrix. Using the vector space representation scheme in document classification, let $\mathcal{T}_1, ..., \mathcal{T}_N$ denote the N d-dimensional training vectors, while their normalized vectors with zero-mean are denoted as $\mathcal{T}_1^*, ..., \mathcal{T}_N^*$. Let the p basis vectors, $\mathbf{e_1}, ..., \mathbf{e_p}$ be a set of orthonormal vectors that best describe the distribution of documents in the p-dimensional subspace (eigenspace), $p \leq d$. It can

be shown that the "best" set of basis vector correspond to the eigenvectors of the covariance matrix Σ. The first basis vector e_1 corresponds to the largest eigenvalue of Σ, the second basis vector e_2 corresponds to the second eigenvalues of Σ, and so on.

With the p-dimensional eigenspace defined, training vectors, $\mathcal{T}_1^*, \cdots, \mathcal{T}_N^*$, can be represented as a set of p-dimensional feature vectors, $\xi_1, \cdots, \xi_p$:

$$\xi_k = \mathbf{e}^T \mathcal{T}_i^*, \ \mathbf{i} = 1, \cdots, N, \tag{4.1}$$

where $\mathbf{e} = (\mathbf{e_1}, ..., \mathbf{e_p})$.

The sum of the first p eigenvalues is the "variance" retained in the subspace, while the sum of all the d eigenvalues is the "variance" in the original pattern space [34]. We can choose p such that $\sum_{i=1}^{p} \lambda_i / \sum_{i=1}^{d} \lambda_i \geq \nu$, where ν is a user-specified value representing the desired "variance" retained in the p-dimensional subspace.

4.3 Discriminant Analysis

Linear discriminant analysis (LDA) also attempts to project patterns onto a lower dimensional space than the original space [34]. Compared to PCA which projects patterns to a p-dimensional space whose orthogonal basis are the p eigenvectors corresponding to the p largest eigenvalues of the covariance matrix of the training samples, LDA projects patterns to $(m-1)$-dimensional space using the basis set $(\mathbf{e_1}, ..., \mathbf{e_{m-1}})$ computed from $\varphi_W^{-1}\varphi_B$, where m is the number of categories, φ_W is the within-class

scatter matrix and φ_B is the between-class scatter matrix. Let $\mathbf{x_{ki}}$ be the vector of ith document in category k, $i = 1, ...N_k$, $\mu_\mathbf{k}$ is the sample mean of class k, μ is the overall mean of the training samples, and $N = (N_1 + ... + N_k)$ is the total number of documents in the training set. Then according to [34, 57], the within-class scatter matrix φ_W and the between-class scatter matrix φ_B are defined as follows:

$$\varphi_W = \frac{1}{N}\sum_{k=1}^{m}\sum_{i=1}^{N_k}(x_{ki} - \mu_k)(x_{ki} - \mu_k)^T \tag{4.2}$$

$$\varphi_B = \frac{1}{N}\sum_{k=1}^{m}(\mu_k - \mu)(\mu_k - \mu)^T. \tag{4.3}$$

The basis set $(\mathbf{e_1}, ...\mathbf{e_p})$ is a set of eigenvectors corresponding to the $(m-1)$ largest eigenvalues of matrix $\varphi_W^{-1}\varphi_B$, which satisfy:

$$\begin{cases} e_i^t e_j = 1 & \text{if i=j,} \\ e_i e_j = 0 & \text{otherwise,} \end{cases} \tag{4.4}$$

where Σ is the covariance matrix of the training samples.

Since the rank of the matrix $\varphi_W^{-1}\varphi_B$ is at most $m-1$, the number of coordinates of the projected space is limited by the number of document categories. Foley and Samon [19] proposed an optimal discriminant plane for the 2-class problem under the orthonomality condition of coordinate axes. Okada et al. [57] generalized it to multi-class problem, called the orthonormal discriminant analysis (ODA), where the number of features being extracted are only constrained by the original feature dimensionality d. Hamamoto et al. [30] proved that "the ODA method is more powerful than LDA in terms of the Fisher criterion".

4.4 Term Grouping in Subspace

The occurrences of different words in documents are usually not independent; there are correlations between words in a group of documents. Wulfekuhler et al. [81] applied K-means clustering within each document category to find clusters of words for that document class. To capture the correlation of all word-pairs, we construct a bigram matrix for each document class k. Let $\mathcal{W}_k = (w_1^*, ..., w_{d_k}^*)$ be a d_k-dimensional word-list in subspace $\mathcal{L}_k$. The bigram matrix $\mathcal{B}_k = (m_{ij})_{d_k \times d_k}$ is a $d_k \times d_k$ matrix with m_{ij} representing the number of documents from class c_k in which both the terms w_i^* and w_j^* jointly appear. A complete-link hierarchical clustering algorithm is applied to the proximity matrix $\mathcal{B}_k$. The resulting dendrogram (tree) is then cut into p term-groups, where p is a user-specified parameter. Let the term-groups be denoted as $\mathcal{T}_k^* = (h_1, ..., h_p)$, where $h_i \bigcap h_j = \emptyset, i \neq j$ and $h_i = (w_{i1}^*, ...w_{in_i}^*)$, $w_{ij}^* \in \mathcal{W}_k$. Given a vector $\mathcal{T}_k$ in subspace $\mathcal{L}_k$, the projection of $\mathcal{T}_k$ to p-dimensional space, $\Xi = (\xi_1, ..., \xi_p)$ is defined as

$$\xi_i = (\mathcal{T}_k^T \mathcal{H}_i)\mathbf{I}, \tag{4.5}$$

where $\mathcal{H}_i = (h_{uv})_{n_i \times d_k}$ is defined as:

$$h_{uv} = \begin{cases} 1 & \text{when the term } w_{iu}^* \text{ is the same as the term } w_v^* \\ 0, & \text{otherwise,} \end{cases} \tag{4.6}$$

and $\mathbf{I}$ is the n_i-dimensional unit vector.

Chapter 5

Experimental Results

The data used in our experiments are the news items down-loaded from the Yahoo news group and Reuters-21578 newswire benchmark. Each document in the data sets is indexed by human experts. We preprocess the HTML news items by (i) document parsing (remove headers and tags in the HTML files), and (ii) removing *stopwords* and low frequency words as mentioned earlier.

- Yahoo News Data. The Yahoo news items were down-loaded from the Yahoo news group in the year of 1997. There were 9 categories on the site. We chose to use 7 of them, which are Business (B), Entertainment (E), Health (H), International (I), Politics (P), Sports (S), and Technology (T). The remaining two categories, Top-stories and Local are excluded, since the associated semantics can not be easily captured by word frequencies or occurrences. Each news item has a title and a short summary, which is about 120 words, on an average. We have constructed two data sets from Yahoo news data. One of them is called

Yahoo news set, which contains 814 training samples, and two test data sets (news items at different time intervals, see Table 5.1). Test data set1 has 680 news items and test data set2 has 621 documents. The second dataset is called *enlarged Yahoo news set*, which has 4,199 training samples and 2,000 testing samples (see Table 5.2).

	Categories	B	E	H	I	P	S	T
Training Data	#of documents	130	133	91	110	130	130	90
	total #of terms	1848	2045	1213	1974	2070	1659	1364
Test Data Set1	#of documents	110	111	79	80	110	111	79
	total #of terms	2155	2583	1535	1999	2439	1952	1618
Test Data Set2	#of documents	100	101	78	70	101	101	70
	total #of terms	2046	2834	1803	2604	2070	1974	1689

Table 5.1: Yahoo news training and test data.

