FRAMEWORK FOR WEB DOCUMENT CLASSIFICATION BASED ON NAÏVE BAYESIAN CLASSIFIER USING VOTING METHOD

A DISSERTATION

Submitted in partial fulfillment of the requirements for the award of the degree of

MASTER OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

By

G. RAJESH

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
ROORKEE - 247 667 (INDIA)

JUNE, 2008

Candidate's Declaration

I hereby declare that the work being presented in the dissertation report titled "**Framework for Web Document Classification based on Naïve Bayesian Classifier using Voting Method**" in partial fulfillment of the requirement for the award of the degree of **Master of Technology** in **Computer Science and Engineering**, submitted in the Department of Electronics and Computer Engineering, Indian Institute of Technology Roorkee, is an authenticate record of my own work carried out under the guidance of Dr. R. C. Joshi, Professor, Department of Electronics and Computer Engineering, Indian Institute of Technology Roorkee.

I have not submitted the matter embodied in this dissertation report for the award of any other degree.

Dated: 28/6/08

Place: IIT Roorkee.

(G. Rajesh)

Certificate

This is to certify that above statements made by the candidate are correct to the best of my knowledge and belief.

Dated: 28.6.08

Place: IIT Roorkee.

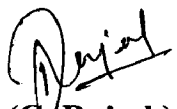
Dr. R. C. Joshi,
Professor,
Department of Electronics and
Computer Engineering, IIT Roorkee,
Roorkee -247667 (India).

ACKNOWLEDGEMENTS

First of all, I would like to thank Prof. Dr. R. C. Joshi, my supervisor, for getting me interested in the area of data mining and Web page classification, and his consistent and sound critiques of my ideas and work. I am particularly grateful for his enthusiasm, constant support and encouragement. My dissertation could not have been done reasonably without insightful advice from him. Working under his guidance will always remain a cherished experience in my memory. I am also thankful to Indian Institute of Technology Roorkee for giving me this opportunity.

I am also grateful to the staff of software laboratory and research scholar's laboratory of department of electronic and computer engineering, for their kind cooperation extended by them in the execution of this dissertation. I am also thankful to all my friends who helped me directly and indirectly in completing this dissertation.

Most importantly, I would like to extend my deepest appreciation to my family for their love, encouragement and moral support. Finally I thank God for being kind to me and driving me through this journey.



(G. Rajesh)

ABSTRACT

Automatic web document classification is the process of assigning a web documents to one or more predefined category. With the continuous increase of the information available in the World Wide Web (WWW) the importance of the web page classification problem grows significantly. As the information flow occurs at a high speed in the WWW, there is a need to organize it in the right manner so that a user can access it very easily. Previously the organization of information was generally done manually, by matching the document contents to some pre-defined classes. In this approach, a human expert performs the classification task, and alternatively, supervised classifiers are used to automatically classify document. In a supervised classification, manual interaction is required to create some training data before the automatic classification task takes place, thus we can reduce this human participation .

In this dissertation we propose a framework for web document classification by solving the semantic and structured keywords. The proposed system is based on Naïve Bayesian (NB) classifier using a voting method on two different feature selection methods. The system uses both latent semantic indexing (LSI) and structure-oriented weighting technique (SWT) for feature selection and, training and classification is performed using Naïve Bayesian classifier. The latent semantic indexing method projects terms and documents into a Boolean term-document matrix to find latent information in the document. At the same time, we also use the structure-oriented weighting technique which project terms and documents into weighted term-document matrix. These two features are sent to the NB classifier for training and testing respectively. Based on the output of the NB classifier, a voting method is used to determine the suitable class of the web page. By using the Voting method, we are taking the advantages of both semantic relationship between terms and documents and structure of the html document to improve the classifier accuracy. The proposed framework describes training and learning the classifier on two different feature vectors. These methods have been evaluated using yahoo directories web pages based on three parameters – recall, precision and F-measure. The results show that the proposed method works significantly better than the considering LSI features and SWT features separately.

Table of Contents

List of Figures

List of Tables

Introduction and Statement of the Problem CHAPTER 1

Today, electronic documents have turned to be the largest information source available on the World Wide Web. The web is a huge, explosive, diverse, dynamic and mostly unstructured data repository, which supplies incredible amount of information, and also raises the complexity of how to deal with the information from the different perspectives of users. The users want to have the effective search tools to find relevant information easily and precisely. With the explosive growth of information sources available on the World Wide Web, it has become increasingly necessary for users to utilize automated tools in find the desired information resources.

1.1 Introduction

Web page (web documents)[1] classification, also known as web page categorization, is the process of assigning a web page to one or more predefined category labels using machine learning algorithms and text mining techniques. 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 new test sample to a class with the highest similarity.

Due to the lack of logical organization of web pages, retrieving relevant information from the web becomes a laborious and time consuming task, and thus motivates the development of automatic web document classification systems. Automatic document classification is an active and challenging field of research, and an extensive range of algorithms has been proposed. Mostly used methods include the Decision tree method [1], k-Nearest neighbor method (kNN) [2], Neural networks (NNet) [3], Support vector machines (SVM) [4] and Naive Bayesian method (NB) [5].

The approaches to document classifications can be classified into two: manual approach and automated approach.

[1] We shall use the terms Web pages, Web documents and documents interchangeably

a. Manual approach

The traditional manual approach to classification involved the analysis of the contents of the web page by a number of domain experts and the classification was based on the textual content as is done to some extent by Yahoo Directory [7]. The sheer volume of data on the web rules out this approach. Moreover, such a classification would be subjective and hence open to question.

b. Automated approach

Automated document classification/categorization, the task is to assign an electronic document to one or more categories, based on its contents. Automated document classification tasks can be divided into two ways, supervised document classification where some external mechanism (training documents) provides information on the correct classification of documents, and unsupervised document classification, where the classification must be done entirely without reference to external information.

i. Supervised learning

Supervised learning is a machine learning technique for learning a function from training data. The training data consist of pairs of input objects (typically documents), and desired outputs. The output of the function is predicting a class label of the input object. The task of the supervised learner is to predict the value of the function for any valid input object after having seen a number of training examples (i.e. pairs of input and target output).

Solving a given problem of supervised learning (e.g. learning to recognize the given document) involves various steps.

- Determine the type of training examples. Before doing anything else, we should decide what kind of data is to be used as an example. For instance, this might be a single word, an entire line of word, or all words in the document.
- Gathering a training set, the training set needs to be characteristic of the real-world use of the function. Thus, a set of input objects is gathered and corresponding outputs are also gathered, either from human experts or from measurements.

- Determine the input feature representation of the learned function. The accuracy of the learned function depends strongly on how the input object is represented. Typically, the input object is transformed into a feature vector, which contains a number of features that are descriptive of the object. The number of features should not be too large, because of the curse of dimensionality; but should be large enough to accurately predict the output.
- Determine the structure of the learned function and corresponding learning algorithm. For example, Naïve Bayesian, Support vector machine, Artificial neural networks or Decision trees.
- Complete the design, and then run the learning algorithm on the gathered training set. Parameters of the learning algorithm may be adjusted by optimizing performance on a subset (called a *validation* set) of the training set, or via cross-validation. After parameter adjustment and learning, the performance of the algorithm may be measured on a test set that is separate from the training set.

Another term for supervised learning is classification. A wide range of classifiers are available, each with its strengths and weaknesses. Classifier performances depend greatly on the characteristics of the data to be classified. There is no single classifier that works best on all given problems. Various empirical tests have been performed to compare classifier performance and to find the characteristics of the data that determine classifier performance. Determining a suitable classifier for a given problem is however still more an art than a science.

ii. Unsupervised learning

Unlike supervised learning, unsupervised learning does not rely on predefined classes and class labeled training data. In machine learning, unsupervised learning often refers as cluster analysis, the process of grouping a set of physical or abstract objects into classes of similar objects. A cluster is a collection of data objects that are similar to one another within the same cluster and are dissimilar to the objects in other clusters. A cluster of data objects can be treated collectively as one group in many applications. As a branch of statistics, cluster analysis has been studied extensively for many years, focusing mainly on distance-based cluster analysis. Categories of clustering methods are

partitioning method, hierarchical method, density based clustering method, grid-based clustering method, and model-based clustering method.

Automated classification is needed for two important reasons. The first is the sheer scale of resources available the web and their ever-changing nature. It is simply not feasible to keep up with the pace of growth and change on the web through manual classification without expending immense time and effort. The second reason is that classification itself is a subjective activity. Different classification tasks are needed for different applications. No single classification scheme is suitable for all applications.

1.2 Problem statement

This dissertation is aimed to classifying (predict the category/group) unlabeled web documents. The problem would be solved by taking a large set of labeled (the category/group of the document) web documents and building a two different feature vectors called latent semantic indexing features and structure oriented weighting techniques features from labeled documents. And then Naïve Bayesian classifier is used to classify an unlabeled example based on the information learned from the labeled examples.

1.3 Scope of the dissertation

The scope of the web document classification is important because of the vast amount of information available on the internet. The ability to classify unlabeled documents would lead to easier access for those doing research in a specific area. Search engines like Google could return better results if information was better organized. In this dissertation we illustrate how simple model like a Naïve Bayesian can lead to accurate results.

1.4 Organization of the Report

This dissertation proposes a framework for web document classification based on Naïve Bayesian classifier using voting method. The organization of the dissertation is as follows:

Chapter 2 gives the background and literature review of web mining, description of some well known feature extraction, feature selection techniques and various supervised

learning algorithms in this field. Chapter 3 is the description of proposed framework for web document classification. Chapter 4 gives detailed description of methods and algorithms used in the proposed framework for web document classification. Chapter 5 discusses the performance metrics used, the data set used for the training and testing purpose, the performance of the system and graphs depicting the performance. Chapter 6 concludes the dissertation work and gives suggestions for future work.

Background and Literature Review CHAPTER 2

In this chapter we describes the background and literature review related to web document classification problems and methods used to solve the problems. Web page categorization/classification is one of the essential techniques for web mining. First we describe web mining and introduction to basic document classification framework, then we describe various methods to solve the document classification problem such as different feature selection methods, various machine learning algorithms and text mining techniques and finally we discuss related work.

2.1 Web Mining

A great challenge of web mining arises from the increasingly large web pages and the high dimensionality associated with natural language. Since classifying web pages of an interesting class is often the first step of mining the web.

Etzioni in [8] first proposed the term of Web mining in his paper. In this paper, he claimed the Web mining is the use of data mining techniques to automatically discover and extract information from World Wide Web documents and services. The general process of the web mining is shown in Fig. 2.1. Web mining categorize into three areas: web content mining, web structure mining, and web usage mining as shown in Fig. 2.2. Web content mining focuses on the discovery/retrieval of the useful information from the Web contents/data/documents, while the Web structure mining emphasizes to the discovery of how to model the underlying link structures of the Web. Web usage mining is which mainly describes the techniques that discover the user's usage pattern and try to predict the user's behaviors.

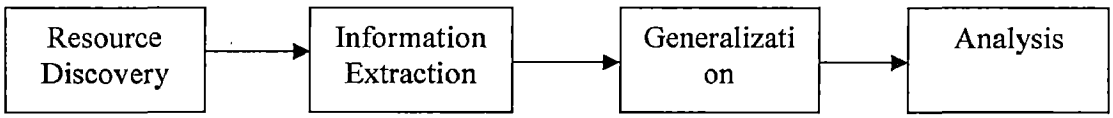


Fig. 2.1 General Process of Web mining

- Resource Discovery: The task of retrieving the intended information from Web.
- Information Extraction: Automatically selecting and pre-processing specific information from the retrieved Web resources.
- Generalization: Automatically discovers general patters at the both individual Web sites and across multiple sites.
- Analysis: Analyzing the mined pattern.

2.2 Web Mining Taxonomy

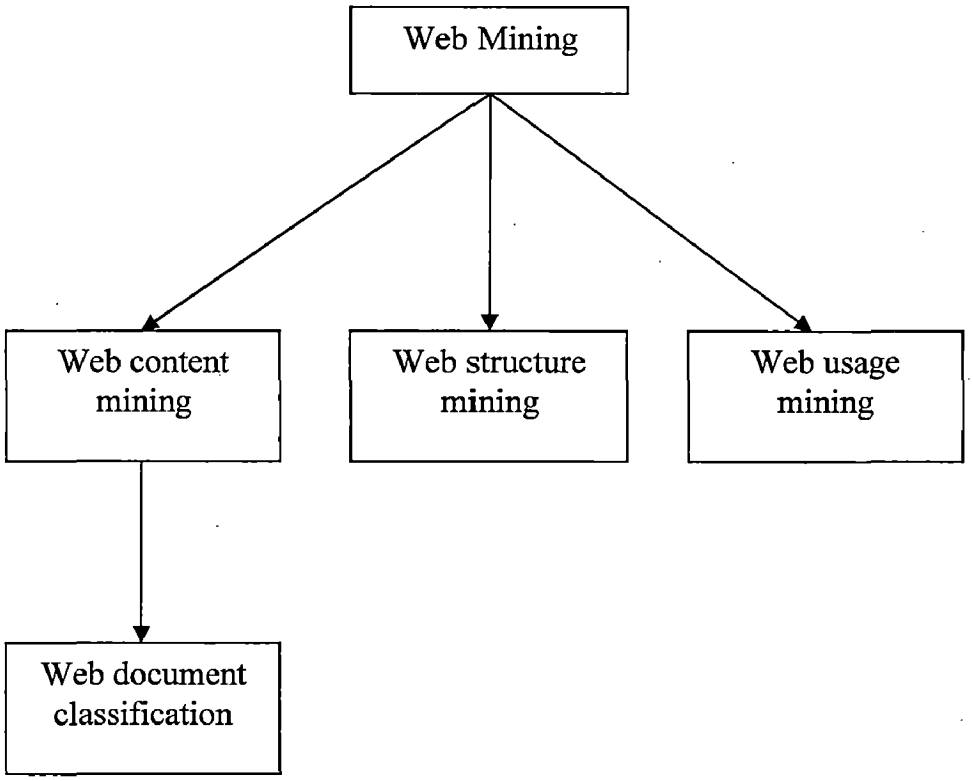


Fig. 2.2 Web Mining Taxonomy

Table 2.1 Comparisons of Web Mining Categories

	WEB MINING			
	Web content mining		Web structure mining	Web usage mining
	IR VIEW	DB VIEW		
View Of Data	Unstructured, Semi-structured	Semi-structured web site as DB	Link structure	Interactivity
Main Data	Text documents, Hypertext documents	Hypertext document	Link structure	Server logs, Browser logs
Representation	Terms, Phrases, Concepts and ontology relational	Edge labeled graph, Relational	Graph	Relational table Graph
Method	Machine learning, Statistical (including NLP)	Algorithms, Association rules, NLP	Proprietary algorithms	Machine learning Statistical (Modified) Association rules
Application Categories	Categorization, Clustering, Finding extraction rules, Finding patterns in text	Finding frequent substructures, web site schema design	Categorization ,Clustering	Site construction, adaptation and management Marketing User modeling

a. Web Content Mining

Web content mining describes the automatic search of information resource available online, and involves mining web data contents. In the web mining domain, web content mining essentially is an analog of data mining techniques for relational databases, since it is possible to find similar types of knowledge from the unstructured data residing in web documents. The web document usually contains several types of data, such as text, image, audio, video, metadata and hyperlinks. Some of them are semi-structured such as HTML documents or a more structured data like the data in the tables or database generated HTML pages, but most of the data is unstructured text data. The unstructured characteristic of web data forces the web content mining towards a more complicated approach.

