Personalized Web Classifier

By

SANDIP J. MODHA

(06MCE009)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
INSTITUTE OF TECHNOLOGY
NIRMA UNIVERSITY OF SCIENCE & TECHNOLOGY,
AHMEDABAD 382481
MAY 2008

Major Project

On

Personalized Web Classifier

Submitted in partial fulfillment of the requirements

For the degree of

Master of Technology in Computer Science and Engineering

By

Sandip J. Modha
(06MCE009)

Under Guidance of

Dr. Dilip P. Ahalpara

Department of Computer Science and Engineering
Institute of Technology
Nirma University of Science & Technology,
Ahmedabad 382481
May 2008

This is to certify that Dissertation entitled

Personalized Web Classifier

Submitted by

Sandip J. Modha

has been accepted toward fulfillment of the requirement

for the degree of

Master of Technology in Computer Science & Engineering

Prof. (Dr.) S. N. Pradhan
Professor In Charge

Prof. D. J. Patel
Head of The Department

Prof. A. B. Patel
Director, Institute of Technology

CERTIFICATE

This is to certify that the Major Project entitled **"Personalized Web Classifier"** submitted by **Mr. Sandip J. Modha (06MCE009)**, towards the partial fulfillment of the requirements for the degree of Master of Technology in Computer Science and Engineering of Nirma University of Science and Technology, Ahmedabad is the record of work carried out by him under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for examination. The results embodied in this major project, to the best of my knowledge, haven't been submitted to any other university or institution for award of any degree or diploma.

Project Guide:-
Dr. Dilip P. Ahalpara
Scientist SF
Institute of Plasma Research
Ahmedabad

Date: / / 2008

ACKNOWLEDGEMENT

It gives me immense pleasure in expressing thanks and profound gratitude to **Dr**. **S.N Pradhan**, Professor, Computer Science Department, Institute of Technology, Nirma University, Ahmedabad for his valuable guidance and continual encouragement throughout my Major project part.

I would like to give my special thanks to Dr. **Dilip P. Ahalpara**, (Scientist SF, Institute of Plasma Research Ahmedabad), my guide for his encouragement and motivation throughout the Major Project part. I am heartily thankful to him for his precious time, suggestions and sorting out the difficulties of my topic that helped me a lot during this study. I am also thankful to **Prof. A. B. Patel**, Director, and Institute of Technology for his kind support in all respect during my study.

I am thankful to all faculty members of Computer Engineering Department, Nirma University, and Ahmedabad for their special attention and suggestions towards the project work.

The blessings of God and family members make the way for completion of Major Project. I am very much grateful to them.

I am thankful to my dear friends Mr. Utam Chauhan and Mr. Amit Lathigara who lent their ear when needed and always support and encourages me.

Sandip Modha

Roll No. 06MCE009

ABSTRACT

Web Page classification is a primary effort for the semantic of web, which helps us for the natural language processing (NLP). Web Page Classification comes under Web mining research area. Web mining is one of the emerging areas in the current world. Web-page classification is much more difficult than pure-text Classification due to a large variety of noisy information embedded in Web pages. 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. Recently Web-pages on the World Wide Web are explosively increasing, and it is now required for portal sites such as Yahoo! service having directory-style search engines to classify Web-pages into many categories automatically. Further we can also classify the web based upon our personal criteria. It is also helpful for the mining Search Engine (Google) results. In this report, a method for Web-page classification is implemented. Various Data Mining techniques like Association, Decision tree are applied in the field of Web Mining.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

Today Internet is the most popular and interactive medium to disseminate information. Number of people using internet is increasing tremendously. More and more people from corporate, researchers, scientists, student communities are uploading their work in internet. The World Wide Web (WWW) has become a popular interface to interact with Internet.

Internet has revolutionized the way in which we collect, view and classifies information. Speed and the ease of its use have made it a part of everyday life and work for most people. Unfortunately, much of the information on the Internet is highly unstructured and does not have any sort of predefined schema, type or pattern. The presentation of information has also become geared more towards visual aesthetics and user friendliness, making it appear more visually pleasing to human readers but at the same time making it hard for users to locate information on the Internet.

This unstructured and unorganized nature of the Internet as well as the sheer vastness of information available on it, has made it quite difficult for users to identify and extract “useful” or specific information on various topics from this large information base and viewing it in a structured way. This has led to users spending large amounts of time manually searching for information on the web and interpreting the results of their search efforts.

At present, the number of Web-pages on World Wide Web is increasing significantly. The tasks to find Web-pages which present information satisfying our requirements by traversing hyperlinks are difficult. Therefore, we use search engines frequently on the portal site.

With more than two billion pages created by millions of Web page authors and organizations, the World Wide Web is a tremendously rich knowledge base. The knowledge comes not only from the content of the pages

themselves, but also from the unique characteristics of the Web, such as its hyperlink structure, its multimedia content and its diversity of content and languages. Analysis of these characteristics often reveals interesting patterns and new knowledge. Such knowledge can be used to improve user's efficiency and effectiveness in searching for information on the Web, and also for applications unrelated to the Web, such as support for decision making or business management.

Machine learning techniques represent one possible approach to addressing the problem. Artificial intelligence and machine learning techniques have been applied in many important applications in both scientific and business domains, and data mining research has become a significant subfield in this area. Machine learning techniques also have been used in information retrieval (IR) and text mining applications. Various activities and efforts in this area are referred to as Web mining.

1.1 Web mining

The term Web mining was coined by Etzioni (1996) to denote the use of data mining techniques to automatically discover Web documents and services, extract information from Web resources, and uncover general patterns on the Web. Over the years, Web mining research has been extended to cover the use of data mining and similar techniques to discover resources, patterns, and knowledge from the Web and Web-related data (such as Web usage data or Web server logs). In this chapter, we have adopted a broad definition that considers Web mining to be "the discovery and analysis of useful information from the World Wide Web" Web mining research overlaps substantially with other areas, including data mining, text mining, information retrieval, and Web retrieval.

The evolution of the World Wide Web has brought us enormous and ever growing amounts of data and information. It influences almost all aspects of people's lives. In addition, with the abundant data provided by the web, it has become an important resource for research. Furthermore, the low cost of web data makes it more attractive. As the information is huge in

web, it is necessary to extract the important knowledge from it. Web mining research area is the mixture of the following research communities

I. Data Mining
II. Information Retrieval
III. Artificial Intelligence
IV. Natural Language Processing

1.2 Webpage classification Vs Text classification

The more general problem of text classification is beyond the scope of this work [4]. Compared with standard text classification, classification of web content is different in the following aspects. First, traditional text classification is typically performed on “structured corpora with well-controlled authoring styles”, while web collections do not have such a property. Second, web pages are semi-structured documents in HTML, so that they may be rendered visually for users. Although other document collections may have embedded information for rendering and/or a semi-structured format, such markup is typically stripped for classification purposes. Finally, web documents exist within a hypertext, with connections to and from other documents. While not unique to the web (consider for example the network of scholarly citations), this feature is central to the definition of the web, and is not present in typical text classification problems. Therefore, web classification is not only important, but distinguished from traditional text classification, and thus deserving of the focused review found in this report.

1.3 Need for Web Mining

Information user currently encounter following problems while interacting with web

I. Finding relevant information: People either browse or use the search service when they want to find specific information from the on the web. When user uses a search service, he usually inputs single keyword query and the query response is the set of pages rank based

upon the similarities to the query. Today's search engines have the following problem

a. Low precision : Due to the irrelevance of many search results
b. Low recall: Due to the inability to index all the information available on the web.

II. Knowledge discovery : The main objective is to extract the potential information from the Web page collection;

III. Personalization of Information: The problem is often associated with the type and presentation of information, since it is likely that people differ in the content and presentation they prefer while interacting with web.

IV. Web mining techniques could be used to solve above problem directly or indirectly

1.4 Types of Web Mining Techniques

Web mining is the use of data miming technique to automatically discover and extract information from the web pages. Web mining research can be divided into three categories [6]:

I. Web Content Mining
II. Web Structure Mining
III. Web Usage Mining

Web content mining refers to the discovery of useful information from Web content, including text, images, audio, and video. Web structure mining studies potential models underlying the link structures of the Web. It usually involves the analysis of in-links and out-links, and has been used for search engine result ranking and other Web applications. Web usage mining focuses on using data mining techniques to analyze search or other activity logs to find interesting patterns. One of the main applications of Web usage mining is to develop user profiles.

1.4.1 Web Content Mining

Web content mining is an automatic process that goes beyond keyword extraction. In other words the goal of Web content Mining is “Extract “snippets” from a Web document that represents the Web Document”. Since the content of a web document presents no machine readable semantic, some approaches have suggested restructuring the document content in a representation that could be exploited by machines. The usual approach to exploit known structure in documents is to use wrappers to map documents to some data model. Web content mining refers to the discovery of useful information from Web content, including text, images, audio, and video. Web content mining research includes resource discovery from the Web document categorization and clustering, and information extraction from Web pages.

1.4.1.1 Relevance of content

Relevance [6] can be measured with respect to following criteria:

I. Document relevance: Measure of how useful a given document is in a given situation commonly seen in the context of queries -results are ordered by some measure of relevance.
II. Query based relevance: This is most common relevance technique. In these technique similarities between query keyword and document is calculated.
III. User based relevance: This is personalization relevance technique. In this technique profile of user is created. Similarity between document and profile is calculated.

1.4.1.2 Application of Web content mining

Following are the applications of web content mining:

I. Identify topics represented by the web document.
II. Categorize the web document.
III. Find web pages across the different server that are similar
IV. Application related to relevance.
V. Web-Page classification

1.4.2 Web Structure Mining

In recent years, Web link structure [6] has been widely used to infer important information about Web pages. Web structure mining has been largely influenced by research in social network analysis. Web structure mining is the process of using graph theory to analyze the node and connection structure of a web site. Web structure mining studies potential models underlying the link structures of the Web. It usually involves the analysis of in-links and out-links, and has been used for search engine result ranking and other Web applications. According to the type of web structural data, web structure mining can be divided into two kinds. The first kind of web structure mining is extracting patterns from hyperlinks in the web. A hyperlink is a structural component that connects the web page to a different location. The other kind of the web structure mining is mining the document structure. It is using the tree-like structure to analyze and describe the HTML (Hyper Text Markup Language) or XML (extensible Markup Language) tags within the web page.