	Categories	B	E	H	I	P	S	T
Training Data	#of documents	682	687	476	510	682	673	489
	total #of terms	5841	8610	5020	6893	6963	5756	4721
Test Data Set	#of documents	322	334	224	245	323	330	222
	total #of terms	3904	5275	3199	4086	4482	3803	3043

Table 5.2: Enlarged Yahoo news training and test data.

- Reuters-21578 benchmark. This dataset consists of 21,578 newswire stories which appeared on the Reuters newswire in 1987. It can be accessed from Lewis' professional home page (http://www/research/att.com/*sim*lewis). We use the *TOPIC* categories set which has been used in almost all previous experiments with the Reuters data. The *TOPIC* set contains 12,668 newswire stories with 135 categories, but only 120 categories have at least one document, and only 57

categories have at least 20 documents (see Table 5.3). The data are split into a training set (9,649 documents) and a test set (3,019 documents).

5.1 Evaluation of Text Classification Effectiveness

In information retrieval, there are two important measures of system effectiveness, called *recall* and *precision* [69]. Recall measures the ability of the system to present all relevant items. It is defined as the number of relevant documents divided by the total number of relevant documents in the collection. Precision measures the ability of the system to present only the relevant items. It is defined as the number of relevant documents retrieved divided by the total number of documents retrieved. For example, suppose there are 100 documents in a collection that are relevant to a query **Q**, and a system retrieves 200 documents (for query **Q**), among which 80 are relevant to **Q**. So, the recall of this system for query **Q** is $80/100 = 0.80$, while the precision is $80/200 = 0.40$.

Text classification is the assignment of documents to one or more pre-existing set of Categories, rather than retrieving them in response to a query. So, the recall and precision measures used in information retrieval do not fit a classification system quite well. Lewis [42] defined *microaveraging recall* and *precision* to evaluate a text classification (categorization) system. For a two-class classification problem, a binary decision is made based on whether a document belongs to a given class or not. For a set of n test documents, a contingency table (Table 5.4) can be made for these n binary decisions [42].

Category	acq	alum	austdlr	austral	barley	bfr	bop
Training Set	1650	35	4	0	37	0	75
Test Set	715	20	0	0	0	0	16
Category	can	carcass	castor -meal	castor -oil	castor seed	citrus pulp	cocoa
Training Set	3	50	0	1	1	1	55
Test Set	0	7	0	0	0	0	17
Category	coconut	coconut -oil	coffee	copper	copra -cake	corn	corn -oil
Training Set	4	4	111	47	2	182	1
Test Set	1	0	23	14	0	1	0
Category	cornglu tenfeed	cotton	cotton -meal	cotton -oil	cotton seed	cpi	cpu
Training Set	2	39	0	1	0	69	3
Test Set	0	9	0	0	0	18	1
Category	crude	cruzado	dfl	dkr	dlr	dmk	drachma
Training Set	389	1	2	1	131	10	0
Test Set	160	0	0	0	16	0	0
Category	earn	escudo	f- cattle	ffr	fish meal	flax seed	fuel
Training Set	2877	0	2	0	2	0	13
Test Set	1083	0	0	0	0	0	7
Category	gas	gnp	gold	grain	ground nut	ground nut-meal	ground nut-oil
Training Set	37	101	94	433	5	0	1
Test Set	15	28	27	127	2	0	0
Category	heat	hk	hog	housing	income	install -debt	interest
Training Set	14	0	16	16	9	5	347
Test Set	4	0	3	2	4	1	99
Category	invent ories	ipi	iron- steel	jet	jobs	l- cattle	lead
Training Set	5	41	40	4	46	6	15
Test Set	0	12	12	1	12	0	9
Category	lei	lin- meal	lin- oil	lin seed	lit	live stock	lumber
Training Set	12	1	1	2	1	75	10
Test Set	3	0	0	0	0	11	4

Category	lupin	meal-feed	mexpeso	money-fx	money-supply	naphtha	nat-gas
Training Set	0	30	0	538	140	2	75
Test Set	0	6	0	151	31	1	15
Category	nickel	nkr	nzdlr	oat	oil seed	orange	palladium
Training Set	8	1	2	8	124	16	2
Test Set	1	0	0	0	19	9	0
Category	palm -meal	palm -oil	palm kernel	peseta	pet-chem	platinum	plywood
Training Set	0	30	2	1	20	5	4
Test Set	0	0	0	0	8	3	0
Category	pork-belly	potato	propane	rand	rape -meal	rape -oil	rape seed
Training Set	3	3	3	2	1	5	18
Test Set	0	3	2	0	0	0	0
Category	red-bean	reserves	retail	rice	ringgit	rubber	rupiah
Training Set	1	55	23	35	1	37	1
Test Set	0	14	1	2	0	9	0
Category	rye	saud riyal	sfr	ship	silk	silver	singdlr
Training Set	1	3	0	197	0	21	0
Test Set	0	0	0	55	0	2	0
Category	skr	sorghum	soy-meal	soy bean	stg	strategic -metal	sugar
Training Set	1	24	13	78	17	16	126
Test Set	0	0	0	1	0	6	27
Category	sun-meal	sun oil	sun seed	tapioca	tea	tin	trade
Training Set	1	5	11	3	9	18	369
Test Set	0	0	0	0	3	11	105
Category	tung	tung -oil	veg -oil	wheat	wool	wpi	yen
Training Set	0	0	87	212	2	19	45
Test Set	0	0	24	3	0	9	6
Category	zinc						
Training Set	21						
Test Set	7						

Table 5.3: Number of documents in each category of Reuters-21578 *TOPIC* set.

	Yes is correct	No is correct	
Decide Yes	a	b	a+b
Decide No	c	d	a+b
	a+c	b+d	a+b+c+d=n

Table 5.4: Contingency table of binary decisions for a test set, from [42].

Given the contingency table, recall and precision are defined as: recall $= a/(a+c)$ and precision $= a/(a+b)$. For m classes and n test documents, if we treat each classification as a binary decision, then a total of mn decisions are made. Microaveraging considers all mn decisions as a single group and computes recall and precision as defined above. There is a tradeoff between recall and precision. We can have a very high recall rate by always deciding *yes*, but then the corresponding precision will be small. A classification system attempts to maximize both precision and recall simultaneously. Usually, a break-even point of recall and precision measurement is used, i.e., when microaveraging recall is equal to microaveraging precision. Linear interpolation is often used to get the break-even values.

We consider the text classification as a multi-class classification problem, i.e., each time a classifier assigns a document to one of the m categories instead of classifying it to one class against all the others. We make an $m \times m$ confusion matrix, where entry (i, j) shows how many documents belonging to category i were assigned to category j. The sum of diagonal entries divided by the number of test examples measures the classification accuracy, which is the microaveraging recall and precision defined by Lewis [42]. If a document indeed belongs to several categories, then if the decision made by a classifier belongs to one of these "true label" categories, we treat this

		NB	NN	DT	NNet	SS
Test Data Set1	#of Misclassifications	115	165	178	131	139
	Recognition Rate (%)	**83.1**	75.7	73.8	80.74	79.6
Test Data Set2	#of Misclassifications	125	179	144	90	111
	Recognition Rate (%)	79.87	71.18	76.8	**85.51**	82.13

Table 5.5: Comparison of the five classification algorithms (NB, NN, DT, $NNet$, and SS).

decision as a correct decision, and incorrect otherwise.