The web content mining is differentiated from two different points of view: Information Retrieval View and Database View. Kosala, R. et. al. in [9] summarized the research works done for unstructured data and semi-structured data from information retrieval view. For the semi-structured data, all the works utilize the HTML structures inside the documents and some utilized the hyperlink structure between the documents for document representation. As for the database view, in order to have the better information management and querying on the web, the mining always tries to infer the structure of the Web site of to transform a Web site to become a database.

Chakrabarti, S. in [10], provides an in-depth survey of the research on the application of the techniques from machine learning, statistical pattern recognition, and data mining to analyzing hypertext. It's a good resource to be aware of the recent advances in content mining research.

Multimedia data mining is part of the content mining, which is engaged to mine the high-level information and knowledge from large online multimedia sources. Multimedia data mining on the web has gained many researchers' attention recently. Working towards a unifying framework for representation, problem solving, and learning from multimedia is really a challenge, this research area is still in its infancy indeed, many works are waiting to be done.

b. Web structure mining

Most of the web information retrieval tools only use the textual information, while ignore the link information that could be very valuable. The goal of web structure mining is to generate structural summary about the web site and web page. Technically, Web content mining mainly focuses on the structure of inner-document, while web structure mining tries to discover the link structure of the hyperlinks at the inter-document level. Based on the topology of the hyperlinks, Web structure mining will categorize the web pages and generate the information, such as the similarity and relationship between different web sites.

Web structure mining can also have another direction – discovering the structure of Web document itself. This type of structure mining can be used to reveal the structure (schema) of web pages; this would be good for navigation purpose and make it possible to

compare/integrate web page schemas. This type of structure mining will facilitate introducing database techniques for accessing information in web pages by providing a reference schema.

Madria, S. K. et. al. in [11], gives a detailed description about how to discover interesting and informative facts describing the connectivity in the web subset, based on the given collection of interconnected web documents. The structural information generated from the web structure mining includes the following: the information measuring the frequency of the local links in the web tuples in a web table; the information measuring the frequency of web tuples in a web table containing links that are interior and the links that are within the same document; the information measuring the frequency of web tuples in a web table that contains links that are global and the links that span different web sites; the information measuring the frequency of identical web tuples that appear in a web table or among the web tables.

In general, if a web page is linked to another web page directly, or the web pages are neighbors, we would like to discover the relationships among those web pages. The relations maybe fall in one of the types, such as they related by synonyms or ontology, they may have similar contents, and both of them may sit in the same web server therefore created by the same person. Another task of web structure mining is to discover the nature of the hierarchy or network of hyperlinks in the web sites of a particular domain. This may help to generalize the flow of information in Web sites that may represent some particular domain; therefore the query processing will be easier and more efficient.

Web structure mining has a nature relation with the web content mining, since it is very likely that the web documents contain links, and they both use the real or primary data on the web. It's quite often to combine these two mining tasks in an application.

c. Web Usage Mining

Web usage mining tries to discover the useful information from the secondary data derived from the interactions of the users while surfing on the web. It focuses on the techniques that could predict user behavior while the user interacts with web. Cooley, R. et. al. [12], abstract the potential strategic aims in each domain into mining goal as:

prediction of the user's behavior within the site, comparison between expected and actual web site usage, adjustment of the web site to the interests of its users. There are no definite distinctions between the web usage mining and other two categories. In the process of data preparation of web usage mining, the web content and web site topology will be used as the information sources, which interacts web usage mining with the web content mining and web structure mining. Moreover, the clustering in the process of pattern discovery is a bridge to web content and structure mining from usage mining. Web mining is the application of data mining techniques to discover usage patterns from web data, in order to understand and better serve the needs of web-based applications.

2.3 Introduction to Document Classification

The document classification is applying to a lot of fields at present, for example, information retrieval in library science, automatic news classification. The Internet prevails in recent years, so digital document grow fast. Therefore, document classification becomes an important technology for information overload today, for example, web pages classification, multimedia document classification, automatic email classification. Document classification usually divides into two stages: the training stage and the classifying stage. In the training stage, we first randomly select a part of documents for training sample and extract the features from the training samples. These features are used to represent the training documents and they are input to classifier. Let classifier discern the category of the documents. In the classifying stage, unknown documents are sent to classifier to classify. According the training model is used to determine the category of the unknown document. The basic procedures show in Fig. 2.3. In this chapter, we will introduce the related technology and the research of automatic document classification.

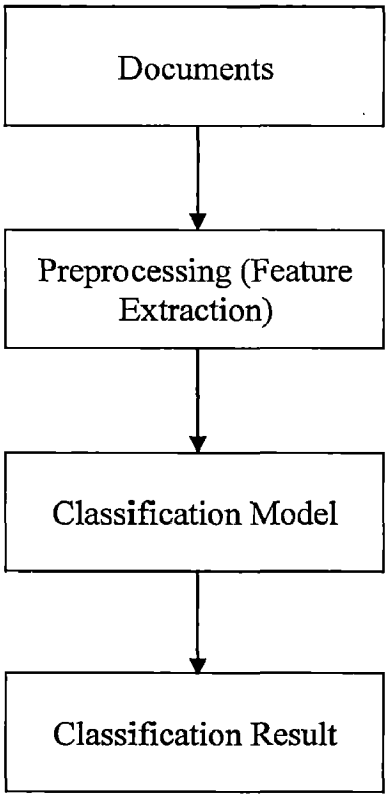


Fig 2.3 The Basic Classification Procedure

2.4 Taxonomy of classification models

The general problem of web page classification can be divided into multiple sub-problems: subject classification, functional classification, sentiment classification, and other types of classification. Subject classification is concerned about the subject or topic of a web page. For example, judging whether a page is about "arts", "business" or "sports" is an instance of subject classification. Functional classification cares about the role that the web page plays. For example, deciding a page to be a "personal homepage", "course page" or "admission page" is an instance of functional classification. Sentiment classification focuses on the opinion that is presented in a web page, i.e., the author's attitude about some particular topic. Other types of classification include genre classification, search engine spam classification etc.

- Based on the number of classes in the problem, classification can be divided into binary classification and multi-class classification, where binary classification categorizes instances into exactly one of two classes as in Fig. 2.4, multi-class classification deals with more than two classes.

- Based on the number of classes that can be assigned to an instance, classification can be divided into single-label classification and multi-label classification. In single-label classification, one and only one class label is to be assigned to each instance. While in multi-label classification, more than one class can be assigned to an instance. If a problem is multi-class, say four-class classification, it means four classes are involved, say Arts, Business, Computers, and Sports. It can be single-label, where exactly one class label can be assigned to an instance as in Fig. 2.5 or multi-label, where an instance can belong to any one, two, or all of the classes as in Fig. 2.6.
- Based on the type of class assignment, classification can be divided into hard classification and soft classification. In hard classification, an instance can either be or not be in a particular class without an intermediate state while in soft classification, an instance can be predicted to be in some class with some likelihood (often a probability distribution across all classes) as in Fig 2.7.
- Based on the organization of categories, web page classification can also be divided into flat classification and hierarchical classification. In flat classification, categories are considered parallel, i.e., all the categories are at one level and one category does not supersede another as shown in Fig 2.8. While in hierarchical classification, the categories are organized in a hierarchical tree-like structure, in which each category may have a number of subcategories as shown in Fig. 2.9.

Classification plays a vital role in many information management and retrieval tasks. On the Web, classification of page content is essential to focused crawling, to the assisted development of web directories, to topic-specific web link analysis, and to analysis of the topical structure of the Web. Web page classification can also help improve the quality of web search. The uncontrolled nature of web content presents additional challenges to web page classification as compared to traditional text classification, but the interconnected nature of hypertext also provides features that can assist the process.

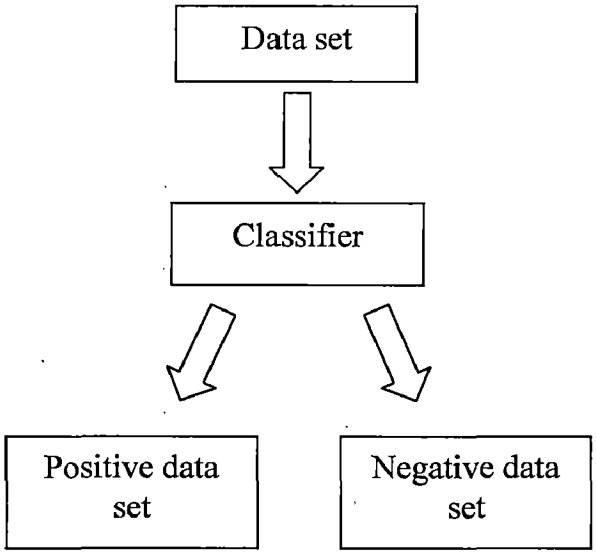


Fig. 2.4 Binary Classification

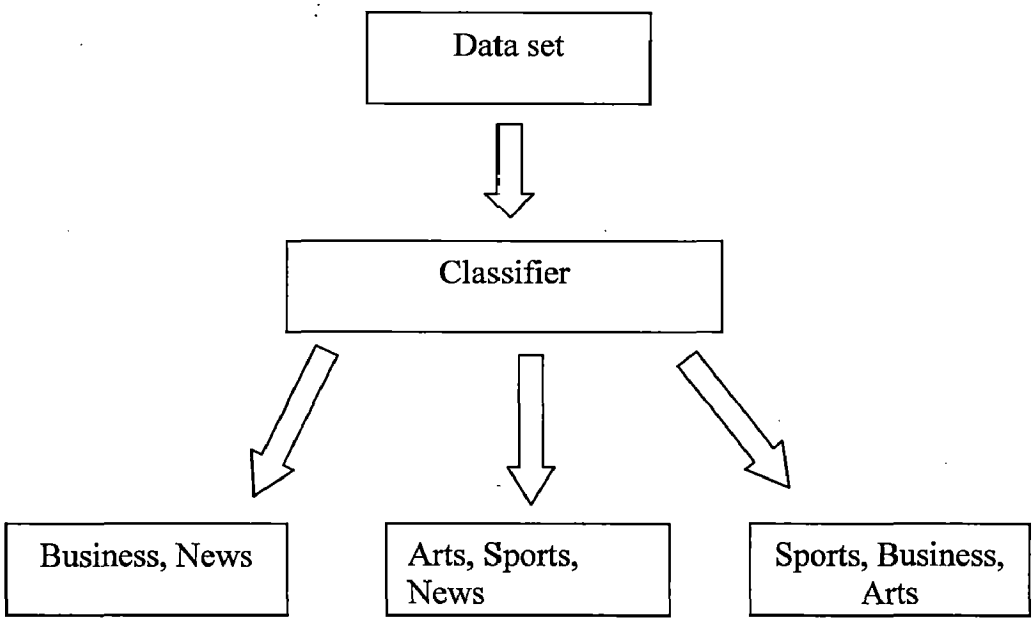


Fig. 2.5 Multi Class, Multi Label, Hard Classification

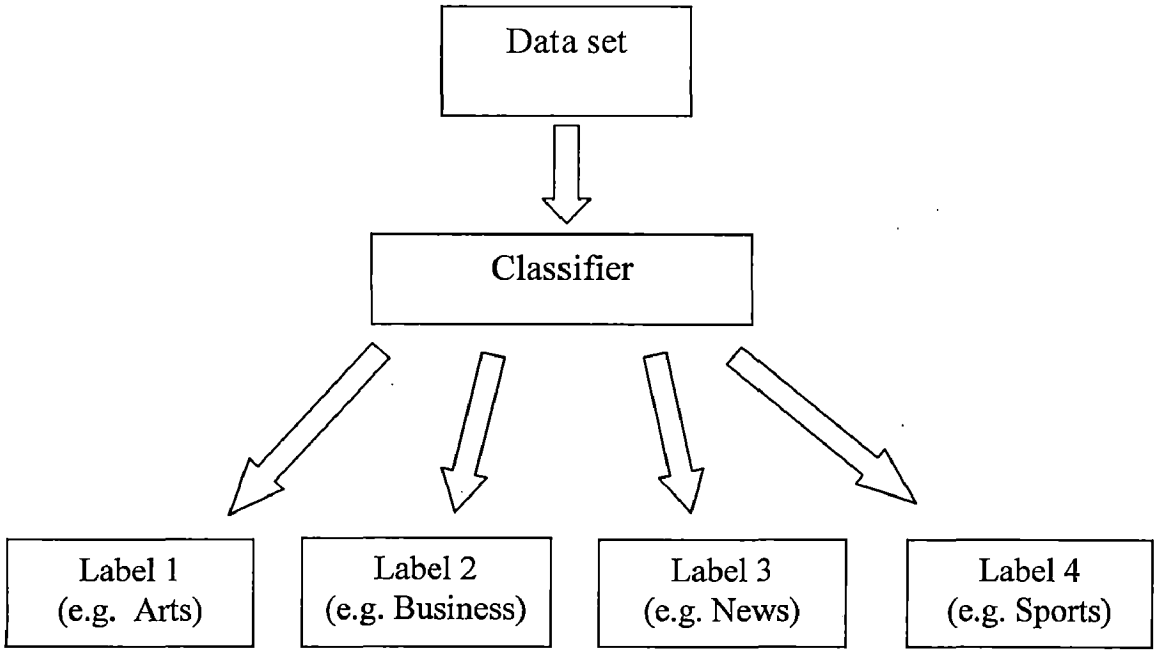


Fig. 2.6 Multi Class, Single Label, Hard Classification

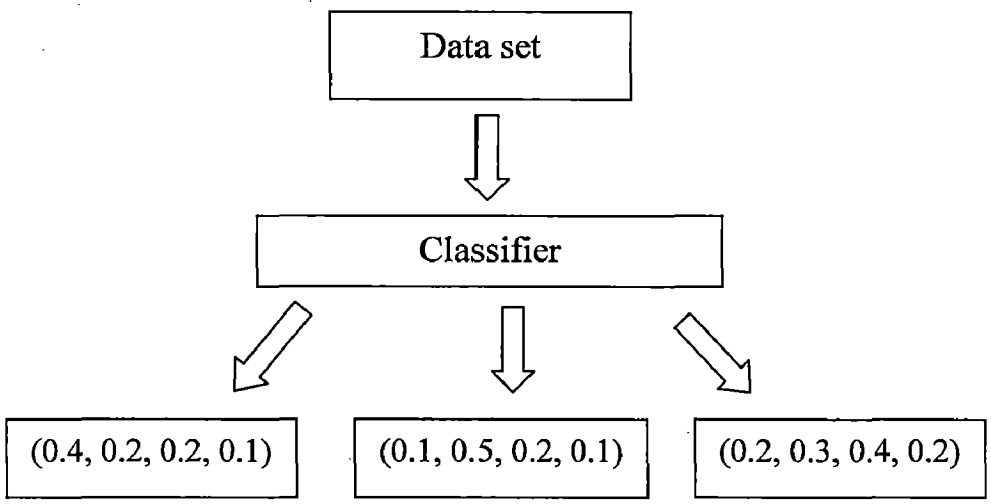


Fig. 2.7 Multi Class, Multi Label, Soft Classification

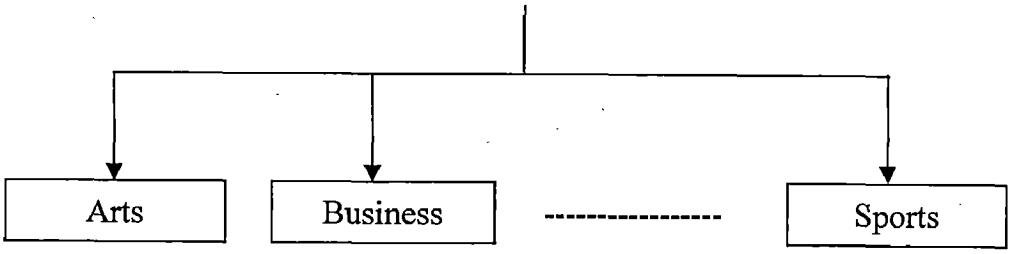


Fig. 2.8 Flat Classification

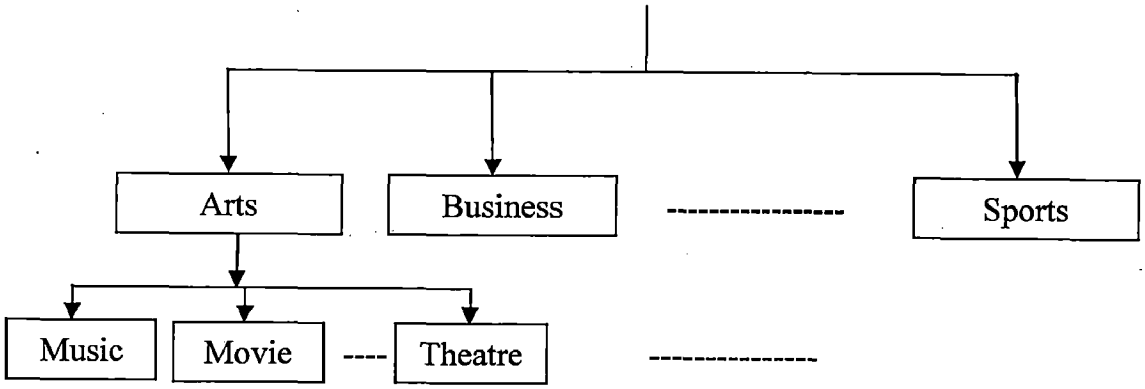


Fig. 2.9 Hierarchical Classification

2.5 Document Representation

2.5.1 Boolean Model

Boolean model is based on fuzzy set theory and Boolean algebra classification model, where only consider the keywords occur or not [6]. The weight of a keyword uses a binary string to represent. Therefore, it also is called a binary model. Its representation is: $W \in \{0, 1\}$

0: The keyword W does not appeared.