Among various Web-structure mining algorithms, PageRank [7] and HITS (Hyperlinked Induced Topic Search) are the two most widely used. The PageRank algorithm is computed by weighting each in-link to a page proportionally to the quality of the page containing the in-link (Brin & Page, 1998). The qualities of these referring pages also are determined by PageRank. Thus, the PageRank of a page p is calculated recursively as follows.

$$PageRank(p) = (1 - d) + d \times \sum_{\substack{all\ q\ linking \\ to\ p}} \left(\frac{PageRank(q)}{c(q)} \right)$$

Where d is damping factor between 0 and 1.

C (q) is number of outgoing link in a page q.

Following are the application of web structure mining

I. Quality of web page
    a. The authority of page on a topic.
    b. Ranking of web page.

II. Web-Page classification
    a. Classify the web pages according to various topic
III. Extracting web structure

1.4.3 Web usage mining

Web usage mining focuses on a technique that could predict user behavior while the user interacts with web. Web usage mining is the application that uses data mining to analyze and discover interesting patterns of user's usage data on the web. The usage data records the user's behavior when the user browses or makes transactions on the web site, in order to better understand and serve the needs of users or Web-based applications. It is an activity that involves automatic discovery of patterns from one or more Web servers. Organizations often generate and collect large volumes of data; most of this information is usually generated automatically by Web servers and collected in server log. Analyzing such data can help these organizations to determine the value of particular customers, cross marketing strategies across products and the effectiveness of promotional campaigns, etc.

The application of web usage mining could be classified in two main categories:

I. Learning user profile or user modeling in adaptive interface
II. Learning user navigation pattern

Web user would be interesting in the information they want. On the other hand information provider would be interested in learning user's way of navigation style so they can design the web according to feedback. In other word they are interested in user's navigational pattern. Then the learned knowledge is useful for personalization of web portal. It also help in business intelligence, system improvement.

1.5 Applications

Web mining is useful in a wide range of applications. It is used in B2C e-commerce in the analysis of user's past behavior and peer group analysis for personalized messages and category recommendations and also used

in clustering, association analysis and temporal sequence analysis. It is used in search engines in content analysis and hyperlink analysis. It is useful in understanding user communities for interests and opinions of group members, understanding auction behavior to categorize the participants into various types, classifying auctions into various bids, and determining fraudulent bids. Web mining is also used in personalized web portals to create personalized messages and deliver media content based on preference and usage. It also helps to analyze the effectiveness of websites thus enabling companies to target customers through segmentation, acquire customers based on their propensity to purchase, analyze purchase behavioral pattern, address privacy and security concerns and correlate online behavior with sales transactions.

1.6 Challenges to Web Mining

There are quite a few challenges [6] in Web mining:

I. Most of the Web documents are in HTML format and contain many mark-up tags mainly used for formatting.
II. Traditional IR systems often contain structured and well-written documents, which is not the case on the Web. Basically, Web documents are more diverse in terms of length, document structure, writing style and many Web pages contain syntactic errors.
III. Web has different types of content including text, images, audio, video in various formats such as HTML, XML, PDF, MS Word, mp3, wav and avl Web applications have to deal with these different formats to retrieve the desired information.
IV. While most of the documents tend to remain static over time, Web pages are much more dynamic and they can be updated every day, every hour or even every minute. Some Web pages do not even have a static form; they are dynamically generated on request, with content varying according to the user and the time of the request. Such dynamics make it much more difficult for retrieval systems such as search engines to keep an up-to-date search index of the Web.

Another characteristic of the Web, perhaps the most important one, is the hyperlink structure. Web pages are hyperlinked to each other, and it is through hyperlink that a Web page author cites other Web pages. Lastly, the size of the Web is larger than traditional data sources or document collections by several orders of magnitude. The number of index able Web pages has exceeded 2 billion, and has been estimated to be growing at the rate of roughly 1 million pages per day. Collecting, indexing, and analyzing these documents presents a great challenge. Similarly, the population of Web users is much larger than that of traditional information systems. Collaboration among users can be more feasible because of the availability of a large user base, but it can also be more difficult because users are more diverse. Most of the web mining activities are still in their early stages and they should continue to develop as the Web evolves. Future directions of Web mining include multimedia data mining, semantic Web mining, wireless Web mining and invisible Web mining.

1.8 Outline of Thesis

The thesis is organized as follows. Chapter 1 will begin by introducing the area of dissertation. Chapter 2 explains various classification schemes and feature selection for the PWS (Personalized Web Classifier). It also describes the various applications of the PWS. In chapter 3, the proposed design is suggested. Various models are used to describe the PWS. Apriori algorithm is explained in detail with example in chapter 3.

Chapter 4 focuses on implementation part and its result via various graphs and tables. It also gives brief overview of the libraries used in development work. Appropriate conclusions and future work are discussed in Chapter 5.

2. PWS CONCEPT AND RELATED WORK

Before reviewing web Page classification research, we first introduce the problem, motivate it with applications, and consider related surveys in web classification.

2.1 Related System

Web page classification, also known as web page categorization, is the process of assigning a web page to one or more predefined category labels. Classification is often posed as a supervised learning problem (Mitchell 1997) in which a set of labeled data is used to train a classifier which can be applied to label future examples.

The general problem of web page classification can be divided into multiple sub-problems [4]: 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 (e.g., (zu Eissen and Stein 2004)), search engine spam classification (e.g., (Gy¨ongyi and Garcia-Molina 2005b; Castillo, Donato, Gionis, Murdock, and Silvestri 2007)) and so on. This work focuses on subject and functional classification.

2.2 Different Classification Schemes

Based on the number of classes in the problem, classification [4] can be divided into

I. binary classification
II. multi-class classification

Binary classification categorizes instances into exactly one of two classes (as in Figure 1(a)); 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 either single-label, where exactly one class label can be assigned to an instance (as in Figure 1(b)), or multi-label, where an instance can belong to any one, two, or all of the classes (as in Figure 1(c)). 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 Figure 1(d)).

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., one category does not supersede another. While in hierarchical classification, the categories are organized in a hierarchical tree-like structure, in which each category may have a number of subcategories. An illustration is shown in Figure 2. Section 4 will address the issue of hierarchical classification further.

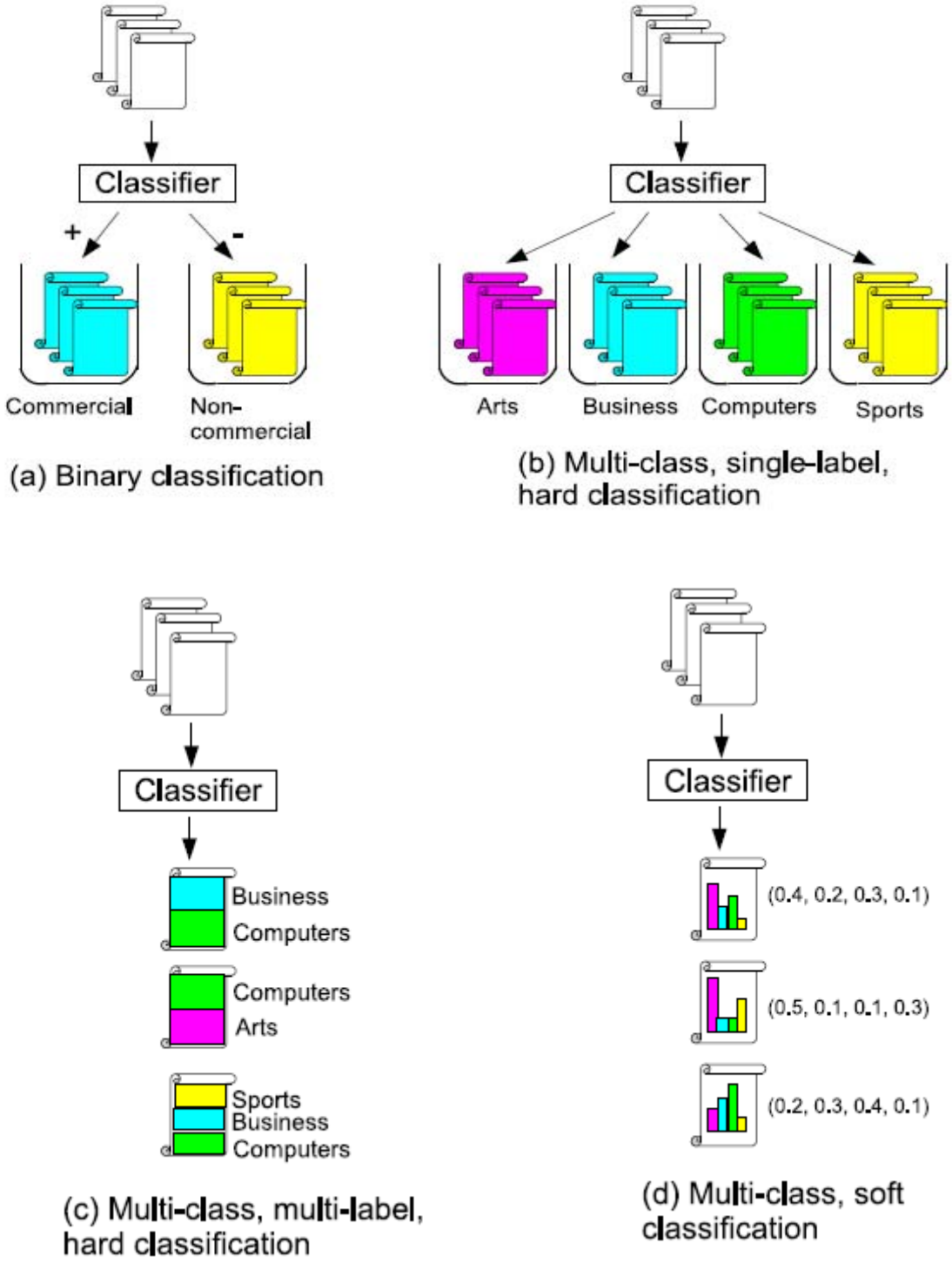


Fig 2.1 Type of Classification Technique [4]

2.3 Feature Selection

Written in HTML, web pages contain additional information, such as HTML tags, hyperlinks and anchor text (the text to be clicked on to activate and follow a hyperlink to another web page, placed between HTML <A> and </A> tags), other than the textual content visible in a web browser. These features can be divided into two broad classes: on-page features, which are directly located on the page to be classified, and features of neighbors, which are found on the pages related in some way with the page to be classified. We will discuss the various features available on web page in below section.