5.2 Individual Classifiers

On the small Yahoo dataset, we compared the five classification algorithms (naive Bayes classifier (NB), nearest neighbor classifier (NN), decision tree classifier (DT), artificial neural network classifier (NNet), and the subspace classifier (SS)) on our two test data sets. Table 5.5 shows a comparison of the recognition rates (microaveraging recall/precision) using these five classification algorithms individually. The experimental results show that all the five classification algorithms perform reasonably well; the NB approach performs the best on test data set1, while the NNet classifier performs the best on test set2; the NNet and the subspace methods outperform the other three classifiers on test data set1 and test data set2, respectively. Confusion matrices of the classification results using the five classifiers on test set1 are shown in Tables 5.6-5.10.

Figure 5.11 shows the classification results using NB classifier on the enlarged Yahoo data set and Reuters-21578 news story benchmark.

	B	E	H	I	P	S	T	Recognition Rate(%)
B	81	1	2	0	7	0	19	73.6
E	1	89	1	8	3	2	7	80.2
H	0	0	79	0	0	0	0	100.0
I	1	1	0	53	25	0	0	66.3
P	5	0	1	10	91	0	3	82.7
S	2	0	1	1	3	102	2	91.9
T	7	0	2	0	0	0	70	88.6

Table 5.6: Confusion matrix of the classification results using the naive Bayes Classifier (NB) without dimensionality reduction.

	B	E	H	I	P	S	T	Recognition Rate(%)
B	74	5	2	2	8	2	17	67.3
E	3	83	4	4	5	8	4	74.8
H	2	1	70	0	0	2	4	88.6
I	5	3	3	50	18	0	1	62.5
P	9	1	4	16	76	0	4	69.1
S	2	0	0	2	0	106	1	95.5
T	19	1	1	0	1	1	56	70.9

Table 5.7: Confusion matrix of the classification results using the nearest neighbor classifier (NN) without dimensionality reduction.

	B	E	H	I	P	S	T	Recognition Rate(%)
B	70	8	1	0	13	1	17	63.6
E	18	77	0	5	2	7	0	70.6
H	1	2	71	4	0	0	1	89.9
I	3	8	0	54	8	5	2	67.5
P	7	5	2	15	73	3	5	66.4
S	1	11	0	1	2	96	0	86.5
T	9	4	2	3	1	0	61	76.3

Table 5.8: Confusion matrix of the classification results using the $C5$ decision tree classifier (DT) without dimensionality reduction.

	B	E	H	I	P	S	T	Recognition Rate(%)
B	72	1	7	0	8	3	19	65.45
E	1	74	14	2	2	14	4	65.00
H	0	0	79	0	0	0	0	100.0
I	1	1	3	52	19	4	0	65.00
P	4	0	4	7	88	2	5	80.00
S	0	0	2	0	0	109	0	98.20
T	0	0	5	0	0	0	75	94.94

Table 5.9: Confusion matrix of the classification results using adaptive neural network classifier (NNet) without dimensionality reduction.

	B	E	H	I	P	S	T	Recognition Rate(%)
B	65	3	6	2	8	2	24	59.1
E	1	89	2	7	1	6	5	80.2
H	0	0	79	0	0	0	0	100.0
I	1	2	3	51	22	0	1	63.8
P	6	0	6	10	84	1	3	76.4
S	2	2	1	0	0	106	0	95.5
T	9	0	2	1	0	0	67	84.8

Table 5.10: Confusion matrix of the classification results using the subspace classifier (SS) without dimensionality reduction.

	Yahoo	Reuters
#of Misclassifications	285	430
Recognition Rate (%)	85.55	85.76

Table 5.11: Classification accuracy using NB on two large data sets.

Combination of Classifiers	Combination Approaches	Testing Data Set1(%)	Testing Data Set2(%)
NB, *SS*, *NN*	Simple Voting	80.29	81.96
	DCS	80.00	77.13
	ACC	82.21	82.45
NB, *SS*	*DCS*	80.44	79.87
	ACC	**83.24**	**82.93**
Best Individual Classifier		83.1	82.13

Table 5.12: Classification Accuracy of combinations of multiple classifiers.

5.3 Combination of Different Classifiers

Results of combinations of multiple classifiers using different combination approaches are summarized in Table 5.12. We set $k = 20$ in our experiments. Note that for these two datasets, there was no significant improvement by using a combination of classifiers. Our opinion is that the performance of a combination of classifiers is data dependent.

5.4 Dimensionality Reduction

Best feature selection, PCA, LDA, ODA, and term grouping in subspace are used to reduce the dimensionality in our experiments. We used mutual information to weigh the words appearing in the training documents; a subset of the words with the highest weights is selected. We compared the four classifiers (NB, NN, DT, and SS) using this feature selection technique. Figure 5.1 shows the recognition rate of the four classification algorithms with different sized feature subsets. From this figure, we can see that (i) a small number of features is not suitable for NB, and (ii) the

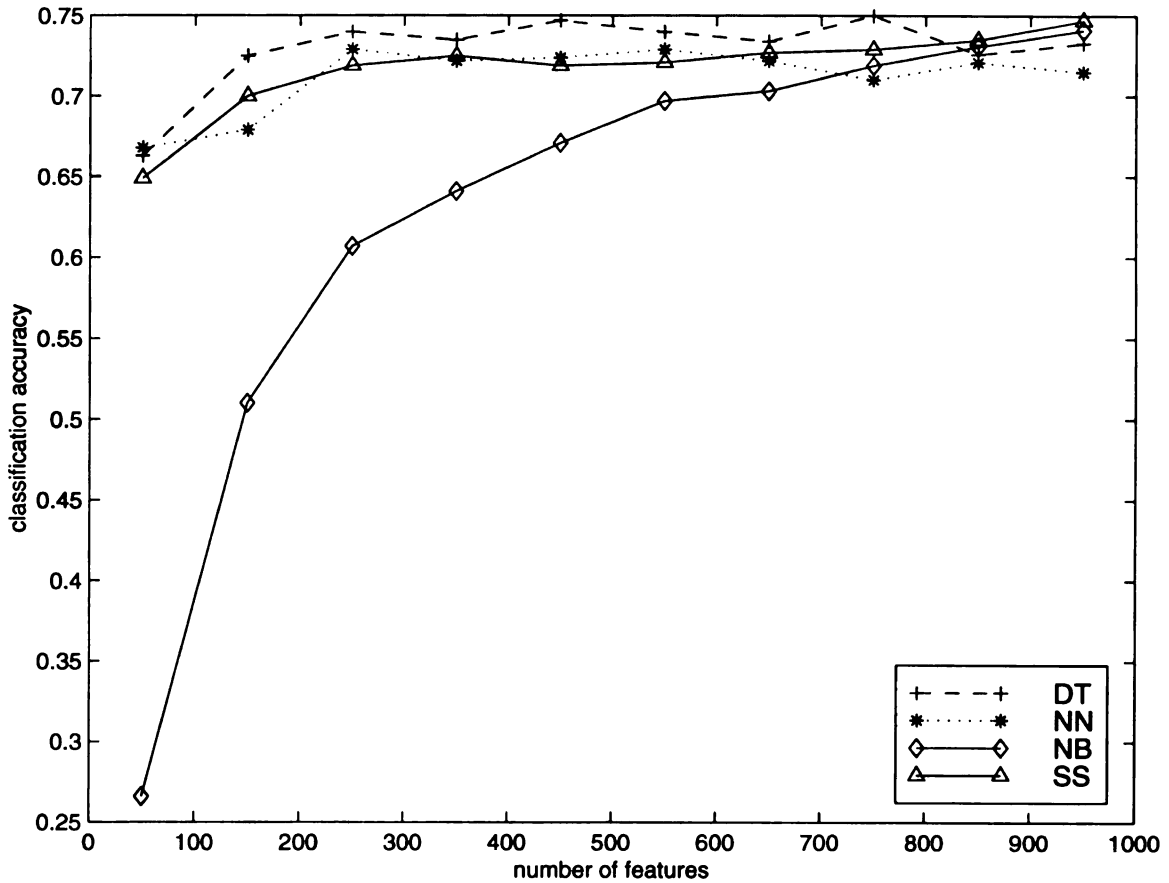

Figure 5.1: The accuracy of each algorithm with a different sized feature subsets.

performance of NB and SS classifiers tend to improve when the number of features is increased.