1: The keyword W appeared.

A query document q be found after transforms Boolean function to calculate similar measure *sim(q, d)*, it has two results:

1: The query document q is related to document d.

0: The query document q does not relate document d.

The advantage of this model is easy to represent and understand its means. But if document content is too long or too short, it cannot clear represent its features. Therefore

most classification methods applying the weight of keyword to represent their different importance value to promote the performance of information retrieval.

2.5.2 Vector Space Model

Vector space model improves Boolean model, only used 0 and 1 to represent weight [13]. It measures similarity between document and document to judge which category of the document belongs to.

A query vector: $q = (w_{1q}, w_{2q}, ..., w_{tq})$

Where

q: A query document

t: The number of keywords in the query document q

Classified document vector j: $d_j = (w_{1j}, w_{2j}, ..., w_{tj})$

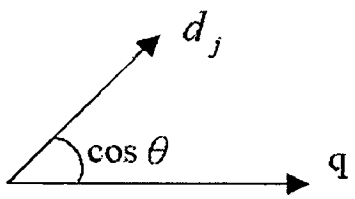


Fig. 2.10 cos θ is a similarity between d_j and q

Fig. 2.10 shows that the similarity d_j between and q is a cosine angle θ.
The cosine similarity formula is :

$$sim(d_j, q) = \frac{\vec{d}_j \bullet \vec{q}}{|\vec{d}_j| \bullet |\vec{q}|} = \frac{\sum_{i=1}^{t} W_{i,j} \times W_{i,q}}{\sqrt{\sum_{i=1}^{t} W_{i,q}^2} \times \sqrt{\sum_{i=1}^{t} W_{i,q}^2}}$$

The advantage of Vector space model is each keyword has different weight to represent their importance. It is effective to promote performance of information retrieval.

2.6 Feature Extraction

Feature extraction in web document classification is used to eliminate redundant and irrelevant term from the document. There are many terms in a document, include important keywords and irrelevant common term among these terms. The important terms can represent this document, but irrelevant terms do not any help to classify, it cause interference to classify. The relationship of term frequency and importance shows in Figures 2.11.

The terms in A area are high frequencies but they do not have any meaning. Their importance is lower, For example, the words "is", "to", the kinds of these words are called stop words. In general, we will remove stop words before extract text features. The terms in B area are keywords that we extract because the terms in this area not only have high frequency but also have high importance. These terms can apply to represent document. In C area, these terms have lower frequencies and lower importance, it cannot represent document. Therefore, we do not consider the terms in this area.

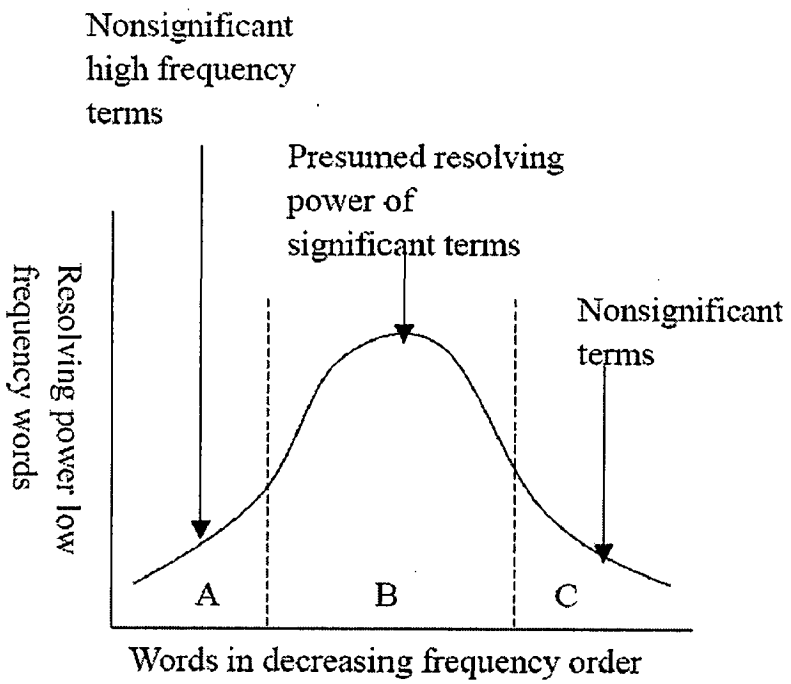


Fig. 2.11 The Relation between Term Frequency and Importance

How to extract important keywords in text features is a very important issue. In general, the extraction methods have: remove stop words, term frequency statistics.

2.7 Feature Selection Method

A major characteristic, or difficulty, of document classification problems is the high dimensionality of the feature space. The native feature space consists of the unique terms that occur in documents, which can be tens or hundreds of thousands of terms for even a moderate sized document collection; this is prohibitively high for many learning algorithms. Few neural networks, for example, can handle such a large number of input nodes. Naïve Bayesian classifier, as another example, will be computationally intractable unless an independence assumption among features is imposed. It is highly desirable to reduce the native space without sacrificing classification accuracy. It is also desirable to achieve such a goal automatically, i.e., no manual definition or construction of features is required.

Automatic feature selection methods include the removal of non- informative terms according to corpus statistics, and the construction of new features which combine lower level features (terms) into higher level orthogonal dimensions. Lewis, D. et. al. [14], used an information gain measure to aggressively reduce the document vocabulary in a naïve Bayesian model and a decision tree approach to binary classification. Wiener, E. et. al. [15], used mutual information and a χ^2 statistic to select features for input to neural networks.

We discuss five methods, each of which uses a term-goodness criterion threshold to achieve a desired degree of term elimination form the full vocabulary of a document corpus. These methods are as follows:

i. Document Frequency (DF) thresholding
ii. Information Gain (IG)
iii. Mutual Information (MI)
iv. χ^2 statistic (CHI)
v. Term Strength (TS)

i. Document Frequency (DF) thresholding

Document frequency is the number of documents in which a term occurs. We computed the document frequency for each unique term in the training corpus and removed from the feature space those terms whose document frequency was less than some predetermined threshold. The basic assumption is that rare terms are either non-

informative for category prediction, or not influential in global performance. In either case removal of rare terms reduces the dimensionality of the feature space. Improvement in classification accuracy is also possible if rare terms happened to be noise terms.

DF thresholding is the simplest technique for vocabulary reduction. It easily scales to very large corpora, with a computational complexity approximately linear in the number of training documents. However, it is usually considered an ad hoc approach to improve efficiency, not a principled criterion for selecting predictive features. Also, DF is typically not used for aggressive term removal because of a widely received assumption in information retrieval. That is, low DF terms are assumed to be relatively informative and therefore should not be removed aggressively. We will re-examine this assumption with respect to text categorization tasks.

ii. Information Gain (IG)

Information gain is frequently employed as terms goodness criterion in the field of machine learning [6]. It measures the number of bits of information obtained for category prediction by knowing the presence or absence of a term in a document. Let $\{c_i\}_{i=1}^{m}$ denote the set categories in the target space. The information gain of term t is defined to be:

$$G(t) = -\sum_{i=1}^{m} P_r(c_i)\log P_r(c_i) + P_r(t)\sum_{i=1}^{m} P_r(c_i \mid t)\log P_r(c_i \mid t)$$
$$+ P_r(\bar{t})\sum_{i=1}^{m} P_r(c_i \mid \bar{t})\log P_r(c_i \mid \bar{t})$$

This definition is more general than the one employed in binary classification models. The general form of classification problem usually have a m-ary category space (where m may be up to tens of thousands), and need to measure the goodness of a term globally with respect to all categories on average.

iii. Mutual Information (MI)

Mutual information is a criterion commonly used in statistical language modeling of word associations and related application [6]. If on considers the two way contingency table of a term t and a category c, where A is the number of times t and c co-occurs, B is the number of times the t occurs without c, C is the number of times c occurs without t,

and N is the total number of documents, then the mutual information criterion between t and c is defined to be

$$I(t,c) = \log \frac{P_r(t \wedge c)}{P_r(t) \times P_r(c)}$$

and is estimated using

$$I(t,c) \approx \log \frac{A \times N}{(A+C) \times (A+B)}$$

$I(t,c)$ has a natural value of zero if t and c are independent. To measure the goodness of a term in a global feature selection, we combine the category specific scores of a term into two alternate ways:

$$I_{avg}(t) = \sum_{i=1}^{m} P_r(c_i) I(t,c_i)$$

$$I_{\max}(t) = \max_{i=1}^{m} \{I(t,c_i)\}$$

A weakness of mutual information is that the score is strongly influenced by the marginal probabilities of terms, as can be seen in this equivalent form:

$$I(t,c) = \log P_r(t \mid c) - \log P_r(t)$$

For terms with an equal conditional probability $P_r(t \mid c)$, rare terms will have a higher score than common terms. The scores, therefore, are not comparable across terms of widely differing frequency.

iv. χ^2 Statistic (CHI)

The χ^2 statistic measures the lack of independence between t and c and can be compared to the χ^2 distribution with one degree of freedom to judge extremeness. Using the two-way contingency table of a term t and a category c, where A is the number of times t and c co-occurs, B is the number of times the t occurs without c, C is the number of times c occurs without t, D is the number of times neither c nor t occurs, and N is the total number of documents, the term goodness measure is defined to be:

$$\chi^2(t,c) = \frac{N \times (AD - CB)^2}{(A+C) \times (B+D) \times (A+B) \times (C+D)}$$

The χ^2 statistic has a natural value of zero if t and c are independent. Compute for each category the χ^2 statistic between each unique term in training corpus and that category, and then combine the category specific scores of each term into two scores:

$$\chi^2_{avg}(t) = \sum_{i=1}^{m} P_r(c_i)\chi^2(t,c_i)$$

$$\chi^2_{\max}(t) = \max_{i=1}^{m}\{\chi^2(t,c_i)\}$$

A major difference between CHI and MI is that χ^2 is a normalized value, and hence χ^2 values are comparable across terms for the same category. However, this normalization breaks down (can no longer be accurately compared to the χ^2 distribution) if any cell in the contingency table is lightly populated, which is the case for low frequency terms. Hence, the χ^2 statistic is known not to be reliable for low frequency terms [6]

v. Term Strength (TS)

This method estimates term importance based on how commonly a term is likely to appear in "closely-related" documents. It uses a training set of documents to derive document pairs whose similarity (measured using the cosine value of the two document vectors) is above a threshold. "Term Strength" then computed based on the estimated conditional probability that at term occurs in second half of a pair of related documents given that it occurs in the first half. Let x and y be an arbitrary pair of distinct but related documents, and t and be a term, then the strength of the term is defined to be:

$$s(t) = P_r(t \in y \mid t \in x)$$

The term strength criterion is radically different from the ones mentioned earlier. It is based on document clustering, assuming that documents with many shared words are related, and that terms in the heavily overlapping area of related documents are relatively informative. This method is not task-specific, i.e., it does not use information about term-category associations. In this sense, it is similar to the document frequency (DF) thresholding criterion, but different from the information gain (IG), mutual information (MI) and the χ^2 statistic. A parameter in the term strength calculation is the threshold on document similarity values. That is, how close two documents must be considered a related pair.

2.8 Supervised Learning Classification Methods

The early document classification is done by manual approach. With the increase of time and the prevailing of internet, no matter what the document, digital document or web pages on Internet, its speed increased very fast. Use of manual approach to classification not only consumes time but also need a lot of human resources. Therefore, we need to use automated system to automatically classify documents. Maderlechner, G. et. al. [16], used the form and content of document classify documents. They cut the whole document into text and non-text. The form of the whole document defines several entities like margins, text columns, header, footer, and so on. Most of journal publisher and business document designers will use this rule to achieve the identification of documents. In content, it uses entropy to measure the information of characters to judge the content of characters. Asirvatham, A. P. et. al. [17], proposed that web pages can classify three categories approximately: information pages, research pages, and personal homepages. According to the structure, these kinds of web pages are used to classify. For example, information pages usually have a logo on the top web page, and have navigation bar to link to others web pages. These kind of web pages click hyperlink to others web pages in a high ratio. The research web pages usually include a lot of text, formula and graphics. The graphics can use colors to detect. Personal home pages have a common scheme that it has personal data and has a picture obvious and close to the bottom will have some link to web sites that the author likes. According these properties to extract features and classify.

There are many learning methods to support web pages classification at present. These methods can divide into supervised learning and unsupervised learning. Unsupervised learning method is according each document features, automatic clustering documents with similar features. It does not previously define document category. Supervised learning method must previously define document category. Apply training sample to train a category model for each category. Then using the model decides the category of a new document. The supervised learning method includes: decision tree, Bayesian probability method, support vector machine, k nearest neighbor, and neural network. It shows in Fig. 2.12.