2.3.1 Meta Information

Every web page has a Meta data [3] written by the Web page author. There are three main types of Meta data under various tags.

I. Title: This tag will use to write the title of the web page. In every web page the upper right most corner will display the title of web page.
II. Description: This Meta word will give the description of the web page.
III. Keyword: This Meta word will give you the list of keyword used in the document.

So selecting Meta data we can classify the web page.

2.3.2 Using the feature of neighbors

Although web pages contain useful features as discussed above, in a particular web page these features are sometimes missing, misleading, or unrecognizable for various reasons. For example, some web pages contain large images or flash objects but little textual content, such as in the example shown in Figure 3. In such cases, it is difficult for classifiers to make reasonable judgments based on features on the page. In order to address this problem, features can be extracted from neighboring pages [1] that are related in some way to the page to be classified to supply supplementary

information for categorization. There are a variety of ways to derive such connections among pages. One obvious connection is the hyperlink. Since most existing work that utilizes features of neighbors is based on hyperlink connection, in the following, we focus on hyperlinks connection. However, other types of connections can also be derived; and some of them have been shown to be useful for web page classification.

Another question when using features from neighbors is that of which neighbors to examine. Existing research mainly focuses on pages within two steps of the page to be classified. At a distance no greater than two, there are six types of neighboring pages according to their hyperlink relationship with the page in question: parent, child, sibling, spouse, grandparent and grandchild, as illustrated in Figure 4. The effect and contribution of the first four types of neighbors have been studied in existing research. Although grandparent pages and grandchild pages have also been used, their individual contributions have not yet been specifically studied. In the following, we group the research in this direction according to the neighbors that are used.

In general, directly incorporating text from parent and child pages into the target page does more harm than good because parent and child pages are likely to have different topics than the target page (Chakrabarti, Dom, and Indyk 1998; Ghani, Slattery, and Yang 2001; Yang, Slattery, and Ghani 2002). This, however, does not mean that parent and child pages are useless. The noise from neighbors can be greatly reduced by at least two means: using an appropriate subset of neighbors, and using an appropriate portion of the content on neighboring pages. Both methods have been shown to be helpful.

Using a subset of parent and child pages can reduce the influence from pages on different topics than the target page. For example, while utilizing parent and child pages, Oh et al. (2000) require the content of neighbors to be

sufficiently similar to the target page. Using a portion of content on parent and child pages, especially the content close enough to the hyperlink that points to the target page, can reduce the influence from the irrelevant part of neighboring pages. Usually, title, anchor text, and the surrounding text of anchor text on the parent pages are found to be useful. This family of approaches takes advantage from both hyperlinks and HTML structure information. Below, we review some existing approaches of this type.

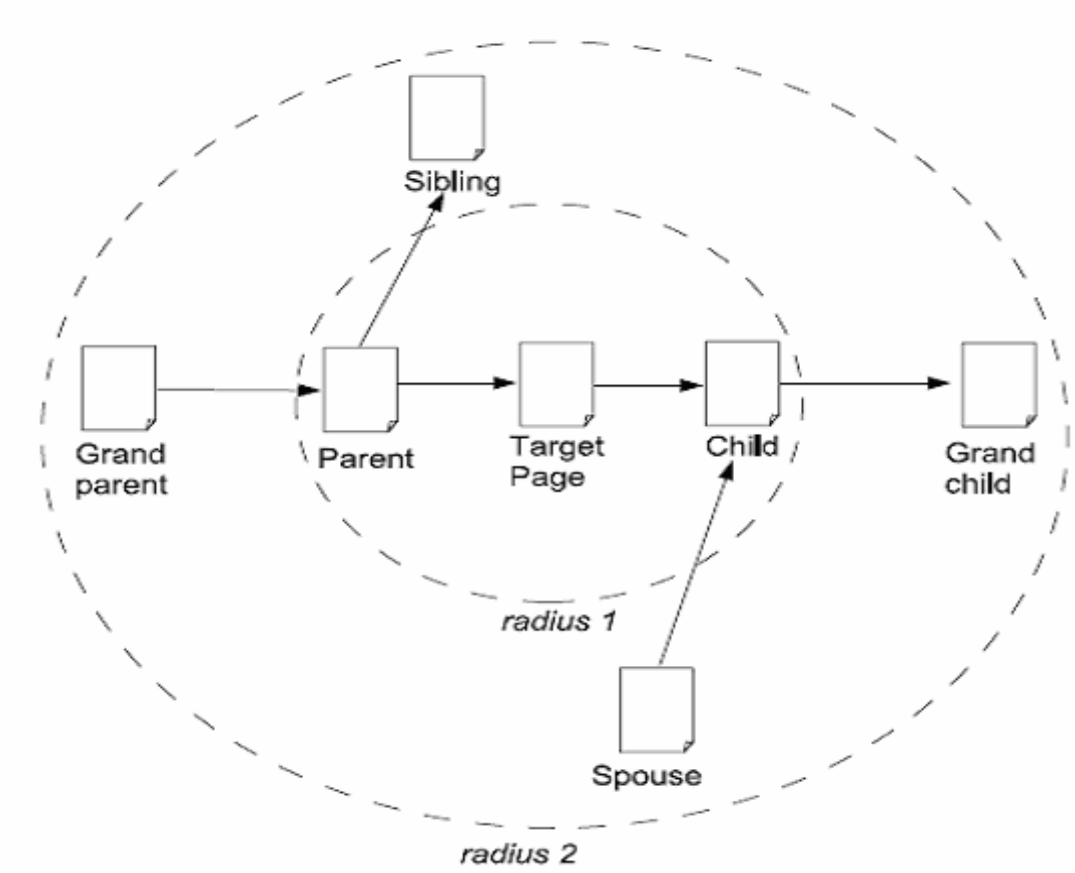


Fig 2.2 Neighbors of Webpage with radius 2 [4]

2.4 Applications of webpage classification (PWS)

As briefly introduced in Section 2.1, classification of web content is essential to many information retrieval tasks. Here, we present a number of such tasks [4].

2.4.1 Constructing web directories (web hierarchies)

Web directories, such as those provided by Yahoo! (2007) [4] and the dmoz Open Directory Project (ODP) (2007), provide an efficient way to browse for information within a predefined set of categories. Currently, these directories are mainly constructed and maintained by editors, requiring extensive human effort. As of July 2006, it was reported (Corporation 2007) that there are 73,354 editors involved in the dmoz ODP. As the Web changes and continues to grow, this manual approach will become less effective. One could easily imagine building classifiers to help update and expand such directories. For example, Huang et al. (2004a, 2004b) propose an approach to automatic creation of classifiers from web corpora based on user-defined hierarchies. Further more, with advanced classification techniques, customized (or even dynamic) views of web directories can be generated automatically. There appears to be room for further interesting work along this direction

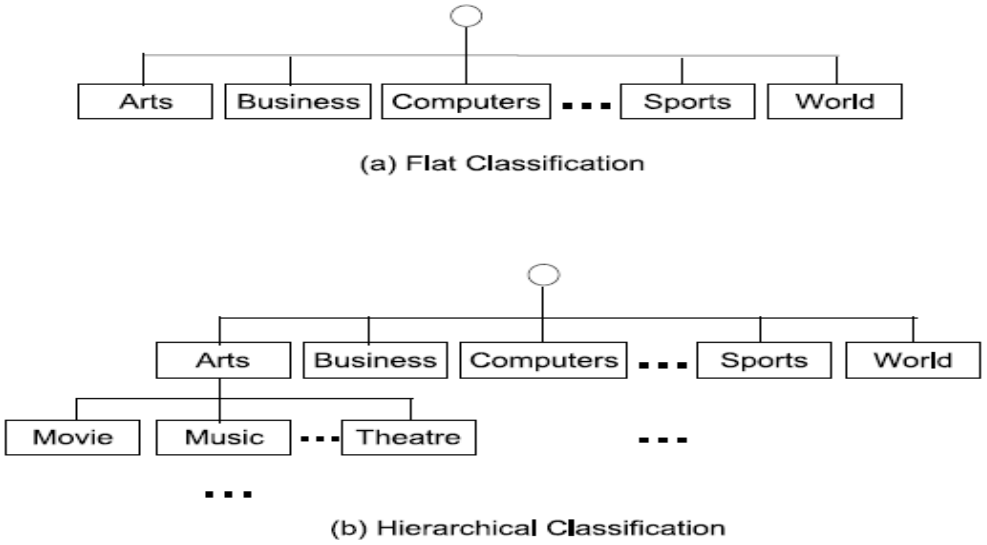


Fig 2.3 Flat Classification and Hierarchical Classification [4]

2.2.2 Improving quality of search results

Query ambiguity is among the problems that undermine the quality of search results. For example, the query term “bank” could mean the border of a water area or a financial establishment. Various approaches have been proposed to improve retrieval quality by disambiguating query terms. Chekuri et al. (Chekuri et al. 1997) studied automatic web page classification in order to increase the precision of web search. A statistical classifier, trained on existing web directories, is applied to new web pages and produces an ordered list of categories in which the web page could be placed. At query time the user is asked to specify one or more desired categories so that only the results in those categories are returned, or the search engine returns a list of categories under which the pages would fall. This approach works when the user is looking for a known item. In such a case, it is not difficult to specify the preferred categories. However, there are situations in which the user is less certain about what documents will match, for which the above approach does not help much.

Search results are usually presented in a ranked list. However, presenting categorized, or clustered, results could be more useful to users. An approach proposed by Chen and Dumais (2000) classifies search results into a predefined hierarchical structure and presents the categorized view of the results to the user. Their user study demonstrated that the category interface is liked by the users better than the result list interface, and is more effective for users to find the desired information. Compared to the approach suggested by Chekuri et al., this approach is less efficient at query time because it categorizes web pages on-the-fly. However, it does not require the user to specify desired categories; therefore, it is more helpful when the user does not know the query terms well. Similarly, K¨aki (2005) also proposed to present a categorized view of search results to users. Experiments showed that the categorized view is beneficial for the users, especially when the ranking of results is not satisfying.