We used PCA method to project the original feature space onto a lower dimensional subspace. We set $\nu = 0.90$ (90% of the variance retained), resulting in about 400 features in the small Yahoo training set. When using LDA and ODA, original features are projected onto a 6 dimensional subspace, since we have 7 classes in the Yahoo data sets. We apply NN classifier on the projected feature vectors. Table 5.13 shows a comparison of different feature extraction methods, including PCA, LDA and ODA. Confusion matrices of the classification results using NN classifier with

		NN	NN with PCA	NN with LDA	NN with ODA
Test Data Set1	#of Errors	165	166	136	121
	Recall (%)	75.7	75.6	80.0	82.21
Test Data Set2	#of Errors	179	180	109	102
	Recall (%)	71.18	71.01	82.45	83.57

Table 5.13: Comparison of the feature extraction methods using NN.

	B	E	H	I	P	S	T	Recognition Rate(%)
B	75	4	2	2	9	3	15	68.2
E	4	86	1	4	4	8	4	77.5
H	2	1	72	0	0	3	1	91.1
I	6	4	1	47	21	1	0	58.8
P	10	3	4	17	71	0	5	64.6
S	3	0	0	2	0	105	1	94.6
T	17	1	1	0	1	1	58	73.4

Table 5.14: Confusion matrix of the classification result using the nearest neighbor classifier with PCA feature extraction.

PCA, LDA and ODA on test set1 are shown in Tables 5.14-5.16, respectively. The experimental results show that ODA outperforms LDA and PCA.

We apply our term-grouping technique to the subspace method on test set1. A total of 30 term groups were chosen in our experiment. A comparison of the recognition rates before and after using the term-grouping technique on data set1 is shown in table 5.17. It shows that the performance of the SS method improved marginally with the term-grouping technique.

	B	E	H	I	P	S	T	Recognition Rate(%)
B	67	2	4	1	10	5	21	60.92
E	2	97	0	3	2	5	2	87.39
H	0	4	74	0	0	1	0	93.67
I	3	1	1	46	26	2	1	57.50
P	9	2	2	12	83	1	1	75.45
S	1	2	1	2	0	105	0	94.59
T	3	2	2	0	0	0	72	91.14

Table 5.15: Confusion matrix of the classification result using the nearest neighbor classifier with *LDA* feature extraction.

	B	E	H	I	P	S	T	Recognition Rate(%)
B	73	3	2	1	8	5	18	66.36
E	1	99	0	6	1	3	1	89.19
H	0	1	77	0	0	1	0	97.47
I	4	6	1	41	26	1	1	51.25
P	9	1	2	6	91	1	0	82.73
S	1	2	1	0	0	107	0	96.40
T	4	1	3	0	0	0	71	89.87

Table 5.16: Confusion matrix of the classification result using the nearest neighbor classifier with *LDA* feature extraction.

	SS without term-grouping	*SS* with term-grouping
#of Misclassifications	139	137
Recognition Rate (%)	79.6	79.9

Table 5.17: Comparison of using the subspace method with or without the term-grouping feature reduction technique on test set1.

Chapter 6

Conclusions and Discussion

We have applied five different classification methods (NB, NN, DT, SS, and $NNet$) to the problem of document categorization. These methods were evaluated individually and in combination. Since document classification involves a high dimensional feature space, we also studied the effect of different feature reduction techniques (the individual best feature selection, PCA, DA, and term-grouping in subspace) on the performance of these classifiers. The seven classes of Yahoo news items used in our experiments have a large overlap of words in their documents (e.g., in 2,948 total words, there are 1,096 common words between *international* and *politics* news categories, 744 out of 2,468 words are common between *business* and *technology* news groups), so this is a difficult classification problem. We can make the following observations based on our experimental results:

1. Comparison of different classifiers:

   (a) For the Yahoo news data, all the five classifiers perform reasonably well

on our data sets. We also asked four students in our laboratory to assign lables to 680 documents in test set1. The average classification accuracy of the label assigned by these subjects is about 81%. Comparing this classification accuracy to the accuracy of the best classifier (83.1%), we see that machine classification performance is reasonable. Weiss et al. [79] also reported that the accuracy of human judgment on 1000 messages on 10 USENET newsgroups (misc.health.diabetes, sci.math.num-analysis, dc.politics, rec.food.restaurants, alt.tv.seinfeld, comp.sys.ibm.pc.games.sports,

rec.arts.comics.dc.universe, sci.military.naval, talk.philosophy.misc and humanities.lit.authors.shakespeare) is about 85%. Both NB and SS classifiers work better than NN and DT methods, but the performance of NB and SS classifiers is data dependent. Most of the misclassifications are between *international* and *politics* categories, and between *business* and *technology* document classes which inherently have a large overlap of terms. If we combine *international* and *politics* news groups, and combine *business* and *technology* news groups, the performance of all the four classification algorithms on the resulting 5-class problem improves by an average of 7%. Classification accuracies on test set1 for the four classifiers on this five-class problem are shown in Table 6.1.

(b) When we reverse the role of the training and test data (combining test set1 and test set2 as a training set, and the old training set is now the

	NB	NN	DT	SS
#of Misclassifications	67	95	128	101
Recognition Rate (%)	90.15	86.03	81.18	85.15

Table 6.1: Comparison of the four classification algorithms (NB, NN, DT, and SS) on test set1 for the reduced 5-class problem.

test set), the classification accuracy remains essentially the same (e.g, the classification accuracy of NB is 83.91%). This shows that the classifier is not sensitive to the choice of the training data (assuming that the training data is sufficiently large).

(c) For the enlarged Yahoo news data and Reuters newswire, NB performs steadily well on these large databases, despite the "independence" assumption which is not always satisfied in document classification. While SS performs well on the enlarged Yahoo news data set, it performs very poor for Reuters data set. It may be because the data is noisy. Some preprocessing, such as fine semantic analysis [27] may improve the performance.

2. Combinations of multiple classifiers do not always improve the classification accuracy. The adaptive classifier combination introduced here works better than simple voting and dynamic classifier selection approaches on our two test data sets.

3. Dimensionality reduction:

(a) There is no significant peaking in classification performance observed in our experiments. In particular, the performance of NB improves as the number of features increase (which is different from the results obtained in [44]).

Additionally, NB does not use a small number of features effectively (which is also different from the observation in [44] that 10 features for Reuters news groups performs the best). This indicates that our data set is less separable than the Reuters newswire.