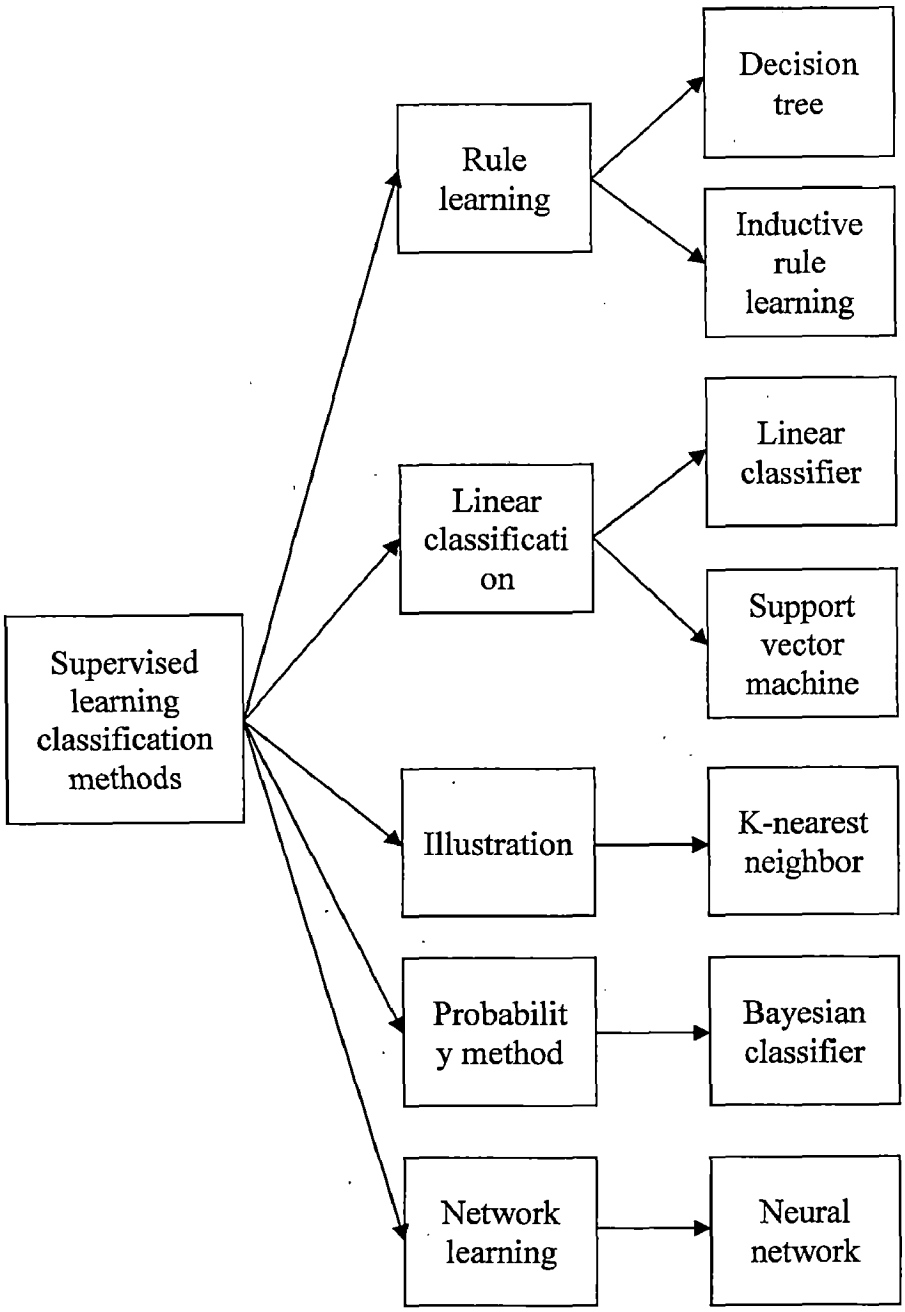


Fig. 2.12 Supervised learning classification methods

Supervised learning classification methods are decision trees, k-nearest neighbor, Bayesian approach, neural networks, regression-based methods, vector-base method etc. We illustrate some of them in the following sub-sections.

a. Decision Trees

We can build a manual categorization of the training document in a decision tree by representing a well defined true/false-queries where nodes represent questions and leafs represent the corresponding category of documents [1]. After having constructed the tree, a new document can easily be categorized by putting it in the root node of the tree

and by letting it runs to through the query structure until it reaches a certain leaf. The main advantage of decision tree is classification result can easy transform to IF-THEN relation and the output tree is easy to understand by even those who not familiar with the details of the model. The disadvantage is that while the categories are more, it will easy have mistakes. The risk is over fitting because there is an existence of an alternative tree that categorizes the training data incorrect way.

b. Bayesian Classifier

This method transforms term frequency of all keywords into condition probability. Then Bayesian probability model calculates each probability of document and category. The category that has the highest probability is this document's category.

c. k-Nearest Neighbor

The previous method is based on a learning phase but k-nearest completely skips the learning phase and categorizes on-the-fly. The categorization is often performed by comparing the category frequencies of the k-nearest documents. The closeness of the documents can be evaluated by the calculating the Euclidian distance between the two vectors. The method is simple. Hence, it does not need any resource for training the documents. The algorithm performs well even if the category of specific documents forms more than one cluster. The category may contain more than one topic [2], although there is a risk of inadequate categorization for different numbers of training documents per category.

The k-NN method measures the similarity between testing document and training document. Here, main goal is to find k number of documents with highest similarity. To judge the category of test documents from these document. The Fig. 2.13 illustrates the concept of k-NN.

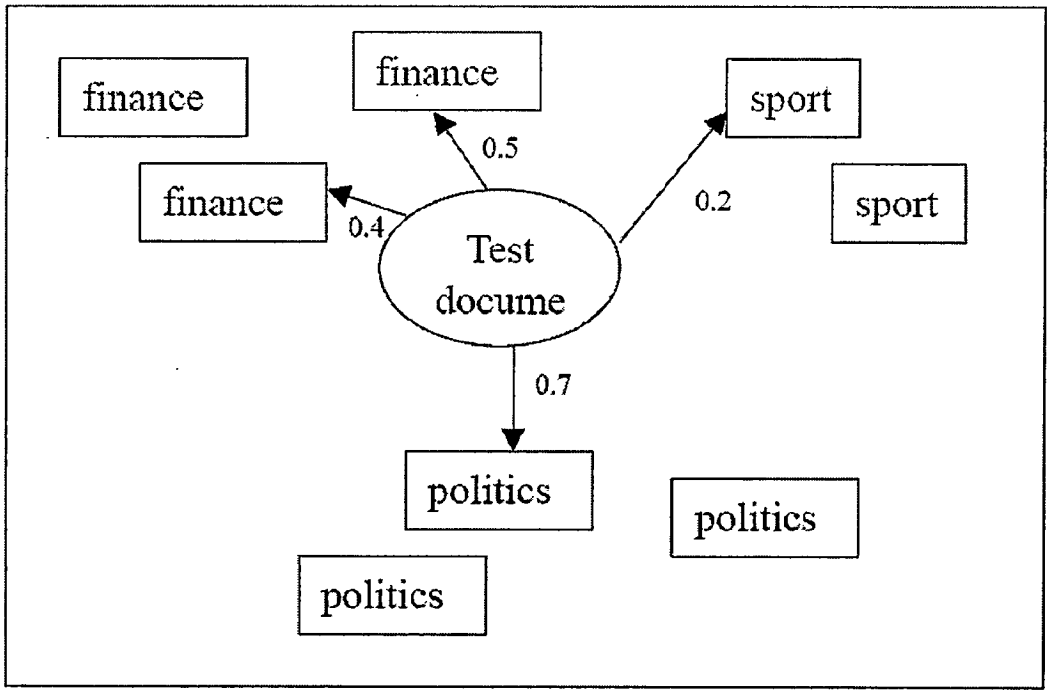


Fig. 2.13 Concept of k-Nearest Neighbor

For example, test document and nearest documents:

Sim(test, politics)=0.7

Sim(test, sport)=0.2

Sim(test, finance)=0.5+0.4

Therefore, the test document will fetch the highest similarity. This example also explains the similarity measure is the major factor that affects k-NN. If the similarity measure fails, then the effectiveness of k-NN is lower. Asirvatham, A. P. et. al. [17], proposed three stages for web site classification: web page selection, web page classification, web site classification. The k-NN is used in web page classification stage. In web page selection stage, it is to restrict the range of web site, then apply connection information assign weight for each web page in the range. Ranking the weights these web pages to obtain the top n web pages that have the higher weight values. In web pages classification stage, k-NN method calculates the likelihood score of category. In this stage, in order to raise performance, it added feature selection, such as HTML tags and new similarity. Finally, in web site classification use web page weight and likelihood of category to calculate the likelihood of each category and web site to classify web site.

The advantage of k-NN is that its classification speed is fast. The disadvantage is the test document must with each training document to calculate similarity. It has large calculation loading. The classification procedures are easy to be interfered by noise item.

d. **Artificial neural network**

Artificial neural network simulates neural network of biology. It used a lot of artificial neuron to simulate the capability of biology neural network. The artificial neurons are getting information from external environment and others artificial neurons, and operate for these information, output its result to external environment and others neurons [3]. The Back Propagation Network (BPN) is the most representative and general in artificial neural network. It is a supervised learning network, the structure shows in Fig. 2.14. The structure has three layers: input layer, hidden layer, and output layer.

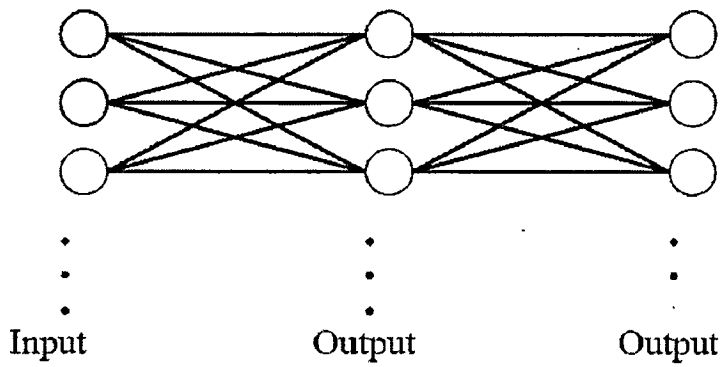


Fig. 2.14 The Structure of Back Propagation Network

The back propagation network is to improve perceptron that lack hidden layer. The hidden layer enhances learning capability of network. The basic axiom is minimum error function. Its characteristic is learning speed slower, but has higher learning accuracy. It suits for diagnosis and prediction and other applications. Tan, S. [2], the content of training sample to stemming and remove stop words. They divided procedures into two parts, one is Principle Component Analysis (PCA) to reduce dimension of feature vector and another is Class Profile-Based Feature (CPBF) to extract keywords and assign weight for each category. They send the results of PCA and CPBF to back propagation network for training. According the result of training to classify web pages.

The advantages of neural network are higher noisy tolerance capability. If input incomplete data, it also through associate to find the most possible output. Neural network

has higher learning capability, according to repeat learning constantly to solve problem. The disadvantage is long time for training and low convergence speed

e. Support Vector Machine (SVM)

The main idea of SVM is on a high dimension space to find a hyper-plane to do binary division achieving the minimum wrong rate [4]. SVM is one of the important capabilities that deal with problem of linear inseparable. SVM is a binary classifier. When test samples input to SVM to classify, SVM just classifies these test samples into "+1" or "-1" category. If SVM will apply to multi-category problem, the simple way is that each category use a SVM to training respectively. Let test samples use different category model of SVM to classification respectively. The category model will classify web pages that belong to this category or not belong to this category.

SVM uses a part of data to do training, to find several support vectors from these training data to represent training data. Let these support vectors to form a model. SVM can accord this model to classify testing data. The classification decision formula is given as:

$$(x_i, y_i),....,(x_n, y_n), x \in R^m, y \in \{+1, -1\}$$

$(x_i, y_i),....,(x_n, y_n)$ are training samples, n is the number of sample, m is the input dimension, y belongs to one of the category +1 or –1 respectively.

In the linear problem, there is a hyper-plane divides into two categories. Fig. 2.15 shows a high dimension space. A hyper-plane is divided into these samples into two categories. The formula of this hyper-plane is:

$$(w \bullet x) + b = 0$$

The classification formula is:

$(w \bullet x) + b > 0 \quad \text{if } y_i = +1$

$(w \bullet x) + b < 0 \quad \text{if } y_i = -1$

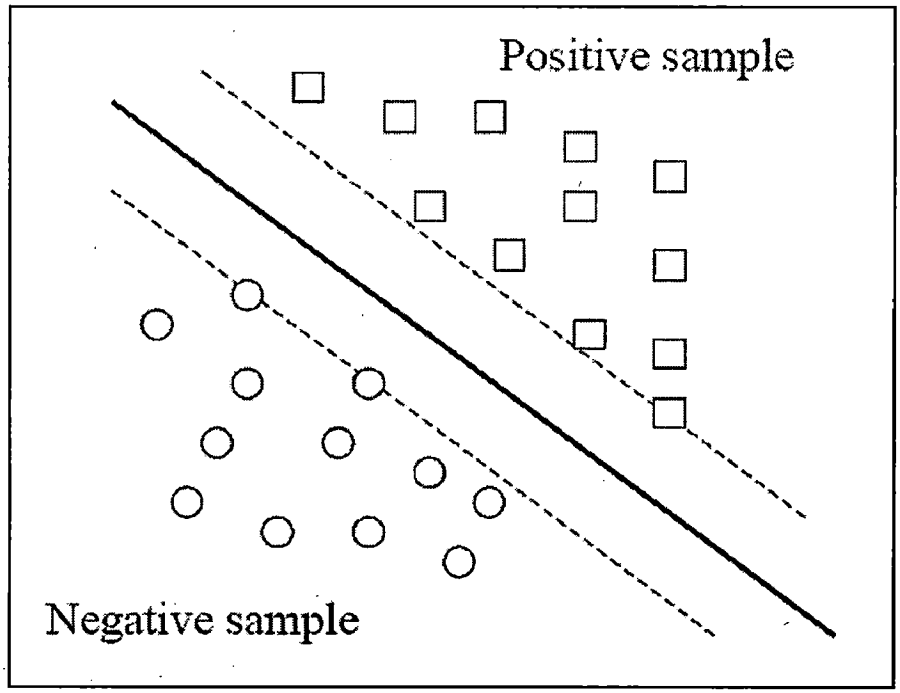


Fig. 2.15 The General Hyper-plane of SVM

SVM except for deal with linear separate problem, another capability is that it can deal with linear inseparable problem. The inner product operation will affect classification function. Therefore, the suit inner product function $K(x_i \bullet x_j)$ can solve certain linear inseparable problem, and it will not increase the complexity. The different kernel functions are suit to different problem types. The kernels that often use are as follows.

- Dot
- Polynomial
- Neural

2.9 Related work

There are many automated classification methods for web pages. A decision tree [1] is a general data classification method. Its two major advantages are (1) it is faster; and, (2) the classification result can be transformed into an IF-THEN relation that the user can easily understand. Common decision tree methods include ID3 and C4.5 [6]. The disadvantage is that when categories are more numerous, it makes mistakes more easily. A support vector machine, named SVM, is a supervised method [4] that uses a portion of the data to train the system and then forms a learning model that can predict the category of documents. k-NN method is often used in text document classification [2] use a k-nearest neighbor (k-NN) approach to calculate the likelihood of a category and relevant

web page. In order to improve performance, they add a feature selection, HTML tags, and a new similarity measure and evaluation. Bayesian classifier [5], transform the frequency of keywords to condition probabilities in which Bayesian probability is used to calculate the probability value between every document and category. Under this system, the category with highest probability is the one the document belongs to. The advantage is that the correlation between two documents can be represented by a probability. However, the processing load is higher.

Framework for Web Document Classification CHAPTER 3

In this dissertation, we propose a frame work for web document classification based on Naïve Bayesian classifier using voting method. In preprocessing phase we remove the html tags and extract the words from html title tag, anchor tag and body tag from each web page at the time of parsing. After the parsing, we use the Porter's stemming algorithm and we also exclude the stop words such as "a", "the", "of" and so forth. After that we use two different feature selection techniques namely LSI and SWT and also constructing Boolean term-document matrix and weighted term document matrix for LSI and SWT respectively. LSI analyze the semantic similarity between term and document by using singular value decomposition (SVD) method to find out the semantic relationship between term and document and also SVD is used to reduce the dimension of Boolean term-document matrix. Whereas SWT analyze the structure of HTML page and assigns the weight according to the structure. These two features are used for training the Naïve Bayesian classifier respectively. Based on the output of the NB classifier, voting method is used to classify the given web page into suitable class. We have used yahoo directories web pages for training and testing the classification method.