In 1998, Page and Brin developed the link-based ranking algorithm called Page Rank (1998). Page Rank calculates the authoritativeness of web pages based on a graph constructed by web pages and their hyperlinks, without considering the topic of each page. Since then, much research has been explored to differentiate authorities of different topics. Haveliwala (2002) proposed Topic-sensitive Page Rank, which performs multiple Page Rank calculations, one for each topic. When computing the Page Rank score for each category, the random surfer jumps to a page in that category at random rather than just any web page. This has the effect of biasing the Page Rank to that topic. This approach needs a set of pages that are accurately classified. Nie et al. (2006) proposed another web ranking algorithm that considers the topics of web pages. In that work, the contribution that each category has to the authority of web pages is distinguished by means of soft classification, in which a probability distribution is given for a web page being in each category. In order to answer the question "to what granularity of topic the computation of biased page ranks make sense," Kohlschutter et al. (2007) conducted analysis on ODP categories, and showed that ranking performance increases with the ODP level up to a certain point. It seems further research along this direction is quite promising.

2.2.3 Page Relevance

Generally Search Engine use Meta data, which present in every web page, to find out the web pages across the internet [7]. When we user search engine to find the web pages according to our search criteria, We often get the pages which are not relevant or which we do not wanted. There are many reasons for above described situation.

Most of search engine compare user‘s specified word with the meta data which present in the web page. If Meta data match with the user's specified query then web page is returned to user without no matter what content is present in web page. During the survey of the different class of the web

pages, it is observed that number of web pages content does not match with Meta data.

Due to heavy commercialization of Internet and to reach to the end user, Web author put booming word in their Meta data and also put the generalize word rather than specific word. So that they will get higher rank in search engine's results. That's why we encounter unwanted web page given by search engine.

In this dissertation work, one module has been designed which will check the relevancy of web page according to user query. So the Proposal for the search engine design is rather than use Meta data to search web page, use following point.

I. Summarize the web pages based upon the content of it.
II. Make a pool of all word (especially noun) that make summery.
III. Use the pool of word to match with user query or search criteria.

Instead of using Meta data, Summarize word will help to increase the page relevancy. User will get the information what he is looking for. Using this approach, we can improve the quality of search. We can also increase the efficiency of the search user.

Using Page Relevancy we can rank the web documents which are similar. In the internet we find many document which are similar as per the meta data. But they slightly differ by the content. So we can rank such similar class of web page based upon the content.

2.2.4 Designing a Firewall

Firewall helps organization to restrict user to access particular Website or some class of website. Generally most of the Firewall restricts the website base upon the following two criteria.

I. System administrator specifies the list of Known URL to firewall directory that use by firewall as restricted access.
II. The other way is System Administrator give the list of keyword to the firewall. If any website's URL matches with list of this keyword then this website is restricted for access.

The above approaches work fine for some situation but in some situation It fails. In the following scenario existing firewall fails.

I. Some Website does not contain any word which belongs to any words specified by the firewall still it comes under restricted for access category.
II. Some website might contain the images or video that can not be resolved by existing Firewall design.

To avoid such situation, Content based classification is necessary for the when designing firewall. When any web page is requested, download the page at firewall, apply the personalize web page classifier to resolve the content. So using this approach organization can strengthen its policy for internet browsing. One can also filter the E-mail.

3. THE PROPOSED SYSTEM DESIGN

The goals of this dissertation work are as follow:

I. Personalized Web Page classifier: This module will automatically classify the web page into user defined various classes, sub-class. E.g. Sports, Entertainment, Social, Business etc.

II. Page Relevancy Checker: This module will check whether the given web page is relevant according to user criteria or not, based upon the its content analysis

3.1 Personalized Web Page Classifier(PWS)

Basically this module classifies WebPages in different classes based upon the content. It also classifies WebPages at subclasses level. In this project following classes and subclasses are considered:

Table 3.1 List of Classes and sub-classes

Classes	**Sub-Classes**
Sports	Cricket , Football etc.
Business	Stock Market, General business
Entertainment	Movie, Music etc.
Academic	Various classes like university, Subjective classes
General	Political, Judiciary, Social,etc.

During the design part, many webpage of different class are analyzed in detail. Before we discuss the design, let's first analyze a typical webpage.

3.2 Analysis of Web Page

To analyze the webpage structure we have to open HTML webpage in to text file. As shown in fig 3.1 Web Page divided in many section. In HTML typically tag term used to denote the section.

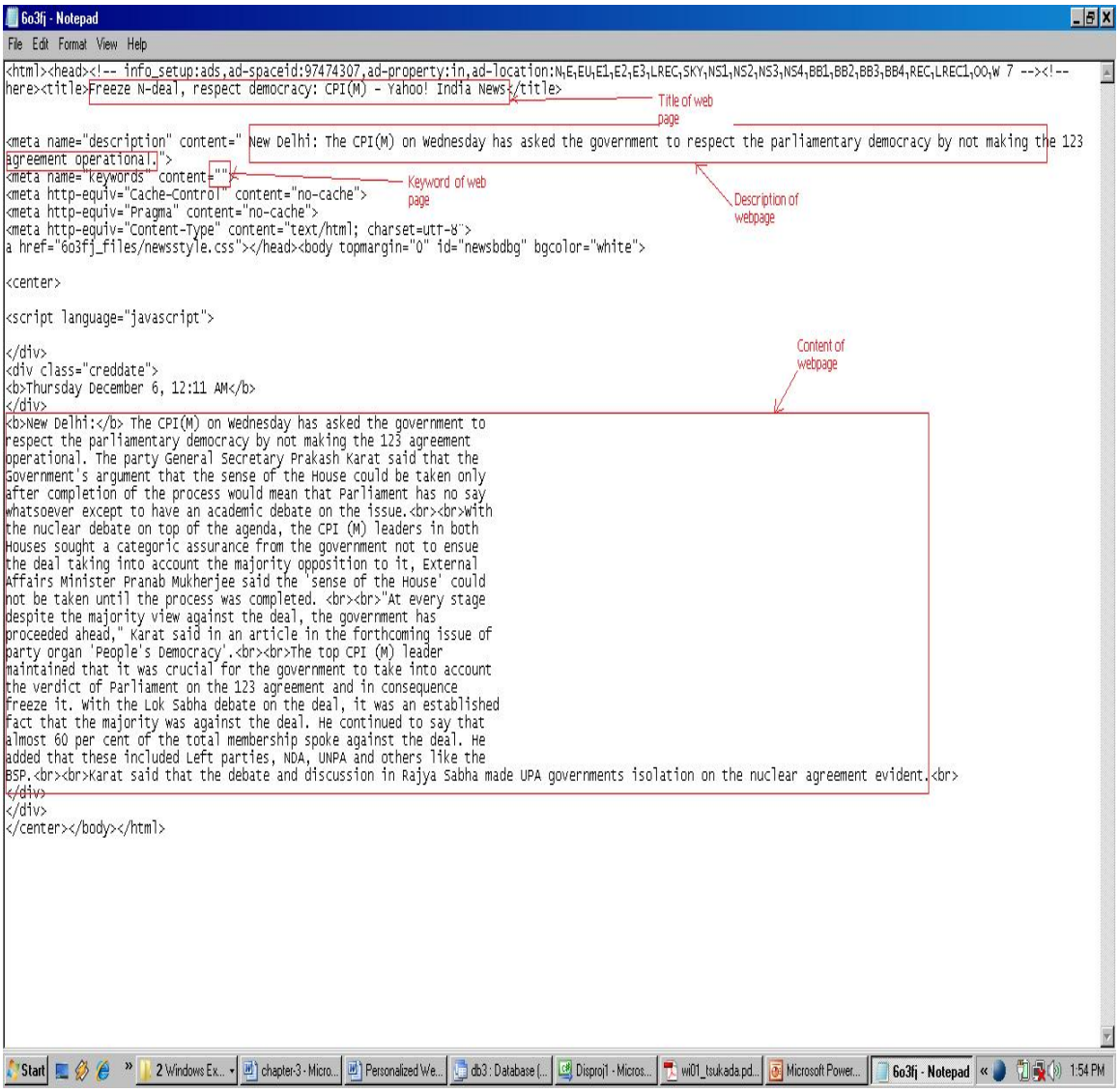


Figure 3.1 Structure of Standard Web Page

As shown in fig 3.1, every webpage has two main Tags

I. <Head> Tag : <Head> tag contain information about the web page. It describes the web page. It also contains the Meta information which is used by the Search Engine (e.g. Google, Yahoo, etc.). The following tags come under the <Head> Tag. <Head> tag contains the <Meta> tag which represent the following metadata:

    a. Title: This tag represents the title of the web page. In browser every Webpage's uppermost left corner display the title tag.
    b. Description: This tag is used to provide the information about the subject of webpage.
    c. Keyword: This tag lists all the important word which is represented by the webpage content.

II. At the end this tag is terminated by </Head>.

III. <Body> Tag : <Body> tag will represent the content of the web page. In <body> tag there are other information like the advertises in form of hyperlink, Images, video etc. Earlier the WebPages were mainly static. But nowadays, with the introduction of scripting language like java script, VB script, etc., WebPages become dynamic. Therefore lots of scripting code in the body tag. This tag is ended by </body>.

Truly speaking nowadays Webpage contains so much noisy data. Internet has become a new media for marketing of products. Due to this, it contains many objects like images, banner, ads, logo which may be irrelevant with respect to main content. So, this adds difficult in extracting the main content of the web page.

3.3 System Design

Design phase consist of following sub-phase.

I. Extraction of Meta Data from the Web-Page.
II. Finding out Itemset from the page
III. Application of Apriori algorithm to find out frequent Itemsets
IV. Application Of decision tree to generate class.

3.3.1 Extraction of Meta Data

Initially a small set of pages (say 50) of each category are downloaded, like Sports, Entertainment, Social, and Business etc. Each class of pages are kept into different folders in local drive.

$$\mathbf{Class_C \leftrightarrow \{Page_C[1]. . . , Page_C[i] , . . .\},}$$

Where $Page_C[i]$ indicates i^{th} Web-page labeled $Class_c$.

All of tags such as <A HREF > and <IMG > are ignored from documents of Web-pages described by Hyper Text Markup Language.