(b) The widely adopted LSI method in information retrieval systems is essentially the same as the PCA when used to compare the similarity of two documents. LDA performs much better than PCA. In particular, ODA performs the best in our experiment.

(c) The problem with feature selection is that the small number of selected words may not generalize well to new documents. However, the advantage of dimensionality reduction is not only to improve the recognition rate (eliminate the problem of overfitting), but the reduced number of features lead to lower time and space complexities. The term-grouping method reduces the feature dimensionality, and overcomes the generalization problem of feature selection, while maintaining the performance of the classifier.

It is difficult to say which classifier is the best. Generally, we believe that simple classifies, such as NB and SS can give fairly good classification results. These methods are simple, fast, and can be easily scaled up for very large databases. $NNet$ classifier also performs well, and it is very fast in the test stage. Feature extraction, such as LDA can improve the classification performance significantly in some cases. However, feature extraction is computationally expensive, and has difficulties in scaling up to large databases (e.g., memory requirement).

Incorporating natural language processing into a classifier (feature representation, feature selection) can remove noisy features, and improve the classification accuracy (e.g., using WordNet[1] [27]). In our experiments, we also observed that, for NB classifier, if we assign a test document to the top two matching categories, the microrecall rate is 97% on enlarged Yahoo news data, and is 99% on Reuters newswire database.

[1]http://www.cogsci.princeton.edu/~wn

Appendices

Appendix A

Examples of Training Samples of Yahoo datasets

We show one training example from each of the seven Yahoo news groups after Web document parsing, and their word list (used to construct the feature vector) after *stopwords* and low-frequency word removal.

Minimum Wage Rises - Nearly 7 million Americans are getting a raise on this Labor Day. The federal minimum wage is rising to $5.15 an hour. Fast food workers, retail clerks, gas station attendants and others will be earning 40 cents an hour more when they report to work as the second phase of the hike goes into effect. It was first raised to $4.75 last Oct. 1. According to a report to be issued tomorrow by the Economic Policy Institute, most of the 6.8 million workers affected by the minimum wage hike are women who work in the service sector. The Washington, D.C.-based think tank's study found that in 18 states, more than 10 percent of the work force will be affected by the minimum wage increase.	minimum wage rises _ nearly _ million americans ___ getting _ raise __ ____ labor day_ ___ federal minimum wage __ rising __ _____ __ hour_ fast food workers_ retail clerks_ gas station attendants ___ ______ ____ __ earning __ cents __ hour ____ ____ ____ report __ ____ __ ___ ______ phase __ ___ hike goes ____ effect_ __ ___ _____ raised __ _____ ____ oct_ __ according __ _ report __ __ issued tomorrow __ ___ economic policy institute_ ____ __ ___ ___ million workers affected __ ___ minimum wage hike ___ women ___ ____ __ ___ service sector_ ___ washington_ _____based _____ tank__ study found ____ __ __ _______ ____ ____ __ percent __ ___ ____ force ____ __ affected __ ___ minimum wage increase_
(a)	(b)

Figure A.1: An example of the *bussiness* news group; (a) a training sample; (b) extracted word list.

Diana Funeral to be Saturday - Princess Diana's funeral will be held Saturday. Buckingham Palace announced Monday that the "people's princess" would be honored with services at Westminster Abbey in London. Diana will be buried in private services at her family's estate in Althorp, central England. Until the funeral, Diana's body will lie in the Chapel Royal at St. Jame's Palace in London. Members of the public will not be allowed to file past, but can write a personal message in books of condolence. Thousands of people have been lining up to do so. French prosecutors Monday disclosed a new twist to the tragedy that has numbed Britons. French officials said the driver who crashed Princess Diana's car -- killing her, her friend Dodi Al Fayed and himself -- was driving at a high speed with twice the blood alcohol level that would have landed him in jail.	diana funeral __ __ saturday _ princess diana__ funeral ____ __ held saturday_ buckingham palace announced monday ____ ___ _people__ princess_ _____ __ honored ____ services __ westminster abbey __ london_ diana ____ __ buried __ private services __ ___ family__ estate __ althorp_ central england_ _____ ___ funeral diana__ body ____ lie __ ___ chapel royal __ st_ jame__ palace __ london_ _______ __ ___ public ____ ___ __ allowed __ file past_ ___ ___ write _ personal message __ books __ condolence_ thousands __ people ____ ____ lining __ __ __ ___ french prosecutors monday disclosed _ ___ twist __ ___ tragedy ____ ___ numbed britons_ french officials ____ ___ driver ___ crashed princess diana__ car __ killing ____ ___ friend dodi al fayed ___ _______ __ ___ driving __ _ ____ speed ____ twice ___ blood alcohol level ____ _____ ____ landed ___ __ jail_
(a)	(b)

Figure A.2: An example of the *entertainment* news group; (a) a training sample; (b) extracted word list.

Heart Failure Therapy Saves Lives - A more aggressive and comprehensive management program for heart failure can reduce hospital admissions by 85% in patients waiting for a heart transplant, a new study suggests. The more comprehensive treatment may not only improve the quality of life for patients, it also saves money. An estimated 400,000 to 800,000 people in the U.S. have severe heart failure, and 200,000 die of the disease each year, according to the report in the Journal of the American College of Cardiology.	heart failure therapy saves lives _ _ ____ aggressive ___ comprehensive management program ___ _ heart failure ___ reduce hospital admissions __ ___ __ patients waiting ___ _ heart transplant_ _ ___ study suggests_ ___ ____ comprehensive treatment ___ ___ ____ improve ___ quality __ life ___ patients_ __ ____ saves money_ __ estimated _____ __ _____ people __ ___ ____ ____ severe heart failure_ ___ _____ die __ ___ disease ____ _____ according __ ___ report __ ___ journal __ ___ american college __ cardiology_
(a)	(b)

Figure A.3: An example of the *health* news group; (a) a training sample; (b) extracted word list.

Diana to Receive 'Unique Funeral' - Buckingham Palace Monday announced that Princess Diana will be given a "unique funeral for a unique person" Saturday in London. Her coffin will be carried through the streets to services at Westminster Abbey, to be followed by a private burial at her family's estate in central England. Until the funeral, Diana's body will lie in the Chapel Royal at St. Jame's Palace in central London, and thousands of mourners have flocked there to sign books of condolences. Crowds have also gathered outside Kensington Palace, Diana's London home, and outside Buckingham Palace, as they continue mourning the late Princess.	diana __ receive _unique funeral_ _ buckingham palace monday announced ____ princess diana ____ __ _____ _ _unique funeral ___ _ unique person_ saturday __ london_ ___ coffin ____ __ carried _______ ___ streets __ services __ westminster abbey_ __ __ followed __ _ private burial __ ___ family__ estate __ central england_ _____ ___ funeral_ diana__ body ____ lie __ ___ chapel royal __ st_ jame__ palace __ central london_ ___ thousands __ mourners ____ flocked _____ __ sign books __ condolences_ crowds ____ ____ gathered outside kensington palace_ diana__ london home_ ___ outside buckingham palace_ __ ____ continue mourning ___ late princess_
(a)	(b)

Figure A.4: An example of the *international* news group; (a) a training sample; (b) extracted word list.