The framework of the system is shown on Fig. 1, described as follows:

1. Preprocessing: Preprocessing includes removal of HTML tags and stop words. HTML tags are removed but the text is retained, to prevent interference. We exclude test representing as link in the anchor tags, removal of stopwords and then we apply Porter's stemming algorithm [18] (variants of a word are reduced to a single form). We use two different types of term-document matrices, Boolean term-document matrix and frequency weighted term-document matrix for latent semantic indexing and structured-oriented weighting respectively.
2. Latent semantic indexing (LSI) [19]: After preprocessing, the system constructs a Boolean term-document matrix X. SVD is applied to decomposing the matrix X and the original data vectors are reduced to a small number of features. The latent semantic relationships between keywords and documents are thus obtained.

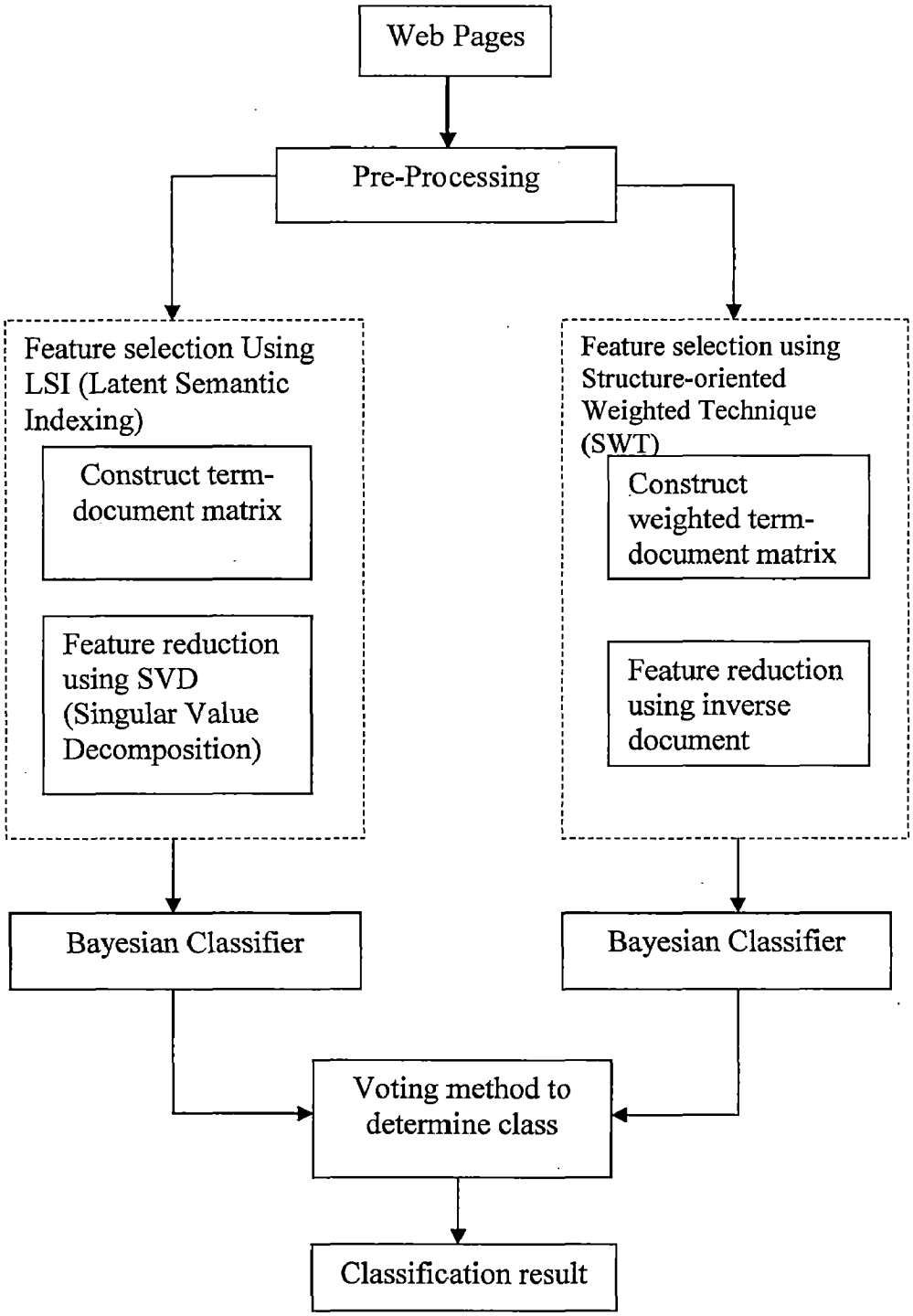


Fig. 3.1 Framework for Web Document Classification

3. Structured-oriented weighting technique (SWT): We also compute feature selection using SWT, the system constructs a weighted term-document matrix *Y*. Term frequency does not exploit the structural information present in web page. The idea is to assign greater importance to terms that belong to the terms that are

more suitable for representing web pages. For example term appearing in the title tag, anchor tag of an html page.

4. Classification: we use semantic features and structure weighted features to train the Naïve Bayesian classifier. The two Naïve Bayesian classifier models are used to predict the category of the web pages.
5. Voting method: After the two Naïve Bayesian classifier models classify the web pages, the two classification results will be used to vote on which category the web page should be placed in. In this approach, we are using voting method to improve the accuracy of the classifier. First we will classify the documents based Naïve Bayesian classifier using LSI features and SWT features separately and then we compare the results of both methods. If both methods determine the document as same category then voting method assign the same category or else the highest probability among both methods is consider to decide the category by voting method.

The detailed description the framework methodologies are discussed in chapter 4.

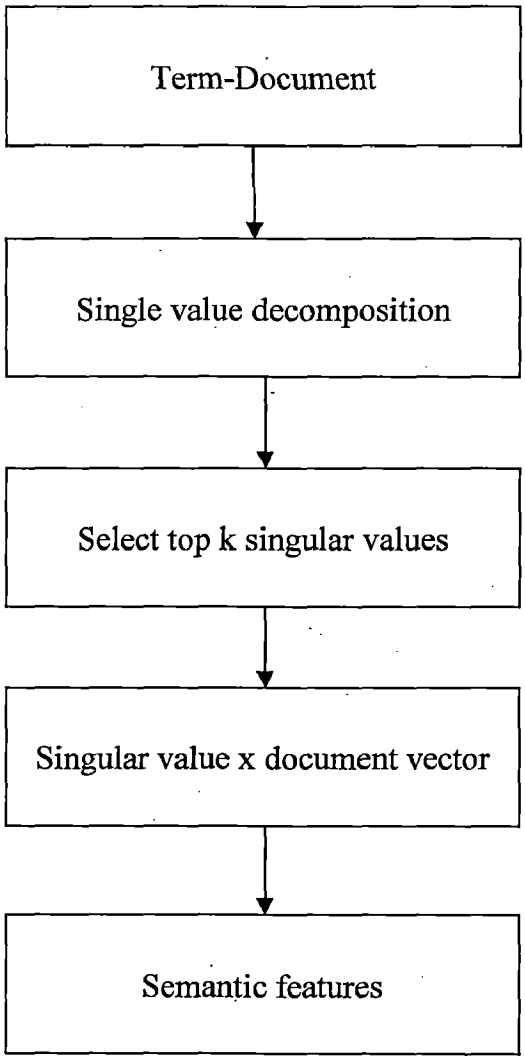


Fig 4.1 Semantic Feature Extraction Procedure

The X of SVD is defined as $X=USV^T$. S=diag(σ_1,....,σ_n), where the elements of S are all singular values of X. Let n=min{t,d}, and the singular value is represented by $\sigma_1 \geq \sigma_2 \geq .... \geq \sigma_n \geq 0$. U and V are d x d, t x t matrices, respectively. After processing by the SVD, $X = USV^T$ simplifies to $X = U_k S_k V_k^T$, as shown in Fig.4.2. The dimensions of U_k, S_k, V_k^T are reduced to d x k, k x k, and k x t. The common element k is less than the original vector space. S_k retains k large singular value in term-document. U_k is a document vector, V_k is a term vector. For the training sample, after the words have been segmented, we construct a term-document matrix for each category. For term-document matrix X_i of each category, we use the SVD to decompose X_i, obtaining three matrixes *U, S, V*. Because we want to find the common semantic relation between different documents, we only process document vector. For the singular value matrix, the top k singular value is selected. The

top k singular value is most important for this data set, as it contains the latent semantic relationship. We add these latent semantic relations into each document vector for the same semantic document. Therefore, we operate (U_k x S_k) to obtain the semantic feature vector of each document.

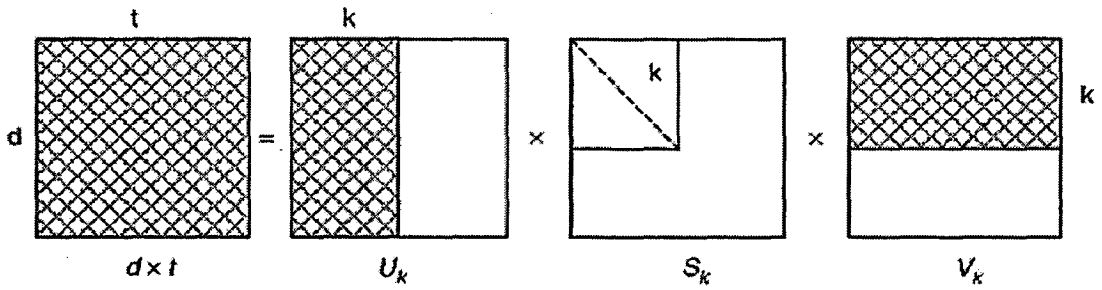


Fig 4.2 SVD Decomposition of term-document Matrix

4.2.2. Structured-oriented Weighting Techniques

Term Frequency:

The baseline method for computing the weight of a term in a document is to count the number of times the term occurs in the document. This method is usually called *Term Frequency (TF),* and is defined by the function

$$TF(t_i, d_j) = \#(t_i, d_j)$$

Where $\#(t_i, d_j)$ denotes the number of times the term t_i occurs in the document d_j

Structure-oriented Weighting Technique (SWT):

Term Frequency does not exploit the structural information present in HTML document. For exploiting HTML structure we must consider not only the number of occurrences of terms in documents but also the HTML element the terms are present in. The idea is to assign greater importance to terms that belong to the elements that are more suitable for representing web pages (the META and TITLE elements)[25].

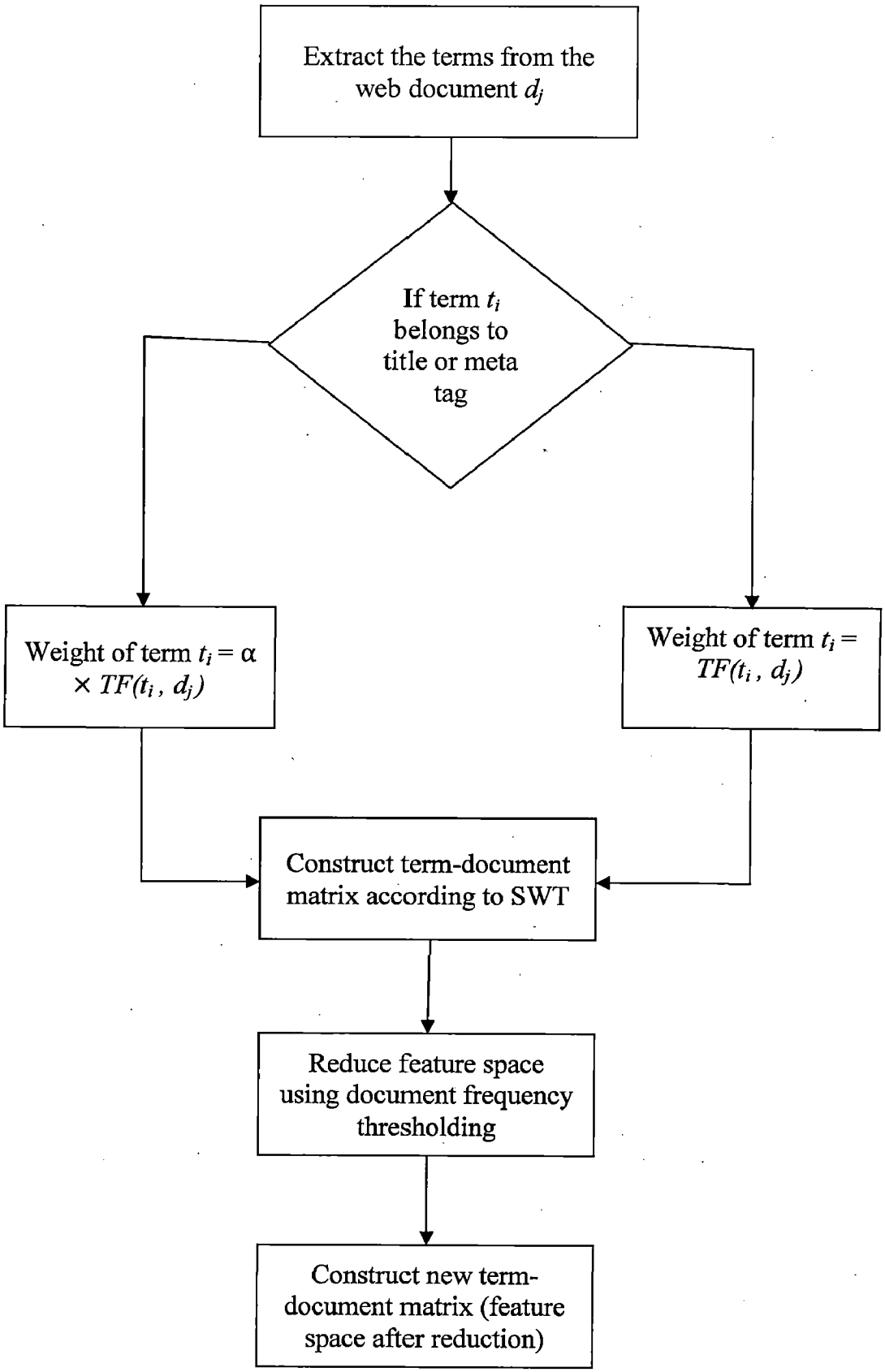


Fig 4.3 Procedure of Structure Oriented Weighting Technique

We call this weighting method *Structure-oriented Weighting Technique (SWT)*. It is defined by the function.

$$SWT(t_i, d_j) = \sum_{e_k} (w(e_k) \times TF(t_i, e_k, d_j))$$

Where e_k is an HTML element, $w(e_k)$ denotes the weight we assign to the element e_k and $TF(t_i, e_k, d_j)$ denotes the number of times the term t_i is present in the element e_k of the HTML document d_j.

Term Frequency is a particular case of *SWT* in which the weight 1 is assigned to every element.

$w(e)$ function is defined as

$w(e) = \propto$ if e = META or TITLE

1 elsewhere

After constructing term-document matrix using SWT method, we apply document frequency thresholding method which is described in section 2.7. The general procedure for applying SWT is shown in Fig. 4.3.

4.3 Naïve Bayesian Classifier

Naïve Bayesian classifier [6][26][27] is a simple probabilistic classifier based on applying Bayes' theorem with strong (Naïve) independence assumptions. A more descriptive term for the underlying probability model would be "independent feature model". Depending on the precise nature of the probability model, Naïve Bayesian classifiers can be trained very efficiently in a supervised learning setting. An advantage of the Naïve Bayesian classifier is that it requires a small amount of training data to estimate the parameters necessary for classification.