Next following information is considered from webpage:

I. Title
II. Keyword
III. Description

All the nouns from the above field of the web page are then extracted. A union is then made of the all the noun. It is assumed that nouns in the above fields reflect the theme of Webpage. Each noun is then referred as an item and forms a transaction $Page_c^i$ which consists of some items $Word^c_{i\ j}$ as follows.

$$\mathbf{Page_C[i] = < Word_C[i][1]\ldots\ldots\ldots\ldots,Word_C[i][j]\ldots\ldots>}$$

Where $word_C[i][j]$ indicates the j^{th} item extracted from $Page_C$ [i] labeled as $class_C$. In addition, we integrate them into a set of transactions for each class label.

e.g. P_1 = {N-deal, CPI (M), Government, Freeze, Democracy, 123, agreement, Operational}

3.3.2 Generation of Itemsets

After retrieving all the nouns from the Meta data, the <Body> tag of the Web page is then considered. Each sentence is considered one by one that actually make the content of webpage. The goal here is that in any sentence successive word are compared with the set of Meta data.

If word occur than we represent the sentence with a word which match with our Meta data set. If any word of sentence does not match with Meta data set than we will not consider this statement. We assume that sentences are the transaction of the nouns of Meta data set.

In this way we find various Itemsets from all WebPages. After that we integrate all the itemset as per below [5].

$$\mathbf{\{Itemset_1^1, Itemset_2^1, Itemset_3^1 \ldots\ldots\ldots\ldots\ldots\ldots Itemset_L^C \ldots\ldots\}}$$

Where $Itemset_C^I$ indicates the I^{th} Itemset extracted from the sets of Transactions labeled $class_C$.

3.3.4 Examples of Generation of Itemsets.

To understand the above process let's take one example of a webpage. Consider that the webpage contains following Meta data:

I. Title : Freeze N-deal, respect democracy: CPI(M) - Yahoo! India News
II. Description: "New Delhi: The CPI (M) on Wednesday has asked the government to respect the parliamentary democracy by not making the 123 agreement operational."
III. Keyword: N-deal, CPI (M), Government, Democracy, 123 agreements.

Now our job is to make the union of the nouns which are present in above metadata.

Meta-Set = {N-deal, CPI (M), Government, Freeze, Democracy, 123, agreement, Operational}

Thus the set of all nouns which make up the entire content is constructed. Now, examine every statement and check every word with our meta-set. If found then we represent the statement with word of Meta set. For more information see the below table:

Table 3.2 List of Itemsets in webpage

Sentence	Itemsets
S1	{123, Agreement, CPI(M), Government, Democracy, Freeze }
S2	{ Government }
S3	{CPI(M),Government ,N-Deal}
S4	{Government, N-Deal, Democracy}
S5	{123, Agreement, CPI(M),Democracy, Operational}
S6	{N-Deal, CPI(M)}
S7	{N-Deal}
S8	{123 , Agreement}

So at the end of 2nd phase we get above output, now we have to apply Apriori algorithm to identify the frequent Itemsets which can be used to classify the document.

3.3.5 Apriori algorithm

Apriori is a classic algorithm [2] for finding frequent Itemsets. Apriori is designed to operate on database containing transactions (for example, collections of items bought by customers, or details of a website frequentation). Given a set of Itemsets (for instance, sets of retail transactions, each listing individual items purchased), the algorithm attempts to find subsets which are common to at least a minimum number C (the cutoff, or confidence threshold) of the Itemsets. Apriori uses a "bottom up" approach, where frequent subsets are extended one item at a time (a step known as candidate generation), and groups of candidates are tested against the data. The algorithm terminates when no further successful extensions are found.

Apriori uses breadth-first search and a hash tree structure to count candidate item sets efficiently. It generates candidate item sets of length k from item sets of length k − 1. Then it prunes the candidates which have an infrequent sub pattern. According to the downward closure lemma, the candidate set contains all frequent k-length item sets. After that, it scans the transaction database to determine frequent item sets among the candidates. For determining frequent items quickly, the algorithm uses a hash tree to store candidate Itemsets. This hash tree has item sets at the leaves and hash tables at internal nodes (Zaki, 99). Apriori, while historically significant, suffers from a number of inefficiencies or trade-offs, which have spawned other algorithms. Candidate generation generates large numbers of subsets (the algorithm attempts to load up the candidate set with as many as possible before each scan). Bottom-up subset exploration (essentially a breadth-first traversal of the subset lattice) finds any maximal subset S only after all 2 | S | − 1 of its proper subsets.

Apriori algorithm is Iterative type algorithm (also called level-wise search): Find all 1-item frequent Itemsets; then all 2-item frequent Itemsets, and so on. In each iteration k, only consider Itemsets that contain some k-1 frequent itemset.

I. Find frequent Itemsets of size 1: F_1
II. From k = 2
    a. C_k = candidates of size k: those Itemsets of size k that could be frequent, given Fk-1
    b. F_k = those Itemsets that are actually frequent, $F_k \subseteq C_k$ (need to scan the database once)

Algorithm Pseudo code

```
Algorithm Apriori(T)          // T is  set of Itemsets represents Transactions
    C1 ← init-pass(T);
    F1 ← {f | f ∈ C1, f.count/n ≥ minsup};    // n: no. of transactions in T
    for (k = 2; Fk-1 ≠ ∅; k++) do
        Ck ← candidate-gen(Fk-1);
        for each transaction t ∈ T do
            for each candidate c ∈ Ck do
                if c is contained in t then
                    c.count++;
            end
        end
        Fk ← {c ∈ Ck | c.count/n ≥ minsup}
    end
Return F ← Uk Fk;
```

Here

C_k : K-Itemset at K-level;

F_k: K-Itemset after applying minimum support on C_k

T : Set Of Itemsets Transactions

The candidate-gen)unction takes F_{k-1} and returns a superset (called the candidates) of the set of all frequent *k*-Itemsets. It has two steps

I. **join step**: Generate all possible candidate Itemsets C_k of length *k*

II. **prune step**: Remove those candidates in C_k that cannot be frequent

Candidate-gen function

Function candidate-gen(F_{k-1})

$C_k \leftarrow \varnothing$;
For all $f_1, f_2 \in F_{k-1}$
With $f_1 = \{i_1 \ldots i_{k-2}, i_{k-1}\}$
And $f_2 = \{i_1 \ldots i_{k-2}, i'_{k-1}\}$
And $i_k\text{-}1 < i'k\text{-}1$ do
$c \leftarrow \{i_1, \ldots, i_{k-1}, i'_{k-1}\}$; // join f_1 and f_2
$C_k \leftarrow C_k \cup \{c\}$;
For each (k-1)-subset s of c do
If ($s \notin F_{k-1}$) then
delete c from C_k; // prune
End
End
Return C_k;

Here

C_k : K-Itemset at K-level;
F_k: K-Itemset after applying minimum support on C_k

3.3.6 Examples of Apriori Algorithm

To understand the working of Apriori algorithm, let's take one example. Consider the table 3.2 that we have derived in phase II. Our goal is to find frequent Itemsets. We will consider this table as a dataset,

Sentence	Itemsets
S1	{123, Agreement, CPI, Government, Democracy, Freeze }
S2	{ Government }
S3	{CPI, Government ,N-Deal}
S4	{Government, N-Deal, Democracy}
S5	{123,Agreement,CPI,Goverment,Democracy, Operational}
S6	{N-Deal, CPI}
S7	{N-Deal}
S8	{123 , Agreement}

As per the algorithm, first we have to find C1 and from that we will apply the Minimum support level which is denoted by Minsup. In this example we will consider Minsup as 15%. Here don't consider the Minsup at 1-level.

Level -1

Table 3.3(a) candidate at 1-level Table 3.3(b) Frequent Itemsets at 1-level

C_1

Itemset	Count	Status
{FREEZE}	1	Y
{DEAL}	5	Y
{DEMOCRACY}	2	Y
{CPI}	3	Y
{DELHI}	1	Y
{GOVERNMENT}	5	Y
{PARLIAMENTARY}	1	Y
{123}	2	Y
{AGREEMENT}	2	Y
{OPERATIONAL}	1	Y

F_1

Itemset	Count	Status
{FREEZE}	1	Y
{DEAL}	5	Y
{DEMOCRACY}	2	Y
{CPI}	3	Y
{DELHI}	1	Y
{GOVERNMENT}	5	Y
{PARLIAMENTARY}	1	Y
{123}	2	Y
{AGREEMENT}	2	Y
{OPERATIONAL}	1	Y

Level - 2

From F_1 we derive C_2. After applying Minsup 15% we derive F_2

Table 3.4(a) candidate at 2nd-level

C_2

Itemset	Count	Status
{FREEZE,CPI}	1	N
{FREEZE,GOVERNMENT}	1	N
{FREEZE,123}	1	N
{FREEZE,AGREEMENT}	1	N
{DEAL,DEMOCRACY}	1	N
{DEAL,CPI}	1	N
{DEAL,GOVERNMENT}	2	Y
{DEMOCRACY,CPI}	1	N
{DEMOCRACY,DELHI}	1	N
{DEMOCRACY,GOVERNMENT}	2	Y
{DEMOCRACY,PARLIAMENTARY}	1	N
{DEMOCRACY,123}	1	N
{DEMOCRACY,AGREEMENT}	1	N
{DEMOCRACY,OPERATIONAL}	1	N
{CPI,DELHI}	1	N
{CPI,GOVERNMENT}	3	Y
{CPI,PARLIAMENTARY}	1	N
{CPI,123}	2	Y
{CPI,AGREEMENT}	2	Y
{CPI,OPERATIONAL}	1	N
{DELHI,GOVERNMENT}	1	N
{DELHI,PARLIAMENTARY}	1	N
{DELHI,123}	1	N
{DELHI,AGREEMENT}	1	N
{DELHI,OPERATIONAL}	1	N
{GOVERNMENT,PARLIAMENTARY}	1	N
{GOVERNMENT,123}	2	Y
{GOVERNMENT,AGREEMENT}	2	Y
{GOVERNMENT,OPERATIONAL}	1	N
{PARLIAMENTARY,123}	1	N
{PARLIAMENTARY,AGREEMENT}	1	N
{PARLIAMENTARY,OPERATIONAL}	1	N
{123,AGREEMENT}	2	Y
{123,OPERATIONAL}	1	N
{AGREEMENT,OPERATIONAL}	1	N

Now, apply Minsup which gives 15% of total no of statement and derive F2.