Clinton Sends Condolences - President Clinton is sending condolences to British Prime Minister Tony Blair on the death of Princess Diana, saying "all of us have lost a friend and a strong voice for those less fortunate." In a letter written at his vacation retreat in Martha's Vineyard, Clinton praised Diana's "untiring and selfless commitment to helping persons in need, particularly children, the victims of AIDS and landmines, and other vital humanitarian concerns." Clinton sent letters expressing similar sentiments to Queen Elizabeth, Diana's ex-husband Prince Charles, and Diana's brother Charles Spencer. A White House spokesman says there's been no decision on who will represent the United States at Diana's funeral Saturday.	clinton sends condolences _ president clinton __ sending condolences __british prime minister tony blair __ ___ death __ princess diana_ saying ____ __ __ ____ lost _ friend ___ _ strong voice ___ _____ ____ fortunate__ __ _ letter written __ ___ vacation retreat __ martha__ vineyard_ clinton praised diana__ _untiring ___ selfless commitment __ helping persons __ _____ particularly children_ ___ victims __ aids ___ landmines_ ___ _____ vital humanitarian concerns__ clinton sent letters expressing similar sentiments __ queen elizabeth_ diana__ ex_husband prince charles_ ___ diana__ brother charles spencer_ _ white house spokesman ____ _______ ____ __ decision __ ___ ____ represent ___ united ______ __ diana funeral saturday_
(a)	(b)

Figure A.5: An example of the *politics* news group; (a) a training sample; (b) extracted word list.

49ers' Rice Has Knee Surgery - The 1997 season is off to a nightmarish start for the San Francisco 49ers. Jerry Rice, who holds all the major receiving records in NFL history, underwent surgery Monday to repair a torn anterior cruciate ligament and torn medial collateral ligament in his left knee and is expected to miss the rest of the season. An MRI taken Sunday night revealed the injury. Dr. Michael Dillingham, the Niners' team physician, performed the surgery and estimated that Rice will be sidelined four to six months. Rice suffered the injury on a reverse in the second quarter when he was dragged to the ground by his facemask by Tampa Bay defensive end Warren Sapp. Rice's left knee buckled as he was pulled awkwardly to the ground on the play, which resulted in a 10-yard loss. Sapp was called for a 15-yard facemask penalty on the play.	__ers_ rice ___ knee surgery _ ___ ____ season __ ___ __ _ nightmarish start ___ ___ san francisco __ers_ jerry rice_ ___ holds ___ ___ major receiving records __ nfl history_ underwent surgery monday __ repair _ torn anterior cruciate ligament ___ torn medial collateral ligament __ ___ left knee ___ __ expected __ miss ___ rest __ ___ season_ __ mri _____ sunday night revealed ___ injury_ dr_ michael dillingham_ ___ niners_ team physician_ performed ___ surgery ___ estimated ____ rice ____ __ sidelined ____ __ six months_ rice suffered ___ injury __ _ reverse __ ___ ______ quarter ____ __ ___ dragged __ ___ ground __ ___ facemask __ tampa bay defensive ___ warren sapp_ rice__ left knee buckled __ __ ___ pulled awkwardly __ ___ ground __ ___ play_ _____ resulted __ _ ___yard loss_ sapp ___ called ___ _ ___yard facemask penalty __ ___ play_
(a)	(b)

Figure A.6: An example of the *sports* news group; (a) a training sample; (b) extracted word list.

Apple Buys Clone Maker Assets - Apple Computer said today it's paying $100 million in stock to buy the core assets of Power Computing, a privately held licensee of Apple's Macintosh line of computers. Power Computing has pioneered direct marketing and sales in the Macintosh market, successfully building a $400 million business," Apple board member and founder Steve Jobs said announcing the agreement. We look forward to learning from their experience, and welcoming their customers back into the Apple family." Apple sold the Macintosh license to Power Computing in December 1994.	apple buys clone maker assets _ apple computer ____ _____ ____ paying ____ million __ stock __ buy ___ core assets __ power computing_ _ privately held licensee __ apple__ macintosh line __ computers_ power computing ___ pioneered direct marketing ___ sales __ ___ macintosh market_ successfully building _ ____ million business__ apple board ______ ___ founder steve jobs ____ announcing ___ agreement_ __ look forward __ learning ____ _____ experience_ ___ welcoming _____ customers ____ ____ ___ apple family__ apple sold ___ macintosh license __ power computing __ december _____
(a)	(b)

Figure A.7: An example of the *technology* news group; (a) a training sample; (b) extracted word list.

Appendix B

The Laplace's Law of Succession

The Laplace's Law of Succession is a Bayesian approach to the problem of multinomial parameter estimation [64]. Given a set of distinct symbols, the problem is to estimate the symbol probabilities based on the known frequency with which each symbol occurred in the past.

Let θ_i be the probability that word w_i occurred in class c. We assume that the words appear in the documents independently. The problem is to find $\hat{\theta}_i$, an estimate of θ_i.

Let n be the total number of occurrences of words in class c, n_{w_i} be the number of occurrences of word w_i in class c, and k be the total number of distinct words in all the documents, i.e, the vocabulary size.

Let the a priori probability density of the unknown parameters $(\theta_1, ..., \theta_k)$ be

uniform, i.e.,

$$P(\theta_1, ..., \theta_k) = \begin{cases} l & 0 \leq \theta_i \leq 1, i = 1, ..., k, \sum_1^k \theta_i = 1 \\ 0 & \text{otherwise.} \end{cases} \tag{B.1}$$

Since the p.d.f. must integrate to 1, $l = (k-1)!$.

The *a posteriori* densities of $(\theta_1, ..., \theta_k)$ can be written as:

$$P(\theta_1, ..., \theta_k | (n_{w_1}, ..., n_{w_k})) = \frac{P(n_{w_1}, ..., n_{w_k} | (\theta_1, ..., \theta_k)) P(\theta_1, ..., \theta_k)}{P(n_{w_1}, ..., n_{w_k}).} \tag{B.2}$$

$P(n_{w_1}, ..., n_{w_k} | (\theta_1, ..., \theta_k))$ is a multinomial distribution:

$$P(n_{w_1}, ..., n_{w_k} | (\theta_1, ..., \theta_k)) = \frac{n!}{n_{w_1}! ... n_{w_k}!} \theta_1^{n_{w_1}} ... \theta_k^{n_{w_k}}, \tag{B.3}$$

when $n_{w_1} + n_{w_2} + ... + n_{w_k} = n$. So,

$$\begin{aligned} P(n_{w_1}, ..., n_{w_k}) &= \int ... \int P(n_{w_1}, ..., n_{w_k} | (\theta_1, ..., \theta_k)) P(\theta_1, ..., \theta_k) d\theta_1 ... d\theta_k & \text{(B.4)} \\ &= \frac{(k-1)! n!}{n_{w_1}! ... n_{w_k}!} \int ... \int \theta_1^{n_{w_1}} ... \theta_k^{n_{w_k}} d\theta_1 ... d\theta_k & \text{(B.5)} \\ &= \frac{(k-1)! n!}{n_{w_1}! ... n_{w_k}!} \times \frac{n_{w_1}! ... n_{w_k}!}{(n+k-1)!} & \text{(B.6)} \\ &= \frac{n!(k-1)!}{(n+k-1)!,} & \text{(B.7)} \end{aligned}$$

where $0 \leq \theta_i \leq 1$ and $\sum_1^k \theta_i = 1$. See [65] for details. For a squared-error loss function,

$$\hat{\theta}_i = \int ... \int \theta_i P(\theta_1, ..., \theta_k | (n_{w_1}, ..., n_{w_k})) d\theta_1 ... d\theta_k \tag{B.8}$$

$$= \frac{(n+k-1)!}{n!} \int \dots \int \frac{n!}{n_{w_1}! \dots n_{w_k}!} \theta_1^{n_{w_1}} \dots \theta_i^{n_{w_1}+1} \dots \theta_k^{n_{w_k}} d\theta_1 \dots d\theta_k \quad \text{(B.9)}$$

$$= \frac{(n+k-1)!}{n!} \times \frac{n!(n_{w_1}+1)}{(n+k)!} \quad \text{(B.10)}$$

$$= \frac{n_{w_1}+1}{n+k} \quad \text{(B.11)}$$

Bibliography

[1] *Gentle Introduction to RainBow*[1].