The Bayes theorem is given by:

$$p(A \mid B) = \frac{p(A)\, p(B \mid C)}{p(B)}$$

This is an interesting theory that states that we can calculate the probability of A given B based on the reverse relation. The Bayes theorem, stated in terms of this task, would be the following.

Methodology CHAPTER 4

4.1 Preprocessing

Preprocessing includes removal of HTML tags and stop words. And then Porter's stemming algorithm is applied. By exploiting HTML structure [20] for web page representation we can choose how a term is representative of the page considering the HTML element present in it. For example, we can represent a web page using only the words of the title, that is to say the words extracted from the TITLE element or Body element. For obtaining good performance in web page representation exploiting HTML structure is important to know where the more representative words can be found. For example, we can think that a word present in the TITLE element is generally more representative of the document's content than a word present in the BODY element.

Five different text sources for web page representation,

- BODY, the content of the BODY tag.
- META, the meta-description of the META tag also known as anchor text.
- TITLE, the page's title.
- MT, the union of META and TITLE content.

4.1.1. HTML Document Structure

HTML documents are structured into two parts, the HEAD, and the BODY. Both of these are contained within the HTML element, this element simply denotes this as an HTML document.

Example of Document Structure

```
<HTML>
                                    <HEAD>
  <TITLE> Web Document Classification </TITLE>
 </HEAD>
 <BODY>
  <h1> Naïve Bayesian Classifier </h1>
  Welcome to the home page of the web document classification.
  This page shows the framework for web document classification based
  on Naïve Bayesian classifier using voting method.
  <p>So how will we do this.  Well we do the following
<ul>
  <li>Feature Selection using <A HREF="lsi.html">LSI Features</A>.
  </ul>
 </BODY>
 </HTML>
```

4.1.2. Contents Extraction from Webpage and removal of HTML tags

The target documents of our application are WebPages. So we need to extract only contents from a webpage because there are much more text which is not related to the contents, for example, HTML tag java script. As we know, webpage is made of HTML and HTML is not well organized language. So, it's not so easy work to extract contents because there are various exceptions and various expressions to represent same thing in HTML.

There are two kinds of texts which should be eliminated in texts formatted by HTML. One is meta text including HTML tag, and the other is contents text which is not related to real contents. First, we eliminate HTML tag marked by "<" and ">" characters. Then, we eliminate java script and style sheet, but there are some ambiguity to decide entry point and end point of them because their entry point is expressed variously, furthermore they has various special characters include "<" and ">". And we eliminate not only these meta texts but also contents texts which are not related to real contents. For example, the texts between anchor tags represents link to other webpage. The other

webpage which is linked is meaningful of course, but the texts represent link itself is not meaningful at all in many cases. So we exclude the text represent link.

4.1.3 Removing stopwords and stemming

In this section, we will discuss two text preprocessing operations: (1) elimination of stopwords and (2) word stemming. Elimination of stopwords with the objective of filtering out words with very low discrimination values for the document classification purpose. Stemming of the words with the objective of removing affixes (i.e., prefixes and suffixes) and allowing the documents containing syntactic variations of words (e.g., connect, connecting, connected etc).

(1) Elimination of Stopwords

In a document, not all words are equally significant for representing the semantics of the document. Those words are too frequent among the documents in the collection are not good discriminators. Such words are frequently referred to as stopwords and are normally filtered out as potential feature words. Articles, prepositions and conjunctions are natural candidates for a list of stopwords. Stopword elimination can provide for compression of the document text. Therefore, some verbs, adverbs, and adjectives also could be treated stopwords. The set of stopwords employed in our document clustering algorithm is listed in **Appendix A**.

(2) Stemming Technique

Stemming techniques are used to improve the efficiency of the information system and to improve recall. Conflation is the term frequently used to refer to mapping multiple morphological variants to a single representation (stem). The idea of equating multiple representations of a word as a single stem term would appear to provide text compression, with associated savings in storage and processing. For example, the stem "comput" could associate "computable, computability, computation, computational, computed, computing, computer" to one compressed word.

The most common stemming algorithm removes suffixes and prefixes, sometimes recursively, to derive the final stem. Other techniques such as table lookup and successor stemming provide alternatives that require additional overheads. Successor stemmers determine prefix overlap as the length of a stem is increased. This information can be

used to determine the optimal length for each stem from a statistical versus a linguistic perspective. Table lookup requires a large data structure. While there are several well known suffix removal algorithm, the Porter's algorithm is the most commonly accepted algorithm [18]. In this dissertation, we will apply this stemming algorithm in our web document classification problem.

Algorithm:

The Porter Algorithm is based upon a set of conditions of the stem, suffix and prefix and associated actions given the condition. Some examples of stem conditions are:

1. The measure, *m*, of a stem is a function of sequences of vowels *(a, e, i, o, u, y)* followed by a consonant. If *V* is a sequence of vowels and *C* is a sequence of consonants, then m is:

$$C(CV)mV$$

where the initial *C* and final *V* are optional and *m* is the number *VC* repeats

Measure	Example
$m = 0$	free, why
$m = 1$	frees, whose
$m = 2$	prologue, computer

2. *<*X*> - stem ends with letter *X*
3. **v** - stem contains a vowel
4. **d* - stem ends in double consonant
5. **o* - stem ends with consonant-vowel-consonant sequence where the final consonant is not *w, x or y*

Suffix conditions take the form current suffix = pattern

Actions are in the form old suffix ⟶ new suffix

Table 4.1 Porter's Stemming Algorithm Rules

RULE	CONDITION	SUFFIX	REPLACEMENT	EXAMPLE
1a	NULL	ssess	ss	stresses ⟶ stress
1b	*v*	ing	NULL	making ⟶ mak
1b1	NULL	at	ate	inflat(ed) ⟶ inflate
1c	*v*	y	i	happy ⟶ happi
2	$m > 0$	aliti	al	formaliti ⟶ formal
3	$m > 0$	icate	ic	duplicate ⟶ duplic
4	$m > 1$	able	NULL	Adjustable ⟶ adjust
5a	$m > 1$	e	NULL	Inflate ⟶ inflat
5b	$m > 1$ and *d and *<L>	NULL	single letter	controll ⟶ control

Rules are divided into steps to define the order of applying the rules. The following are some examples of the rules:

Given the word "duplicatable", the following are the steps in the stemming process:

Step 1:	duplicatable ⟶ duplicat	(by rule 4)
Step 2:	duplicat ⟶ duplicate	(by rule 1b1)
Step 3:	duplicate ⟶ duplic	(by rule 2)

Note that only one rule from each step is allowed to be applied

4.1.4 Document Vector

After the eliminating the irrelevant terms extracted from document, each document is represented by a sparse vector of weights $d_j = (w_1, w_2, ..., w_r)$ where w is the number of terms that occur at least once in the collection of documents and r is an index of terms and j is an index of documents. This sparse vector is known as Document Vector or term-document matrix [21]. This document vector is very sparse vector because index r should be counted by whole training document set and one document has only few terms of them of course. An index r should be global unique in whole training document set of the category because each of them will be used by one feature dimension in training step of Machine Learning. So we made an index table which is implemented by hash

table to assign an index r to each term. And, to get an index r, we used an index table which is made in previous step. An index table for category of each document is also made by same way.

As mentioned in section x , We use two different types of term-document matrices, Boolean term-document matrix and frequency weighted term-document matrix for latent semantic indexing and structured-oriented weighting respectively.

1. Boolean term-document matrix: If the word w_i occurs in the document d_j then at position (i , j) in the matrix is 1, else 0

$$(w_i, d_j) = 1 \text{ ,if } w_i \text{ occurs in } d_j$$
$$= 0 \text{ ,else}$$

2. Frequency weighted term-document matrix: If the word w_i occurs n times in the d_j then at position (i , j) in the matrix is n , else 0.

$$(w_i, d_j) = n \text{ ,if } w_i \text{ occurs } n \text{ times in } d_j$$
$$= 0 \text{ ,else}$$

4.2. Feature Selection

The main focus of Feature selection [22] method is used for dimensionality reduction and construction of new features. The overall feature selection procedure is to score each potential feature according to a particular feature selection metric, and then take the best k features.

Many researchers have addressed the problem of feature selection [23]. In the naïve Bayesian method, extremely long feature vectors may result in an extremely high cost for the computation of the values of $p(Ci|X)$ and $p(X)$ where Ci represents category i and X represents the given document. On the other hand, feature vectors that are too short are unable to distinguish among the documents. For a given document, the feature vector representation gives the frequency with which each word occurred in that document. Our feature vector has more than a thousand features. Some of the features are quite useful in distinguishing among the documents, but others are not. The goal of feature selection is to remove those features that are not informative, thus reducing the length of the feature

vector. In our experiments, we used two different feature selection methods: latent semantic indexing and inverse document frequency.

4.2.1. Latent Semantic Indexing

After preprocessing, the system extracts the web page text features (only extracting Body elements) and then constructs a Boolean term-document matrix X. SVD (Singular Value Decomposition) [24] is applied to decomposing the matrix X and the original data vectors are reduced to a small number of features. The latent semantic relationships between keywords and documents are thus obtained. The procedure is shown in Fig. 4.1.

SVD is a reliable tool for matrix factorization. For any matrix X, $X.X^T$ has a nonnegative eigen values. The nonnegative square roots of the eigen values of $X.X^T$ are called the singular values of X, and the number of non-zero singular values is equal to the rank of X, *rank*(X). An SVD can reduce the original high term-document matrix dimensions to a low term-document matrix. Assume a Boolean term-document matrix X, which is a t * d matrix, where t is the number of keywords and d the number of documents in Table 4.2. Each element X [t,d] is the occurrences of keyword t in document d. For example, if the position of X [1,1] is 1, T1 occurs in document D1 and if it is 0 then term does not occur in document.

Table 4.2 Term-Document Matrix

Terms	Documents						
	D1	*D2*	*D3*	*D4*	....	....	*Dn*
T1	*1*	*0*	*0*	*1*	.....	.....	*0*
T2	*0*	*1*	*1*	*0*	....	....	*1*
....	.....	....	....	....	....	....	...
Tn	*1*	*1*	*0*	*1*	....	....	*0*

$$p(Class \mid Document) = \frac{p(Class)\, p(Document \mid Class)}{p(Document)}$$

The above equation can be written as

$$\text{posterior} = \frac{\text{prior} \times \text{likelihood}}{\text{evidence}}$$

Using Bayes' theorem,

$$p(C \mid F_1,...F_n) = \frac{p(C)\, p(F_1,....,F_n \mid C)}{p(F_1,....,F_n)}$$

In practice we are only interested in the numerator of that fraction, since the denominator does not depend on C and the values of the features F_i are given, so that the denominator is effectively constant. The numerator is equivalent to the joint probability model

$p(C \mid F_1,...F_n)$

this can be rewritten as follows, using repeated applications of the definition of conditional probability:

$$p(C \mid F_1,...F_n)$$

$$= p(C)\, p(F_1,...F_n \mid C)$$

$$= p(C)\, p(F_1 \mid C)\, p(F_2,...F_n \mid C, F_1)$$

$$= p(C)\, p(F_1 \mid C)\, p(F_2|C,F_1)\, p(F_3,...F_n \mid C, F_1, F_2)$$

$$= p(C)\, p(F_1 \mid C)\, p(F_2|C,F_1)\, p(F_3|C,F_1,F_2)\, p(F_3,...F_n \mid C, F_1, F_2, F_3)$$

and so forth. Now the "naive" conditional independence assumptions come into play: assume that each feature F_i is conditionally independent of every other feature F_j for j is not equal to i. This means that

$$p(F_i \mid C, F_j) = p(F_i \mid C)$$

and so the joint model can be expressed as

$$p(C \mid F_1,...F_n) = p(C)\, p(F_1 \mid C)\, p(F_2 \mid C).....$$

$$= p(C) \prod_{i=1}^{n} p(F_i \mid C)$$

and this can be rewritten as :

$$p(Class \mid Document) = p(Class)\prod_i p(Word_i \mid Class)$$

Normalization of probabilities:

The result of applying Bayesian learning is a set of (very) small probabilities. These probabilities and how they are related to each other can be hard to analyze. However, by normalize the probabilities, i.e. the sum of all probabilities equal one, they becomes more readable. The normalization are described by the following algebraic formula:

$$\frac{p_1}{p_1 + ... + p_n} + ... + \frac{p_n}{p_1 + ... + p_n} = 1$$

$$e.g.\ p(C_1) = \frac{p_1}{p_1 + ... + p_n}$$

Moreover, we will only consider a specific set of words, called the vocabulary, a distinct set of words. The size of the vocabulary must be carefully chosen; too many words will introduce difficulties due to space limitations, but if the words are too few, the classifier will work poorly. The words not represented in the vocabulary will be disregarded in the classification.

NB learning algorithm

Let D be a document represented as a set of finite terms $D = \{w_1, w_2, \ldots, w_n\}$.

Let doc_{si} be the number of documents in category C_i , and |Examples| be the number of documents in the training set of labeled documents.

Step 1: collect the word list, which is defined as the set of distinct words in the whole training set

Step2: For each category Ci do the following

Compute $p(C_j) = | doc_{sj} |/|Examples|$ (1)

where doc_{sj} is the number of training documents for the category is C_j.

For each term w_k in word list

Compute $p(w_k/C_j)= (N_{kj} +1)/(n_j +| Text_j |)$ (2)

where N_{kj} is the number of times w_k occurs in C_j, n_j is the total number of distinct terms in all training documents labeled C_j, and $Text_j$ is a single documents generated by concatenating all the training documents for category C_j .

Equation (1) and (2) make use of the following two assumptions:

1) Assuming that the order of the words in a document does not affect the classification of the document:

$p(D|C_j)=p(\{w_1, w_2,..., w_n\}|C_j)$

2) Assuming that the occurrence of each word is independent of the occurrence of other words in the document then:

$p(w_1,...,w_n|C_j)=p(w_1|C_j)*p(w_2|C_j)*...*p(w_n|C_j)$

4.4 Voting method

In this approach, we are using voting method to improve the accuracy of the classifier. First we will classify the documents based Naïve Bayesian classifier using LSI features and SWT features separately and then we compare the results of both methods. If both methods determine the document as same category then voting method assign the same category or else the highest probability among both methods is consider to decide the category by voting method.

```
Algorithm:
//get the class determined by LSI and probability
//here LSI features as training set
LSI_result = LSI_Bayesian (training_set,test_doc)

//get the class determined by SWT and probability
//here SWT features as training set
SWT_result = SWT_Bayesian (training_set,test_doc)

//Compare the results of both LSI and SWT
IF (LSI_result.clas=SWT_result.clas)
THEN classified_clas= LSI_result.clas (or SWT_result.clas)

ElSE IF (LSI_result.prob > SWT_result.prob)
THEN classified_clas=LSI_result.clas

ELSE
Classifier_clas = SWT_result.class
```

Results and Discussions CHAPTER 5

5.1 Experiment Environment

In our experiment, each category uses two Naïve Bayesian models to do the training and testing. Our experiment uses Pentium-D 2.66 GHz computer with 2 GB RAM, Java language was implemented on a Windows XP operating system. One motivation for using Java as implementation languages is that it's platform independent, i.e. that that a compiled program can be run on many different architectures. This is accomplished by compiling to an intermediate format, called Java bytecode that is later run with a JVM, Java Virtual Machine, and an interpreter. Another motivation is that it's a strict, imperative language, which is well suited for a task with a large amount of data that have to be accessed rapidly and also it is the well developed API (library), which simplifies and speeds up the implementation effort.

The Java implementation consists of programs as follows: preprocessing, feature selection using LSI, feature selection using SWT, Naïve Bayesian trainer and test runner.