Table 3.4(b) Frequent Itemsets at 2nd-level

F_2

Itemset	Count	Status
{DEAL,GOVERNMENT}	2	Y
{DEMOCRACY,GOVERNMENT}	2	Y
{CPI,GOVERNMENT}	3	Y
{CPI,123}	2	Y
{CPI,AGREEMENT}	2	Y
{GOVERNMENT,123}	2	Y
{GOVERNMENT,AGREEMENT}	2	Y
{123,AGREEMENT}	2	Y

In the same way we apply the above process till we get one itemset,

Table 3.5 Frequent Itemsets at 3rd-level

F_3

Itemset	Count	Status
{CPI,GOVERNMENT,123}	2	Y
{CPI,GOVERNMENT,AGREEMENT}	2	Y
{CPI,123,AGREEMENT}	2	Y
{GOVERNMENT,123,AGREEMENT}	2	Y

Similarly

Table 3.6 Frequent Itemsets at 4th-level

F_4

Itemset	Count	Status
{CPI,GOVERNMENT,123,AGREEMENT}	2	Y

As the end of algorithm we get the final frequent Itemset. Now integrate all the frequent Itemsets at each level and store it in database.

3.3.7 Application of Decision Tree.

After Getting All the frequent Itemset of every page, Merge them in one set and store it in database.

{Itemset$_1^1$, Itemset$_2^1$, Itemset$_3^1$ …….. Itemset$_L^C$ ……}

Where Itemset$_c$ [I] indicates the I-th-frequent Itemset extracted from the sets of transactions labeled class$_c$. These attributes are numbered as Attribute$_1$, Attribute$_2$, . . .,Attribute$_I$,. . . in a sequential order. Then, the sub-data D_c composed of transactions labeled class$_c$ is constructed as depicted in Table 1 where every flag$^c_{mn}$ is represented as

$$flag^c_{mn} = \begin{cases} 1 : \forall Itemset^c_n \subset page^c_m \\ 0 : \text{the others} \end{cases}$$

This serves to predict the class of new examples by verifying whether specific nouns exist in the document of the Web-page or not. We repeat this procedure across all classes, and integrate Dc of every class into a whole data Data = {D_1, . . . ,D_C} where C is the number of classes.

Table 3.7 Set of all Itemsets

	Attribute$_1$	**Attribute$_N$**	**Class**
Page$_1^C$	**Flag$^C_{11}$**	**….**	**Class$_C$**
…..	**….**	**…..**	**Class$_C$**
Page^C_M	**….**	**Flag$^C_{MN}$**	**Class$_C$**

Once $Data_C$ is obtained for each class, a decision tree learning technique C4.5 [5] is applied for the classification of Web-pages. Decision tree algorithms begin with a set of examples and create a tree data structure that can be used to classify new examples. Each node of a decision tree contains a test, the result of which is used to decide which branch to follow from that node. The leaf nodes contain class labels instead of tests. When a test example reach a leaf node, the decision tree classifies it using the label stored there. A decision tree is inferred by growing it from the root downward and greedily selecting the next best attribute for each new branch added to the tree. C4.5 uses a statistical criterion called gain ratio to evaluate the "goodness" of a test.

Once the tree is obtained, C4.5 algorithm [3] applies n-fold cross-validation to evaluate the error rate of the tree. This method divides all examples into n subsets of approximately equal size. Each time one of the n subsets is used as a set of testing examples and the other n−1 subsets are put together to form a set of training examples. The same trial is repeated n times. These n trials present the average error rate properly, and robust evaluation of the learned decision trees comparatively, although we specifically need to generate attributes and evaluate the accuracy of decision trees respectively after dividing all examples into n subsets.

Decision trees constructed by C4.5 algorithm can provide a set of comprehensive rules to classify new examples as described later. These rules are clearly described in the form of tests and the results derived from them. This is an advantage of our approach which uses rule-based inductive classification.

3.4 Dataflow Diagram

Dataflow diagram display the sequence of the data flown across various module of the system. In this project there are Four module.

- Meta-Extractor.
- Itemset-Creator.
- Basket-analyzer.
- Decision Tree.

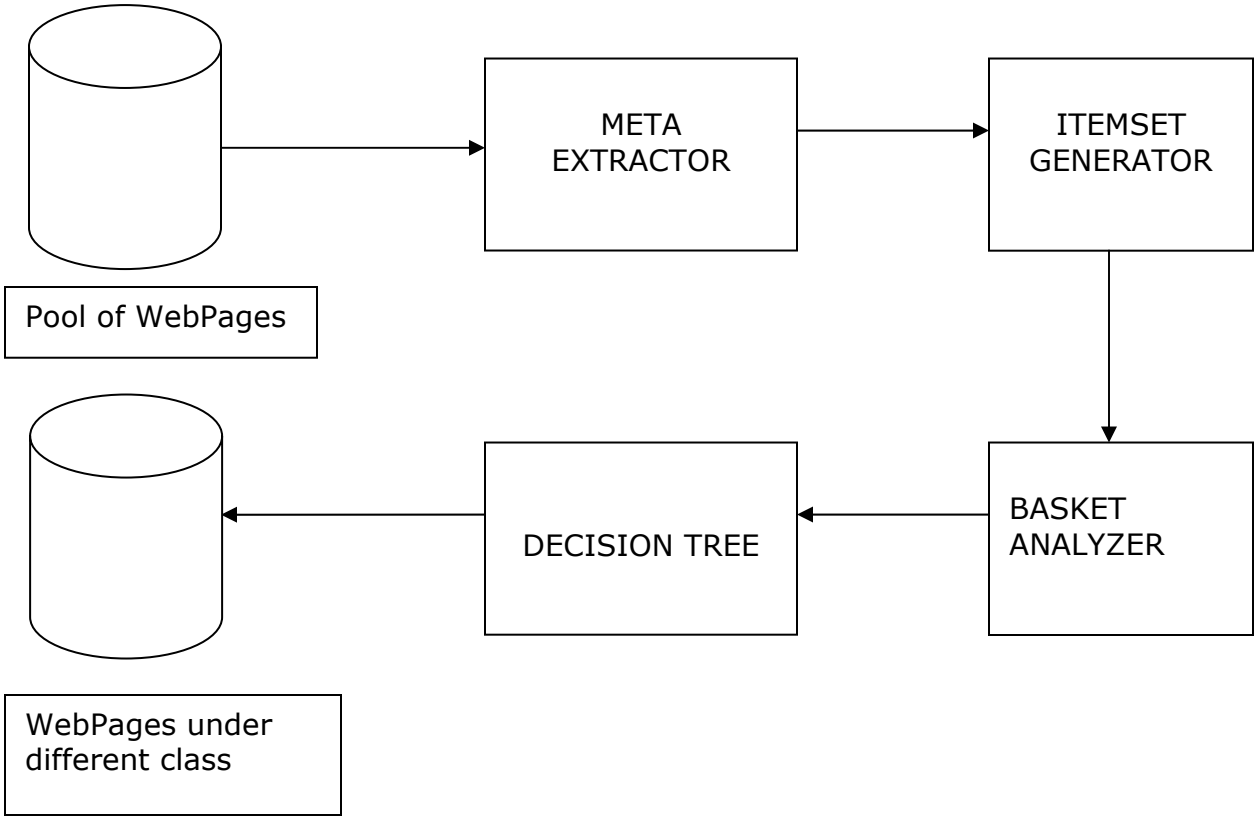


Figure 3.2 Context Diagram for proposed System (PWS).

3.5 E-R Diagram

In the Personalize Web classifier following entity has been used.

I. Entity : Page_Master
   a. Page_Id
   b. Page_Name
   c. Page_Path
   d. Class_Id
   e. Status

II. Entity : Meta_Data
   a. Page_Id
   b. Key_Id
   c. Keyword

III. Entity : Transaction_Master
   a. Page_Id
   b. Transaction _Id
   c. Item

IV. Entity : Itemset
   a. Page_Id
   b. Level
   c. KeySet_Id
   d. Counter
   e. Status

V. Entity : Class_Master
   a. Class_Id
   b. Class_Name
   c. HasChild

VI. Entity : Child_Master
   a. Child_Id
   b. Child_Name
   c. Class_Id
   d. Status

VII. Entity : Frequent_Itemsets
   a. Is_Id
   b. Is_Name
   c. Class_Id
   d. Level
   e. Subclass_Id

Based upon the above attribute we will draw the E-R. Diagram.