[2] C. Apte, F. Damerau, and S. M. Weiss. Text Categorization: a Symbolic Approach. In *ACM Trans. on Information Systems, Vol. 12, No. 3*, pages 233–251, 1994.

[3] D. B. Aronow, S. Soderland, J. M. Ponte, F. F. Feng, W. B.Croft, and W. G. Lehnert. Automated Classification of Encounter Notes in Computer-Based Medical Records. In *Proc. of the Eighth World Congress on Medical Informatics Vol. 8*, pages 8–12, 1995.

[4] M. Balabanovic, Y. Shoham, and Y. Yun. An Adaptive Agent for Automated Web Browsing. Technical Report SIDL-WP-1995-0023, Stanford University, 1995.

[5] H. Berghel. Cyberspace 2000: Dealing with Information Overload. In *Digital Village, Communications of the ACM, Vol. 40, No. 2*, pages 19–24, 1997.

[6] J. Bradshaw, editor. *Software Agents.* AAAI Press/The MIT Press, 1997.

[1] http://www.cs.cmu.edu/afs/cs/project/theo-11/www/naive-bayes/gentle_introduction.html

[7] J. P. Callan, W. B. Croft, and S. M. Harding. The INQUERY Retrieval System. In *Proc. of the 3rd International Conference on Database and Expert Systems Applications*, pages 347–356, 1993.

[8] J. G. Carbonell, R. S. Michalski, and T. M. Mitchell. *Machine Learning, An Artificial Intelligence Approach*, chapter An Overview of Machine Learning, pages 3–23. Tioga Publishing Company, 1983.

[9] A. Chavez and P. Maes. Kasbah: An Agent Marketplace for Buying and Selling Goods. In *Proceedings of the First International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology*, 1996.

[10] H. Chen. Machine Learning for Information Retrieval: Neural Networks, Symbolic Learning, and Genetic Algorithms. In *Journal of the American Society for Information Science, Vol. 46, No. 3*, pages 194–216, 1995.

[11] H. Chen and V. Dhar. Cognitive process as a basis for intelligent retrieval system design. In *Information Processing and Management, Vol. 27, No. 5*, pages 405–432, 1991.

[12] W. W. Cohen. Fast Effective Rule Induction. In *Machine Learning: Proc. of 12th International Conference*, Lake Tahoe, California, 1995.

[13] W. W. Cohen. Learning Rules that Classify E-Mail. In *Proc. 1996 AAAI Spring Symposium on Machine Learning in Information Access*, Stanford, 1996.

[14] M. W. Craven and J. W. Shavlik. Learning Symbolic Rules Using Artificial Neural Networks. In *Proc. of the Tenth International Conference on Machine Learning*, pages 73–80, 1993.

[15] V. Dasigi and R. C. Mann. Neural Net Learning Issues in Classification of Free Text Documents. In *AAAI Spring Symposium on Machine Learning in Information Access Technical Papers*, Palo Alto, March 1996.

[16] P. Domingos and M. Pazzani. Beyond Independence: Conditions for the Optimality of the Simple Bayesian Classifier. In *Proc. MLC96*, pages 105–112, 1996.

[17] R. O. Duda and P. E. Hart. *Pattern classification and Scene Analysis.* Wiley & Sons Inc., 1973.

[18] C. Faloutsos and D. Oard. A Survey of Information Retrieval and Filtering Methods. Technical Report CS-TR-3541, University of Maryland, 1995.

[19] D. H. Foley and J. W. Samon Jr. An Optimal Set of Discriminant Vectors. In *IEEE Trans. Comput. C-24*, pages 281–289, 1975.

[20] L. N. Foner. Yenta: A Multi-Agent, Referral Based Matchmaking System. In *The First International Conference on Autonomous Agents (Agents '97)*, California, 1997.

[21] C. Fox. *Lexical Analysis and Stoplist*, pages 102–130. Prentice Hall, 1992. edited by W.B. Frakes and R. Baeza-Yates.

[22] W. B. Frakes. Stemming Algorithms. In *Information Retrieval Data Structure and Algorithms*, pages pp.131–160. Prentice Hall, 1992. edited by W.B. Frakes and R. Baeza-Yates.

[23] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *Proc. Second European Conference on Computational Learning Theory*, pages pp 23–37, 1995.

[24] N. Friedman and M. Goldszmidt. Building Classifiers Using Bayesian Networks. In *AAAI, vol. 2*, pages 1277–1284, 1996.

[25] N. Friedman, D. Heckerman, M. Goldszmidt, and S. Russell. Challenge: Where is the Impact of Bayesian Networks in Learning? In *Proc. Fifteenth International Joint Conference on Artificial Intelligence (IJCAI)*, 1997.

[26] G. W. Gates. The reduced nearest neighbor rule. In *IEEE Trans. Information Theory, IT-18*, pages 431–433, 1972.

[27] B. Gelfand, M. Wulfekuhler, and W. F. Punch. Automatic Concept Extraction From Plain Text. Technical report, Michigan State University, 1998.

[28] G. Giacinto and F. Roli. Adaptive Selection of Image Classifiers. In *Proc. of ICIAP'97 (Springer Verlag Lecture Notes in CS Vol. 1310)*, pages 38–45, 1997.

[29] G. Golub and W. Kahan. Calculating the singular values and pseudo-inverse of a matrix. In *SIAM Journal Numerical Analysis, Series B, Vol. 2, No. 2*, pages 205–224, 1965.

[30] Y. Hamamoto, T. Kanaoka, and S. Tomita. On a Theoretical Comparison between the Orthonormal Discriminant Vector Method and Discriminant Vector. In *Pattern Recognition, Vol. 26, No. 12*, pages 1863–1867, 1993.

[31] S. Haykin. *Neural Networks.* Macmillan College, 1994.

[32] D. Heckerman. A Tutorial on Learning with Bayesian Networks. Technical Report MSR-TR-95-06, Microsoft Research, 1996.

[33] A. K. Jain and B. Chandrasekaran. *Dimensionality and Sample Size Considerations in Pattern Recognition Practice. In Handbook of Statistics*, pages 835–855. North-Holland, 1982. edited by P.R. Krishnaiah and L.N. Kanal.

[34] A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data.* Prentice Hall, 1988.

[35] A. K. Jain and D. Zongker. Feature Selection: Evaluation, Application and Small Sample Performance. In *IEEE Trans. PAMI, Vol. 19, No. 2*, pages 153–157, 1997.

[36] T. Joachims, D. Freitag, and T. Mitchell. WebWatcher: A Tour Guide for the World Wide Web. In *Proc. IJCAI'97*, 1997.

[37] D. Koller and M. Sahami. Toward Optimal Feature Selection. In *Proceedings of the 13th International Conference on Machine Learning (ML)*, pages 284–292, Bari, Italy, 1996.

[38] K. Lang. NewsWeeder: Learning to Filter Netnews. In *Proc. 12th International Conference on Machine Learning*, pages 331–339, Tahoe City, CA, 1995.