5.2 Data Set

The data set of web pages is downloaded from Yahoo! Directories [7]. In our classification model we used first level of Yahoo! Directories are as follows:

1. Arts & Humanities
2. Business & Economy
3. Computers & Internet
4. Education
5. Entertainment
6. Sports

The whole date is departed into 6 classes with 4414 web pages, include 621 web pages of Arts & Humanities class, 710 web pages of Business & Economy class, 794 web pages of Computers & Internet class, 742 web pages of Education class, 642 web pages of Entertainment class and 635 web pages of Sports class. We used holdout method (2N/3; N/3) : Train the classification model on 2/3rd of data available. Test on remaining 1/3rd. For the training set, we randomly selected a part of data from each category, leaving

the remainder for the test set. The ratio of training set to test set is 2:1 approximately. The test data set is of 207 web pages of Arts & Humanities class, 236 web pages of Business & Economy class, 264 web pages of Computers & Internet class, 247 web pages of Education class, 214 web pages of Entertainment class and 211 web pages of Sports class.

5.3 Performance Evaluation

In our web page classification, we used Precision, Recall and F-value measure for performance test. Precision is the number of correct categories assigned divided by the total number of categories assigned, and serves as a measure of classification accuracy, the higher the precision, the smaller the amount of false categories. Recall is the number of correct categories assigned divided by the total number of known correct categories, the higher recall means a smaller amount of missed categories.

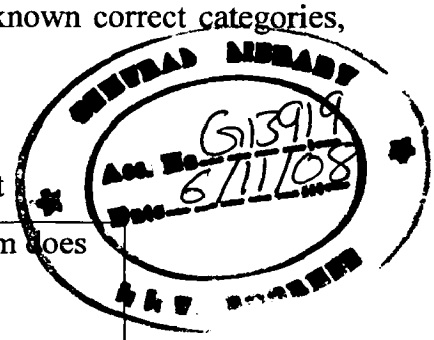


Table 5.1 Situations of Classifier Result

	The system classified category X	The system does not classify category X
Belongs to category X	A	B
Not belongs to category X	C	D

A, The number of pages classified to Category X and belonging to Category X;

B, The number of pages not classified to Category X, but belonging to Category X;

C, The number of pages classified to Category X but not belonging to Category X;

D, The number of pages not classified to Category X and not belonging to Category X.

The formula of precision, recall and F-value is listed as follows.

Precision (P)=A/(A+B).

Recall(R)=A/(A+C).

F-value=2PR/(P+R)

In this experiment, we compare the three different methods for classifying a web page: Voting method (proposed system), LSI-NB (Naïve Bayesian classifier using LSI

features) and SWT-NB (Naïve Bayesian classifier using SWT features. An LSA-NB and SWT-NB are given the same data set, respectively, to compare them to a Voting method. The LSA-NB sends the features extracted from the LSA operation to the Naïve Bayesian classifier to train and classify. In the same manner, SWT-NB sends the features extracted from SWT to the Naïve Bayesian classifier to train and classify. And then we compare the both methods results using voting method.

In order to obtain the highest classification performance, we tested LSI-NB model by varying the threshold parameter k values such as 500, 700, 900, 1200 and 1500 and compared the classification performance on these threshold parameters. The best effective threshold parameter was then selected. And then we tested SWT-NB model to find effective threshold parameter α (at 1, 2, 4, 6, 8 and 10).

First, we apply the preprocessing methods includes extraction of terms from the documents, removal of html tags and irrelevant content, removal of stopwords and stemming process. And we prepare the vocabulary word list on the relevant terms extracted and we construct term document matrix.

Second, we tested the LSI-NB (Naïve Bayesian classifier on LSI features) method using above test data on varying the features selection threshold k, values such as 500, 700, 900, 1200 and 1500 and we obtained different average F-value's measure on 6 classes which is shown in Table 5.2.

Table 5.2 Average F-value's of LSI-NB

Number of terms (k)	Average F-value
500	0.8262
700	0.8304
900	0.8396
1200	0.8329
1500	0.8256

As shown in the Table 5.2, the F-value at k=900 obtains the best performance with 900 terms. Table 5.3 shows the precision, recall and F-value of LSI-NB method on 6

classes (1.Arts & Humanities, 2.Business & Economy, 3.Computers & Internet, 4.Education, 5.Entertainment, 6.Sports) at k=900.

Table 5.3 Performance of LSI-NB method at K=900

Category Number	Category Name	Precision	Recall	F-value
1	Arts & Humanities	0.8357	0.9057	0.8692
2	Business & Economy	0.8093	0.8232	0.8161
3	Computers & Internet	0.8484	0.7887	0.8174
4	Education	0.8502	0.8433	0.8467
5	Entertainment	0.8271	0.8428	0.8348
6	Sports	0.8578	0.8497	0.8537

Third, we tested the SWT-NB (Naïve Bayesian classifier on SWT features) method using the above test data set on varying the structural weight α (assigning the more weight depending on the structure of a HTML document), such as 1,2,4,6,8. We used document frequency (DF) thresholding as feature reduction method. And we obtained average F-value's on varying weight α as shown in Table 5.4. At α =1 works same as term frequency method.

Table 5.4 Average F-value's of SWT-NB

Weight	Average F-value
α = 1 (TF)	0.8064
α = 2	0.8106
α = 4	0.8184
α = 6	0.8275
α = 8	0.8214
α = 10	0.8186

Experimental results in Table 5.4 show that Structure-oriented Weighting Technique (SWT) can improve classification accuracy, assigning to META and TITLE elements a greater weight than to the other elements. As shown in the Table 5.4, the F-value at α = 6 and document frequency thresholding as feature reduction method, obtains

the best performance of SWT-NB method. Table 5.5 shows the precision, recall and F-value of SWT-NB method on 6 classes (1.Arts & Humanities, 2.Business & Economy, 3.Computers & Internet, 4.Education, 5.Entertainment, 6.Sports) at $\alpha = 6$.

Table 5.5 Performance of SWT-NB at $\alpha = 6$

Category Number	Category Name	Precision	Recall	F-value
1	Arts & Humanities	0.8115	0.8527	0.8315
2	Business & Economy	0.8262	0.8024	0.8141
3	Computers & Internet	0.8030	0.7940	0.7984
4	Education	0.8016	0.8215	0.8114
5	Entertainment	0.8504	0.8584	0.8543
6	Sports	0.8720	0.8401	0.8557

Finally, we tested the proposed model called voting method- naïve Bayesian classifier (voting-NB) by combining the two method LSI-NB and SWT-NB using voting scheme. The precision of Voting method, LSI-NB and SWT-NB is shown in Fig 5.1 and in Table 5.6, the recall of Voting method, LSI-NB and SWT-NB is shown in Fig 5.2 and in Table 5.7, F-value measure of Voting method, LSI-NB and SWT-NB is shown in Fig 5.3 and in Table 5.8, on 6 classes (1.Arts & Humanities, 2.Business & Economy, 3.Computers & Internet, 4.Education, 5.Entertainment, 6.Sports).

Table 5.6 Precision of LSI-NB, SWT-NB and Voting-NB

Category Number	Category Name	LSI-NB	SWT-NB	Voting-NB
1	Arts & Humanities	0.8357	0.8115	0.913
2	Business & Economy	0.8093	0.8262	0.8855
3	Computers & Internet	0.8484	0.803	0.8371
4	Education	0.8502	0.8016	0.8825
5	Entertainment	0.8271	0.8504	0.8831
6	Sports	0.8578	0.872	0.872

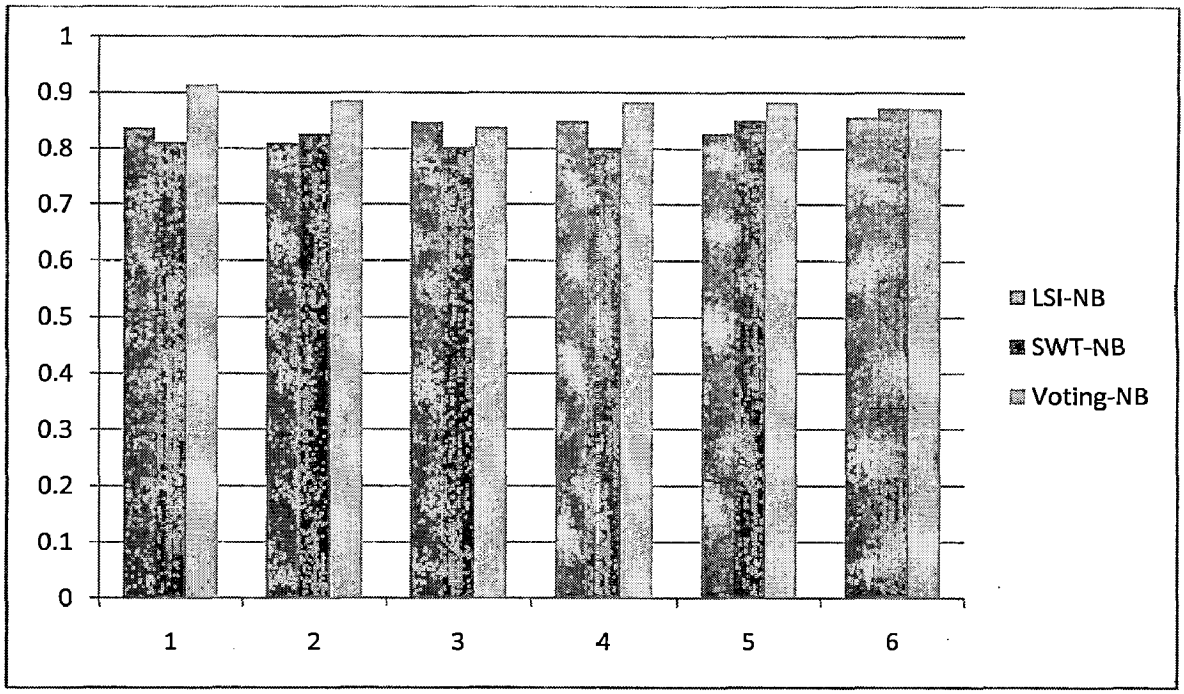


Fig.5.1 Precision Comparison of LSI-NB, SWT-NB, Voting-NB

The above Fig. 5.1 shows the precision comparison of LSI-NB, SWT-NB, Voting-NB. With the exception of computer & internet class, the Voting-NB method yields better precision than LSI-NB and SWT-NB.

Table 5.7 Recall of LSI-NB, SWT-NB and Voting-NB

Category Number	Category Name	LSI-NB	SWT-NB	Voting-NB
1	Arts & Humanities	0.9057	0.8527	0.931
2	Business & Economy	0.8232	0.8024	0.8495
3	Computers & Internet	0.7887	0.794	0.87
4	Education	0.8433	0.8215	0.879
5	Entertainment	0.8428	0.8584	0.8709
6	Sports	0.8497	0.8401	0.8846

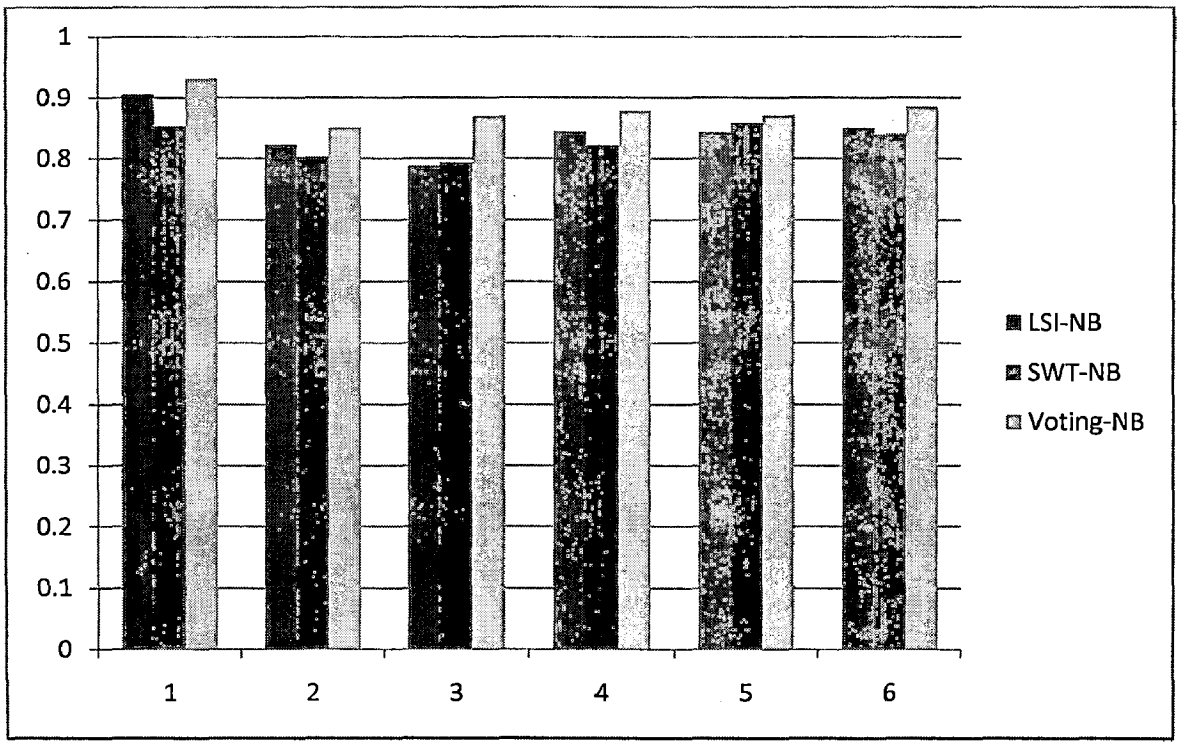


Fig 5.2 Recall Comparison of LSI-NB, SWT-NB, Voting-NB

The Voting-NB method has better recall value in all 6 categories compare to LSI-NB and SWT-NB as shown in Fig. 5.2.

Table 5.8 F-value measure of LSI-NB, SWT-NB and Voting-NB

Category Number	Category Name	LSI-NB	SWT-NB	Voting-NB
1	Arts & Humanities	0.8692	0.8315	0.9219
2	Business & Economy	0.8161	0.8141	0.8671
3	Computers & Internet	0.8174	0.7984	0.8532
4	Education	0.8467	0.8114	0.8807
5	Entertainment	0.8348	0.8543	0.8769
6	Sports	0.8537	0.8557	0.8782

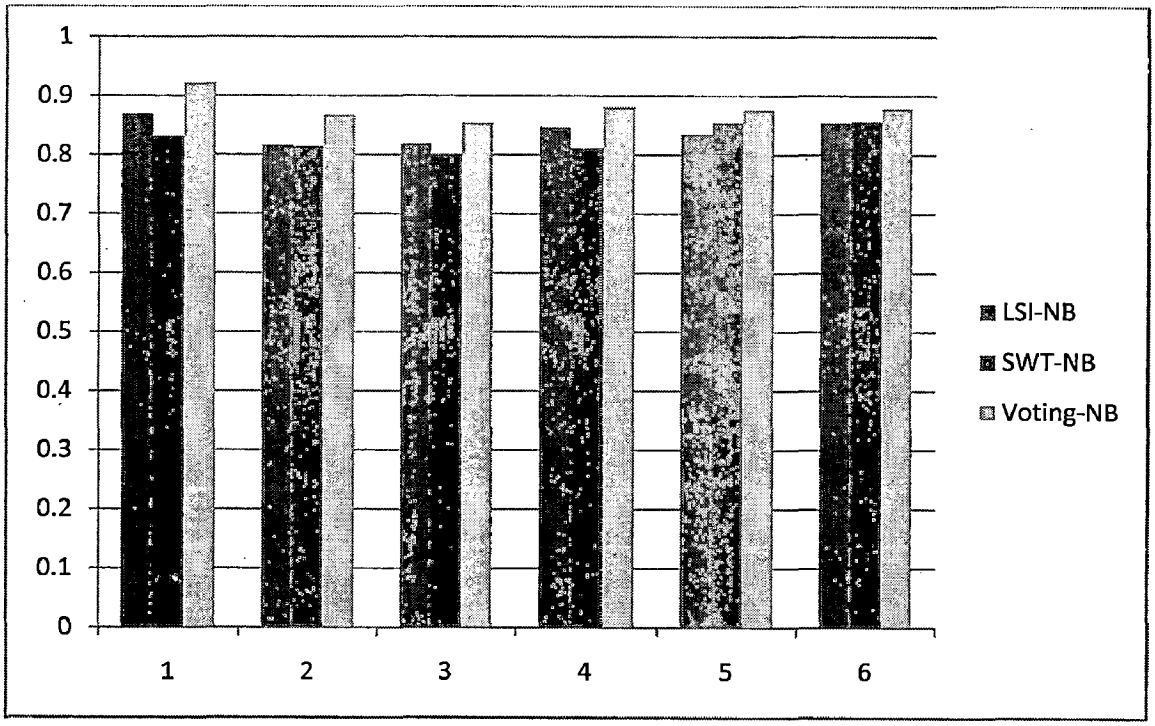


Fig 5.3 F-value measure Comparison of LSI-NB, SWT-NB, Voting-NB

By combination of both semantic relationship between term and document features and weighted structured feature in a Voting-NB method yields better F-value than considering LSI-NB method and SWT-NB separately.