3.6 Use Case Diagram

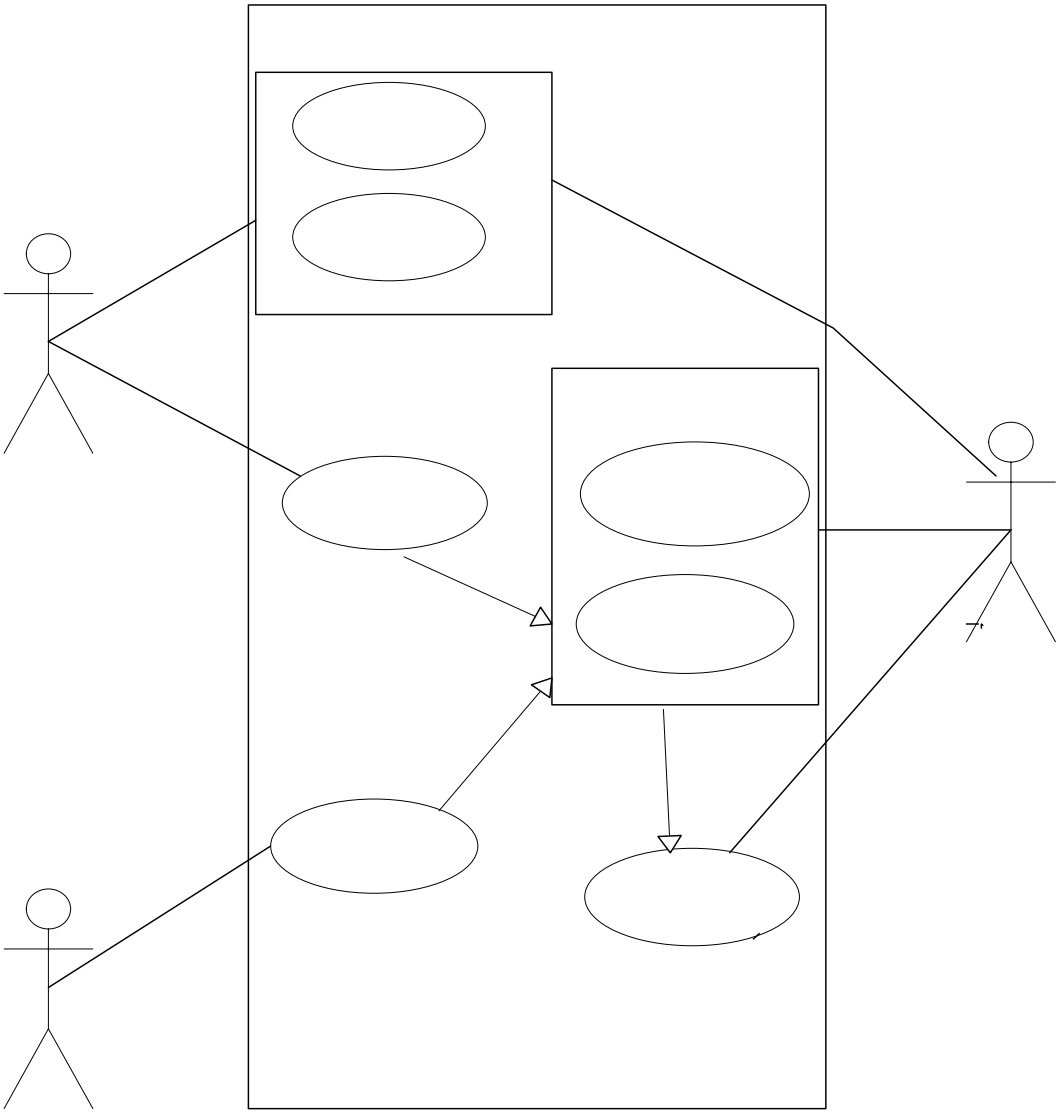


Figure 3.4 Use Case diagram for PWS

3.6 Sequence Diagram

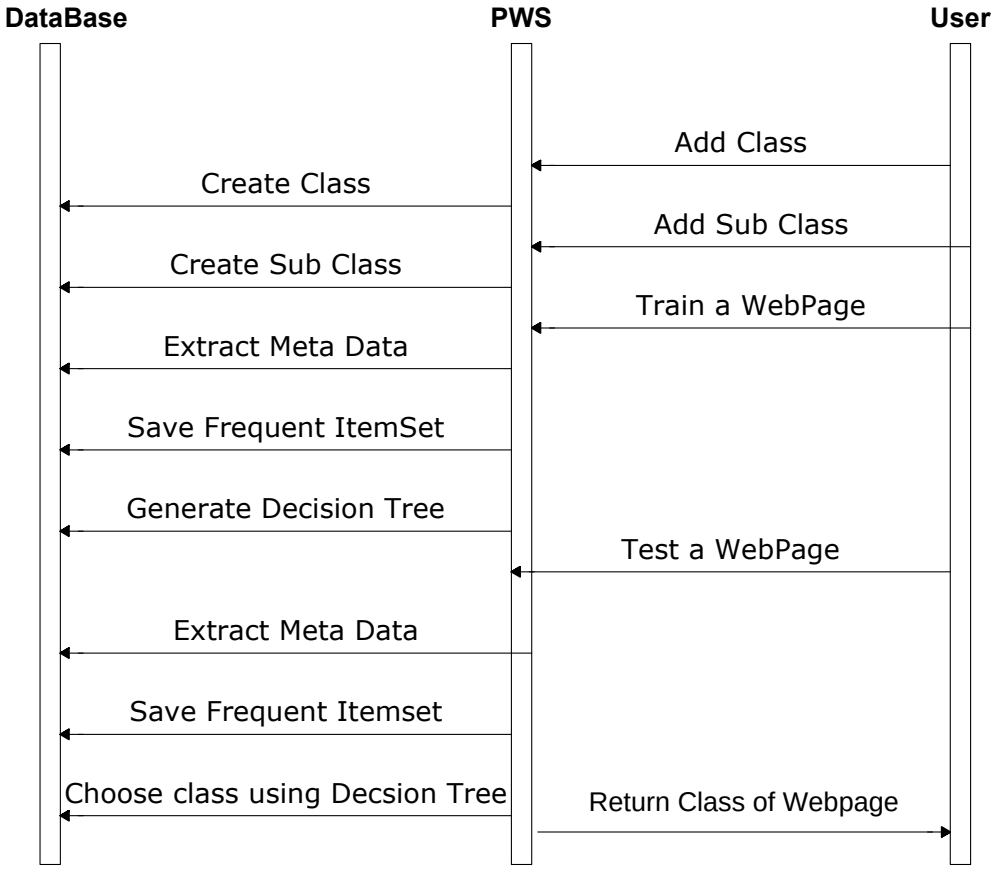


Figure 3.5 Sequence diagram for PWS

3.8 Database Design

Table: Page_Master

Field Name	Data Type	Constraint	Description
Page_Id	Varchar2(10)	Primary Key	Identify the page uniquely
Page_Name	Varchar2(50)	Not null	Name of file
Page_Path	Varchar2(50)	Not null	Path in local drive
Class_Id	Varchar2(10)	Foreign key	Reference to Class_Master

Table: Meta_Data

Field Name	Data Type	Constraint	Description
Page_Id	Varchar2(10)	foreign Key	Reference to Page_Master
Key_Id	Varchar2(25)	Not null	
Keyword	Varchar2(50	Not null	

Table: Transaction

Field Name	Data Type	Constraint	Description
Page_Id	Varchar2(10)	foreign Key	Reference to Page_Master
Tran_Id	Varchar2(10)	Not null	
Key_Id	Varchar2(25)	Foreign key	Reference to Meta

Table: Itemsets

Field Name	Data Type	Constraint	Description
Page_Id	Varchar2(10)	foreign Key	Reference to Page_Master
Key_Set_Id	Varchar2(1000)	Not null	
Level	Number(1)	Not null	
Status	Varchar2(1)	Not null	

Table: Ig (Ignore word)

Field Name	Data Type	Constraint	Description
IG_Id	Varchar2(10)	Primary Key	Identify the page uniquely
Ig_Name	Varchar2(25)	Not null	Name off word

Table: Frequent_Itemsets

Field Name	Data Type	Constraint	Description
IS_Id	Varchar2(10)	Primary Key	Identify the page uniquely
IS_Name	Varchar2(1000)	Not null	
Level	Number(1)	Not null	
Class_Id	Varchar2(10)	Foreign key	reference to Class_Master

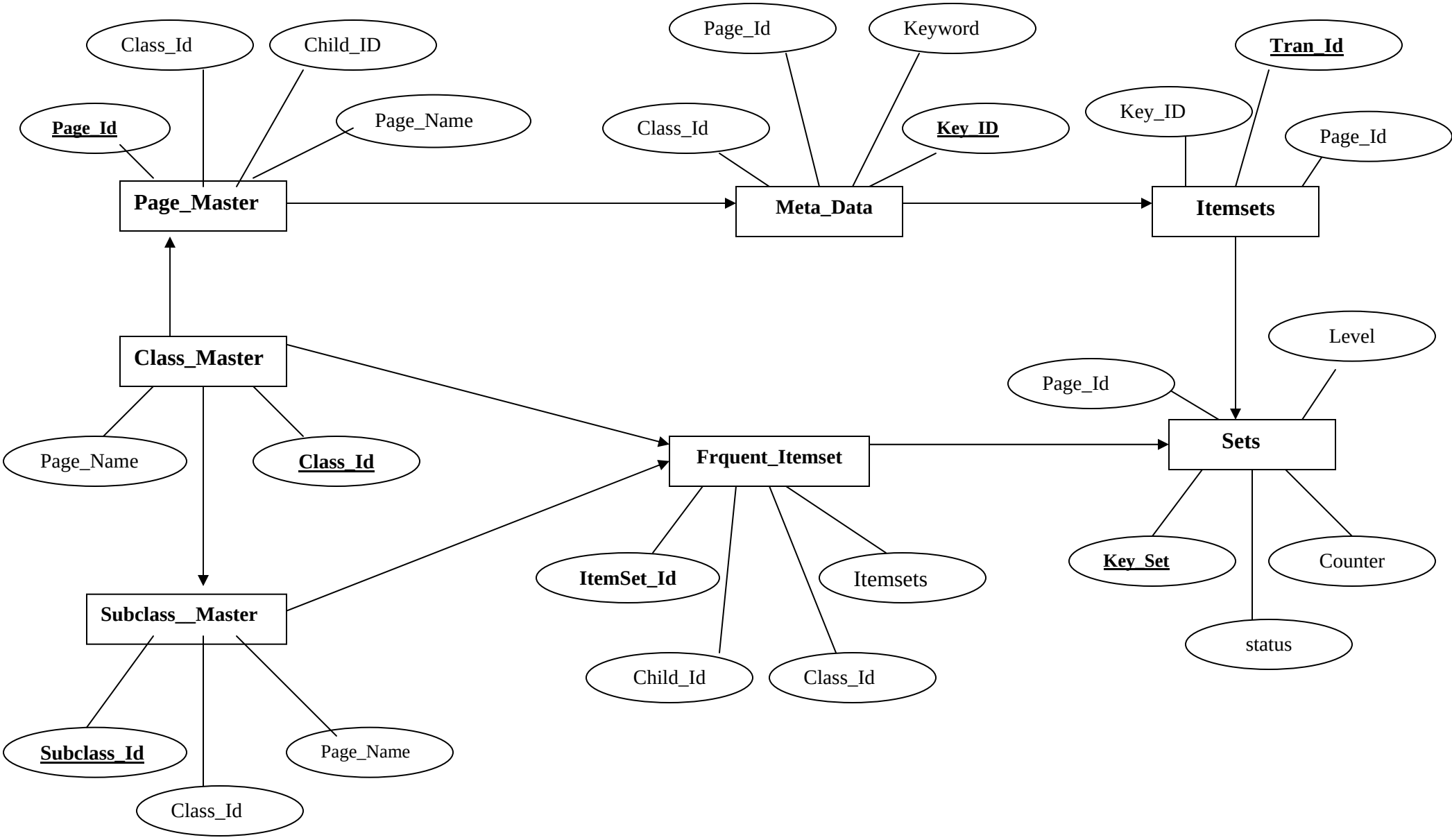
Table: Class_Master

Field Name	Data Type	Constraint	Description
Class_Id	Varchar2(10)	Primary Key	Identify the page uniquely
Class_Name	Varchar2(25)	Not null	
Has_Child	Boolean	Not null	

Table: SubClass_Master

Field Name	Data Type	Constraint	Description
Subclass_Id	Varchar2(10)	Primary Key	Identify the page uniquely
Subclass_Name	Varchar2(25)	Not null	
Class_Id	Varchar2(10)	Foreign key	

Fig 3.3 E-R Diagram for PWS



4. IMPLEMENTATION AND RESULTS

As discuss in the previous Chapter of Design, the project is divided in four parts (module).