[39] L. S. Larkey and W. B. Croft. Combining Classifiers in Text Classification. In *Proc. of SIGIR'96*, pages 289–297, 1996.

[40] Y. Lashkari, M. Metral, and P. Maes. Collaborative Interface Agents. In *Proceedings of AAAI '94 Conference*, Seattle, Washington, 1994.

[41] J. Lee and S. M. Chung. Information Discovery on the Internet: Beyond Search Engines. In *SERI Journal, Vol. 2, No. 1*, pages 33–45, 1998.

[42] D. D. Lewis. Evaluating Text Categorization. In *Proc. of Speech and Natural Language Workshop*, pages 312–318, Kaufmann, San Mateo, CA, 1991.

[43] D. D Lewis. *Representation and Learning in Information Retrieval.* PhD thesis, University of Massachusetts, 1992.

[44] D. D. Lewis and M. Ringutte. A comparison of Two Learning Algorithms for Text Categorization. In *Third Annual Symposium on Document Analysis and Information Retrieval*, pages 81–93, Las Vegas, NV, 1994.

[45] H. Lieberman. Letizia: An Agent That Assists Web Browsing. In *International Joint Conference on Artificial Intelligent*, Montreal, 1995.

[46] C. Lynch. Searching the Internet. In *Scientific American*, pages 52–56, 1997.

[47] Y. Ming. Sampling strategies and learning efficiency in text categorization. In *Proc. 1996 AAAI Spring Symposium on Machine Learning in Information Access*, pages 88–95, Stanford, 1996.

[48] T. Mitchell. *Machine Learning.* McGraw-Hill, 1997.

[49] D. Mladenic. Personal WebWatcher: Design and Implementation. Technical Report IJS-DP-7472, Carnegie Mellon University, October 1996.

[50] I. Moulinier, G. Raskinis, and J.-G. Ganascia. Text Categorization: a Symbolic Approach. In *SDAIR96*, Las Vegas, 1996.

[51] S. Deerwestera nd S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. In *Journal of the American Society for Information Science, Vol. 41, No. 6*, pages 391–407, 1990.

[52] S. A. Nene and S. K. Nayar. A simple algorithm for nearest neighbor search in high dimension. In *IEEE Trans. PAMI, vol. 19, No. 9*, pages 989–1003, 1997.

[53] H. S. Nwana. Software Agents: An Overview. In *Knowledge Engineering Review, Vol. 11, No. 3*, pages 1–40, 1996.

[54] M. J. H.S. Nwana and N. R. Jennings. Intelligent Agents: Theory and Practice. In *Engineering Review, Vol. 10, No. 2*, pages 115–152, 1995.

[55] D. W. Oard and G. Marchionini. A Conceptual Framework for Text Filtering. Technical Report EE-TR-96-25, CAR-TR-830, CLIS-TR-96-02, CS-TR-3643, University of Maryland, 1996.

[56] E. Oja. *Subspace Methods of Pattern Recognition.* Wiley, 1983.

[57] T. Okada and S. Tomita. An Optimal Orthonormal System for Discriminant Analysis. In *Pattern Recognition, Vol. 18, No. 2*, pages 139–144, 1985.

[58] M. Pazzani, J. Muramatsu, and D. Billsus. Syskill & Webert: Identifying interesting Web Sites. In *AAAI Spring Symposium on Machine Learning in Information Access Technical Papers*, Palo Alto, March 1996.

[59] M. Pazzani, L. Nguyen, and S. Mantik. Learning from Hotlists and Coldlists: Toward a WWW information Filtering and Seeking Agent. In *Proc. 6th International Conference on Tools with Artificial Intelligence*, Washington D.C., 1995.

[60] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann, San Mateo, CA, 1988.

[61] J. R. Quinlan. Induction of Decision Trees. In *Machine Learning, Vol. 1*, pages 81–106, 1986.

[62] J. R. Quinlan. Rule Induction with Statistical Data-a Comparison with Multiple Regression. In *Journal of the Operational Research Society, Vol. 38*, pages 347–352, 1987.

[63] J. R. Quinlan. *C4.5:Programs for Machine Learning.* Morgan Kaufmann, 1993.

[64] E. S. Ristad. A Natual Law of Succession. Technical Report CS-TR-495-95, Princeton University, 1995.

[65] S. M. Ross. *Probability Models.* Academic Press, 1993.

[66] M. Sahami. Learning Limited Dependence Bayesian Classifiers. In *proc. of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)*, pages 335–338, Portland, Oregon, 1996.

[67] G. Salton. *Automatic Text Processing.* Addison-Wesley, 1989.

[68] G. Salton, J. Allan, and C. Buckley. Automatic Structuring and Retrieval of Large Text Files. In *Communications of the ACM, Vol. 37, No. 2*, pages 97–108, 1994.

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

[70] S.Chen. *Learning-Based Vision and its Application in Indoor Navigation.* PhD thesis, Michigan State University, 1998.

[71] H. Schutze, D. A. Hull, and J. O. Pedersen. A Comparison of Classifiers and Document Representations for the Routing Problem. In *SIGIR95*, pages 229–237, Washington D.C., 1995.

[72] E. Selberg and O. Etzioni. The MetaCrawler Architecture for Resource Aggregation on the Web. In *IEEE Expert, Vol. 12, No. 1*, pages 8–14, 1997.

[73] B. D. Sheth. A learning Approach to Personal Information Filtering. Master's thesis, Massachusetts Institute of Technology, 1994.

[74] A. F. Smeaton and F. Crimmins. Using a Data Fusion Agent for Searching the WWW. In *Sixth International World Wide Web Conference*, Santa Clara, CA., 1997.

[75] H. Turtle and W. B. Croft. Inference Networks for Document retrieval. In *Proc. of the 13th Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval*, pages 1–24, Brussels, Belgium, 1990.

[76] H. Turtle and W. B. Croft. Evaluation of an Inference Network-based Retrieval Model. In *ACM Trans. on Information Systems, Vol. 9, No. 3*, pages 187–222, 1991.

[77] Wallace C. Koehler, Jr. An End User's View of Mining the Web: Focused and Satisfied Internet Search and Retrieval Strategies. In *INET 97 proceedings, Education 3*, 1997[2].

[78] A. R. Webb and D. Lowe. The Optimized Internal Representation of Multilayer Classifier Networks Performs Nonlinear Discriminant Analysis. In *Neural Networks, Vol. 3*, pages 367–375, 1990.

[79] S. A. Weiss, S. Kasif, and E. Brill. Text Classification in USENET Newsgroup: A Progress Report. In *AAAI Spring Symposium on Machine Learning in Information Access Technical Papers*, Palo Alto, March 1996.

[2]http://www.isoc.org/isoc/whatis/conferences/inet/97/proceedings/D3/INDEX.HTM

[80] K. Woods, W. P. Kegelmeyer, and K. Bowyer Jr. Combination of Multiple Classifiers Using Local Accuracy Estimates. In *IEEE Trans. PAMI, Vol. 19, No. 4*, pages 405–410, 1997.

[81] M. R. Wulfekuhler and W. F. Punch. Finding Salient Feature for Personal Web Page Categories. In *Sixth International World Wide Web Conference*, Santa Clara, CA., 1997.

[82] T. W. Yan and H. C. Molina. SIFT-A Tool for Wide-Area Information Dissemination. In *Proc. 1995 USENIX Technical Conference*, pages 177–186, 1995.