Table 5.9 Performance of LSI-NB, SWT-NB, Voting-NB

	Average Precision	Average Recall	Average F-value
LSI-NB	0.8380	0.8422	0.8396
SWT-NB	0.8274	0.8281	0.8275
Voting-NB	0.8789	0.8808	0.8796

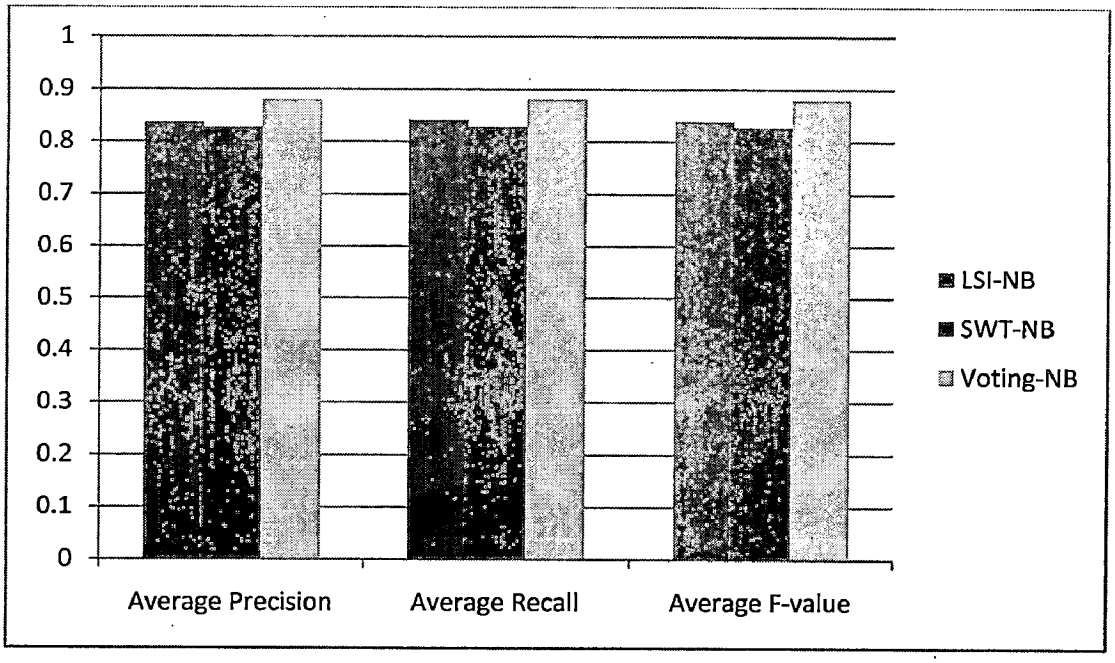


Fig 5.4 Performance Comparison of LSI-NB, SWT-NB, Voting-NB

The average precision for Voting-NB, LSI-NB and SWT-NB are 87%, 82%, and 83%, respectively as shown in Table 5.9. With the exception of computers & internet category, the precision of each category for Voting-NB method is higher than that of the LSI-NB and SWT-NB. This indicates that the two Naïve Bayesian classifier using voting methods yield better precision than the using separately.

The average recall for Voting-NB, LSI-NB and SWT are 88%, 84% and 83%, respectively as shown in the Table 5.9. The recall of each category for Voting method is higher than that of the LSI-NB and SWT-NB.

The average F-value measure for Voting-NB, LSI-NB and SWT-NB are 88%, 84% and 83%, respectively as shown in Table 5.9. The average F-value of Voting-NB method outperforms when combining LSI and SWT features. It shows the 4% improvement over considering LSI-NB and SWT-NB separately.

Conclusion and Scope for Future Work CHAPTER 6

6.1 Conclusion

In this dissertation, we proposed a framework for web page classification method based on Naïve Bayesian classifier using voting method. The proposed model describes the classifier learning on two different features, semantic and structure feature vectors. By using the voting method, the advantages of LSI features and SWT features in obtaining feature vectors. The LSI can extract common semantic relations between terms and documents and thus classifies semantically related web pages, offering more complete information. The SWT extracts four different html structured text features from the web page content. We compared the Voting -NB method performance with LSI-NB, SWT-NB. The experimental results show that the Voting method-NB yields the best result than LSI-NB and SWT-NB.

In our experiment we used data set of web pages downloaded from Yahoo! Directories. In our classification model we used first level of Yahoo! Directories are as follows: Arts & Humanities, Business & Economy, Computers & Internet, Education, Entertainment, Sports. The whole data is departed into 6 classes with 4414 web pages. For the training set, we randomly selected a part of data from each category, leaving the remainder for the test set. The ratio of training set to test set is 2:1 approximately.

The following conclusions can be made from the results obtained using the proposed system and above mentioned data:

- The system can yield good performance if sufficient training data is obtained, and significant amount of supporting data is used for prediction.
- The classification module resulted in good precision, recall and F-value of Voting-NB method around 87%, 88% and 88% respectively.
- The results show that performance improves as the number of relevant documents increases. This implies that our system classifies is accurate and with good recall.

6.2 Scope for Future Work

There is a significant room for improving the methods used in this dissertation for the web document classification. The possible improvements in the future are listed as below:

- Feature selection in our system is done using LSI and SWT with document frequency thresholding, but many more techniques can also be explored for this purpose. Feature selection is itself an area of research.
- For Classification purpose we used Naïve Bayesian classifier, other techniques can also be explored in this area.
- Our system classifies results into single label-flat classification model; it can be modified to multi label-hierarchical classification model.

References

1 Apte, C., Damerau, F., and Weiss, S. M., "Text Mining with Decision Trees and Decision Rule", *In Workshop on Learning from Text and Web, Conference on Automated Learning and Discovery,* pp. 1-4, 1998.

2 Tan, S., "Neighbor-Weighted k-Nearest Neighbor for Unbalanced Text corpus", *Expert Systems with Applications*, vol. 28, issue 4, pp. 667-671, 2005.

3 Selamat, A. and Omatu, S., "Web Page Feature Selection and Classification using Neural Networks", *Information Sciences- Informatics and Computer Science Intelligent System Applications*, vol. 158, issue 1, pp. 69–88, 2004.

4 Joachims, T., "Text Categorization with Support Vector Machines: Learning with Many Relevant Features", *In Proceedings of ECML-98, 10th European Conference on Machine Learning,* vol. 1398, pp. 137--142, 1998.

5 Mccallum, A., and Nigam, K., "A Comparison of Event Models for Naïve Bayes Text Classification", *In Proceedings of AAAI-98 Workshop on Learning for Text Categorization,* pp. 41-48, 1998.

6 Mitchell , T. M., "Machine Learning", *Boston, MA: McGraw-Hill,* 1997.

7 Yahoo! Directory, available at http://dir.yahoo.com.

8 Etzioni. O., "The World Wide Web: Quagmire or Gold Mine", *In Proceedings of Communications of the ACM,* vol. 39, issue 11, pp. 65-68, 1996.

9 Kosala, R., and Blockeel, H., "Web Mining Research: A Survey", *In Proceedings of ACM SIGKDD Exploration Newsletter*, vol. 2, issue 1, pp. 1-15, July 2000.

10 Chakrabarti, S., "Data Mining for Hypertext: A Tutorial Survey", *In Proceedings of ACM SIGKDD Exploration Newsletter*, vol. 1, issue 2, pp. 1-11, 2000.

11 Madria, S. K., Bhowmick, S. S., Ng, W. K., and Lim, E. P., "Research Issues in Web Data Mining", *In Proceedings of Data Warehousing and Knowledge Discovery, First International Conference, DaWaK'99,* vol. 1676, pp. 301-312, 1999.

12 Cooley, R., Mobasher, B., and Srivastava, J., "Data Preparation for Mining World Wide Web Browsing Patterns", *In Journal of Knowledge and*

Information Systems, vol. 1, issue 1, pp. 5-32, 1999.

13 Antonellis, I. and Gallopoulos, E., "Exploring term-document Matrices from Matrix Models in Text Mining", *In Proceedings of SIAM Text Mining Workshop, Technical Report,* pp. 1-11, 2006.

14 Lewis, D. D. and Ringuette, M., "Comparison of two learning algorithms for text categorization", *In Proceeding of the Third Annual Symposium on Document Analysis and Information Retrieval (SDAIR-94),* pp. 81-93, 1994.

15 Wiener, E., Pederson, J.O. and Weigend, A., "A Neural Network Approach to Topic Spotting", *In Proceeding of the Fourth Annaul Symposium on Document Ananlysis and Information Retrieval (SDAIR-95)*, pp. 317-332, 1995.

16 Maderlechner, G., Suda, P. and Brückner, T., "Classification of Documents by Form and Content", *In Pattern Recognition Letters*, vol. 18, issue 11, pp. 1225-1231, 1997.

17 Asirvatham, A. P. and Ravi K. K., "Web Page Classification based on Document Structure", *In National Level Paper Contest Conducted by IEEE India Council, Technical Report* , pp. 1-10, 2001.

18 Porter, M.F., "An Algorithm for Suffix Stripping", *Published in Program*, vol. 14, issue 3, pp 130–137, 1980.

19 Deerwester, S., Dumais, S., Furnas, G., Landauer, T. and Harshman, T., "Indexing by Latent Semantic Analysis", *Journal of the American Society for Information Science*, vol. 41, issue 6, pp. 391–407, 1990.

20 Robert, P., Futrelle, Andrea, G. and Mingyan, S., "Extracting Structure from HTML Documents for Language Visualization and Analysis", *In Proceeding of Workshop on Web Document Analysis (WDA-2003)*, pp. 3-6, 2003.

21 Salton, G. and Buckley, C., "Term Weighting Approaches in Automatic Text Retrieval", *In Proceeding of Information Processing and Management*, vol. 24, issue 5, pp. 513-523, 1998.

22 Yang, Y. and Pedersen, J.P., "A Comparative Study on Feature Selection in Text Categorization", *In Proceedings of the 14th International Conference on Machine Learning*, pp. 412-420, 1997.

23 Hwee, T.N., Wei, B. G. and Kok, L. L., "Feature Selection, Perceptron Learning and A Usability Case Study for Text Categorization", *In Proceedings*

of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, vol. 31, pp. 67-73, 1997.

24 Papadimitriou, C.H., Raghavan, P., Tamaki H. and Vempala, S., "Latent Semantic Indexing: A Probabilistic Analysis", *In Journal of Computer System Science*, vol. 61, issue 2, pp.217–235, 2000.

25 Eric, J. G., Kostas, T., Steve, L., David, M. P. and Gary, W. F., "Using Web Structure for Classifying and Describing Web Pages", *In Proceedings of the 11th international conference on World Wide Web (WWW-02)*, pp. 562-569, 2002.

26 Rish, I., "An Empirical Study of the Naive Bayes classifier", *In Proceedings of IJCAI-01 Workshop on Empirical Methods in Artificial Intelligence AI*, pp. 41–46, 2001.

27 Wang, Y., Julia, Hodges E. and Bo Tang, "Classification of Web Documents using a Naive Bayes Method", *In Proceedings of the 15th IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2003*, pp. 560-564, 2003.

Appendix A

List of stopwords used in our system

a	arial	between	differ	evenly
about	around	big	different	ever
above	as	both	differently	every
across	ask	browser	do	everybody
after	asked	but	does	everyone
again	asking	by	doesn	everything
against	asks	c	doing	everywhere
all	at	came	done	extens
almost	away	can	down	f
alone	b	cannot	downed	face
along	back	case	downing	faces
already	backed	cases	download	fact
also	backing	certain	downs	facts
although	backs	certainly	during	far
always	be	class	e	fax
among	became	clear	each	felt
an	because	clearly	early	few
and	become	click	either	file
another	becomes	color	email	find
any	been	com	e-mail	finds
anybody	before	come	end	first
anyone	began	content	ended	font
anything	behind	copyright	ending	for
anywhere	being	could	ends	found
are	beings	d	enough	four
area	best	data	error	frame
areas	better	did	even	from

full
fully
further
furthered
furthering
furthers
g
gave
general
generally
get
gets
gif
give
given
gives
go
going
good
goods
got
great
greater
greatest
group
grouped
grouping
groups
h
had
has
have
having
he
her
here
herself
high
higher
highest
him
himself
his
home
how
however
href
http
i
if
image
important
in
init
interest
interested
interesting
interests
into
is
it
its
itself
j
jpeg
just
k
keep
keeps
kind
knew
know
known
knows
l
label
large
largely
last
later
latest
least
length
less
let
lets
like
likely
link
links
list
long
longer
longest
m
made
mail
make
making
man
many
map
math
may
me
member
members
men
might
more
most
mostly
mr
mrs
msg
much
must
my
myself
n
necessary
need
needed
needing
needs
never
new
newer
newest
news
next
no
nobody
non
noone
not
nothing
now
nowhere
number

numbers
o
of
off
often
old
older
oldest
on
once
one
online
only
open
opened
opening
opens
or
order
ordered
ordering
orders
org
other
others
our
out
over
p
page
pages
part
parted
parting
parts
per
perhaps
place
places
point
pointed
pointing
points
possible
post
present
presented
presenting
presents
problem
problems
put
puts
q
quite
r
rather
really
right
rights
room
rooms
s
said
same
san
saw
say
says
second
seconds
see
seem
seemed
seeming
seems
sees
serif
several
shall
she
should
show
showed
showing
shows
side
sides
since
site
sites
small
smaller
smallest
so
some
somebody
someone
something
somewhere
src
state
states
still
substr
such
sure
t
take
taken
text
than
that
the
their
them
then
there
therefore
these
they
thing
things
think
thinks
this
those
though
thought
thoughts
three
through
thus
to
today
together
too
took
toward
try

turn	w	who	yet
turned	want	whole	you
turning	wanted	whose	young
turns	wanting	why	younger
two	wants	will	youngest
u	was	with	your
under	way	within	yours
until	ways	without	z
up	we	work	
upon	web	worked	
us	well	working	
use	wells	works	
used	went	worldwide	
uses	were	would	
usually	what	write	
v	when	www	
var	where	x	
version	whether	y	
very	which	year	
view	while	years	