I. Extraction of Meta Data from the Web-Page.
II. Finding out Itemsets from the page
III. Application of Apriori algorithm to find out frequent Itemsets
IV. Application Of decision tree to generate class

4.1 Tools.

To implement the system, following tools have been used.

I. Platform : Windows XP
II. Environment : Microsoft.Net Framework 2005
III. Language: Microsoft Visual C#. 2005
IV. Database: SQL Server 2005.

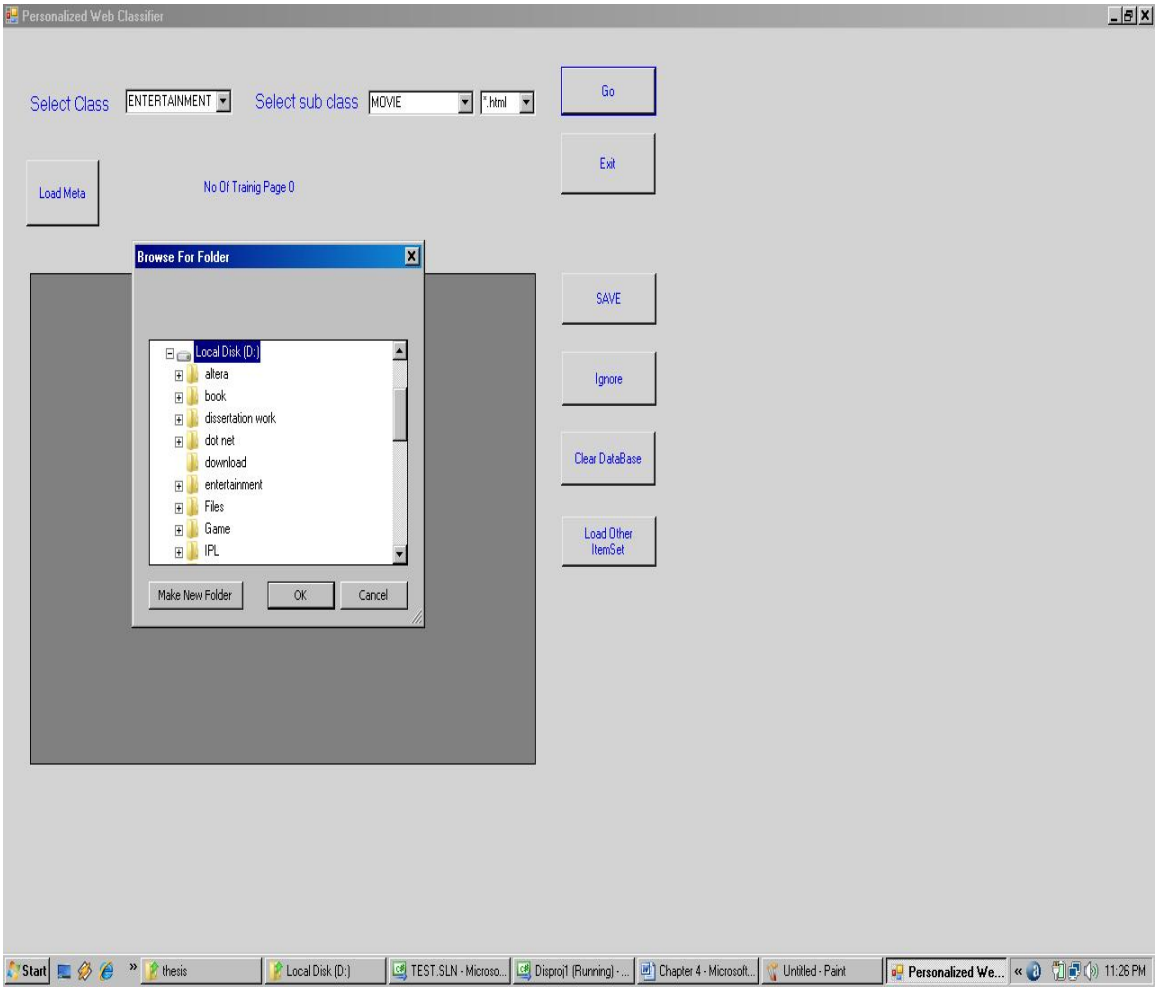
4.2 Implementation

The Personalized Web Classifier (PWS) classifies webpage according to the classes specified by user. Before using it directly, we have to train it according to our requirement. This Software is basically used in two modes.

- Train Mode
- Test Mode

4.2.1 Train mode

Initially we provide the pages that are going to train by the system. In this project 250 pages are downloaded from each category. Folder-Browser control used to browse the folder of local drive. Before selecting the Pages, choose class and subclass of the pages that are going to train. Also select the type of the page. Screenshot of this action is shown in figure 4.1.



4.1 Selection Of train Page

After selecting the pages our job is to find out Meta Data of the web page. To retrieve the metadata following class and function used.

Class: MetaFetcher

Function:

I. GetTitle (): This function will return the title of the web page in terms of array.

II. GetMetaTag (): This function will return the <Meta> tag.

III. GetDescription (): This function will return the description of the web page.

IV. GetKeyword (): This function will return the keyword specifies by the webpage.

The above function will return all the Meta information of the web page. Data-View-Grid Control is used to display the meta-data. Now user has to decide which Meta data are not relevant for page. The screenshot of the above action is shown as below.

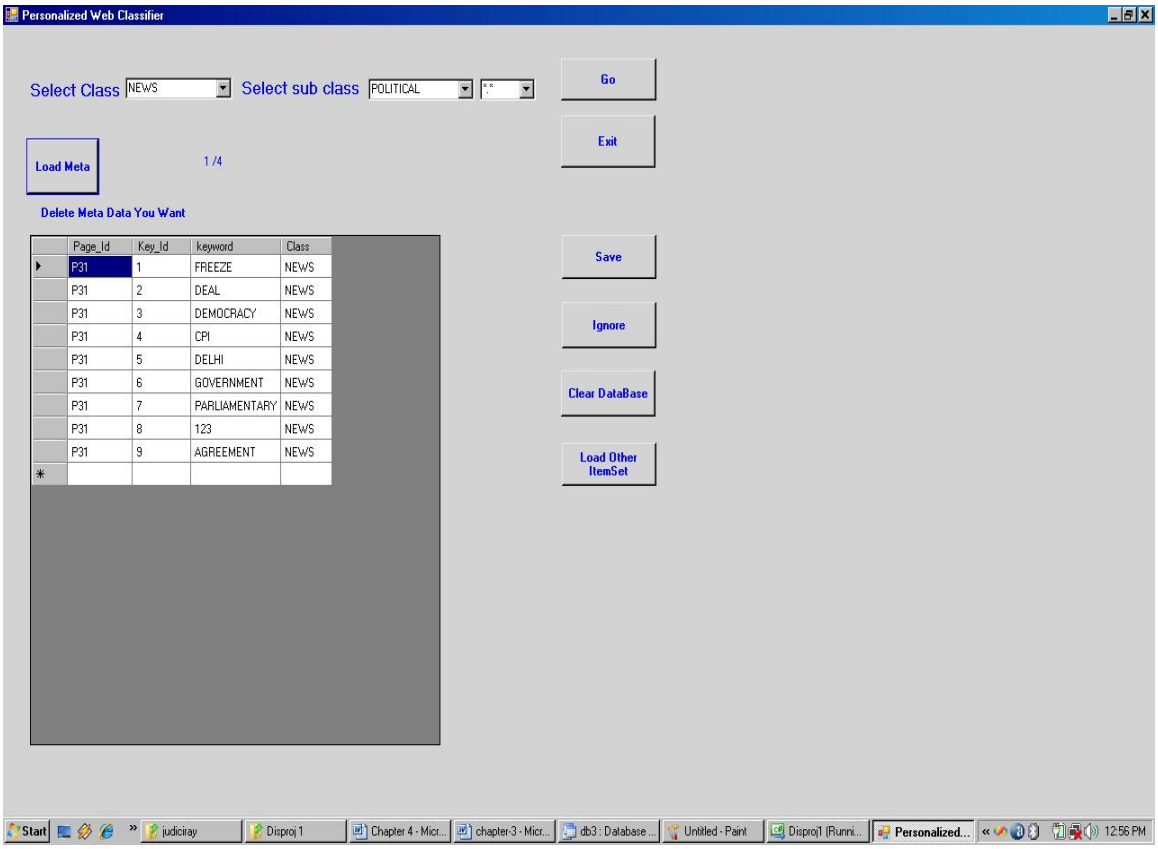


Fig 4.2 Selection of Metadata of the webpage.

After selecting Metadata, our next task is to find out Frequent Itemsets. So represent the each statement of webpage using metadata. In other word we can say that in each sentence of webpage find out the meta data and remove all word which are not the part of the Meta-Set (the set of all metadata). We will consider every sentence of webpage as the transaction of the metadata. to find out Itemset I have used following function.

Class: MetaFetcher

Function:

I. GetItemset (): This function will return all the Itemset from the webpage.
II. SaveItemset (): This function will save the itemset into database.

Now we will apply the Apriori algorithm to generate the frequent-Itemsets. We have discussed the detail description of Apriori algorithm in chapter 3. the following function used for Apriori algorithm

Class: Apriori

Function:

- Apriori(): This function will find the 1-Itemset (1-level itemset)
- Icount (): This function will find 2-Itemset……N-Itemset.
- Minsup (): This function will apply the minimum support on itemset. So the Itemset that appear below the minimum support are not frequent.
- FilterData (): This function will use to eliminate the duplication of Itemset.

Using the above function we get the frequent Itemsets that actually represent the meaning or class of the document (or web page). In this project, all the Itemsets that support minimum threshold are displayed in grid. So during the training, user has given choice to select Itemsets which can be used for to classify the unknown webpage.

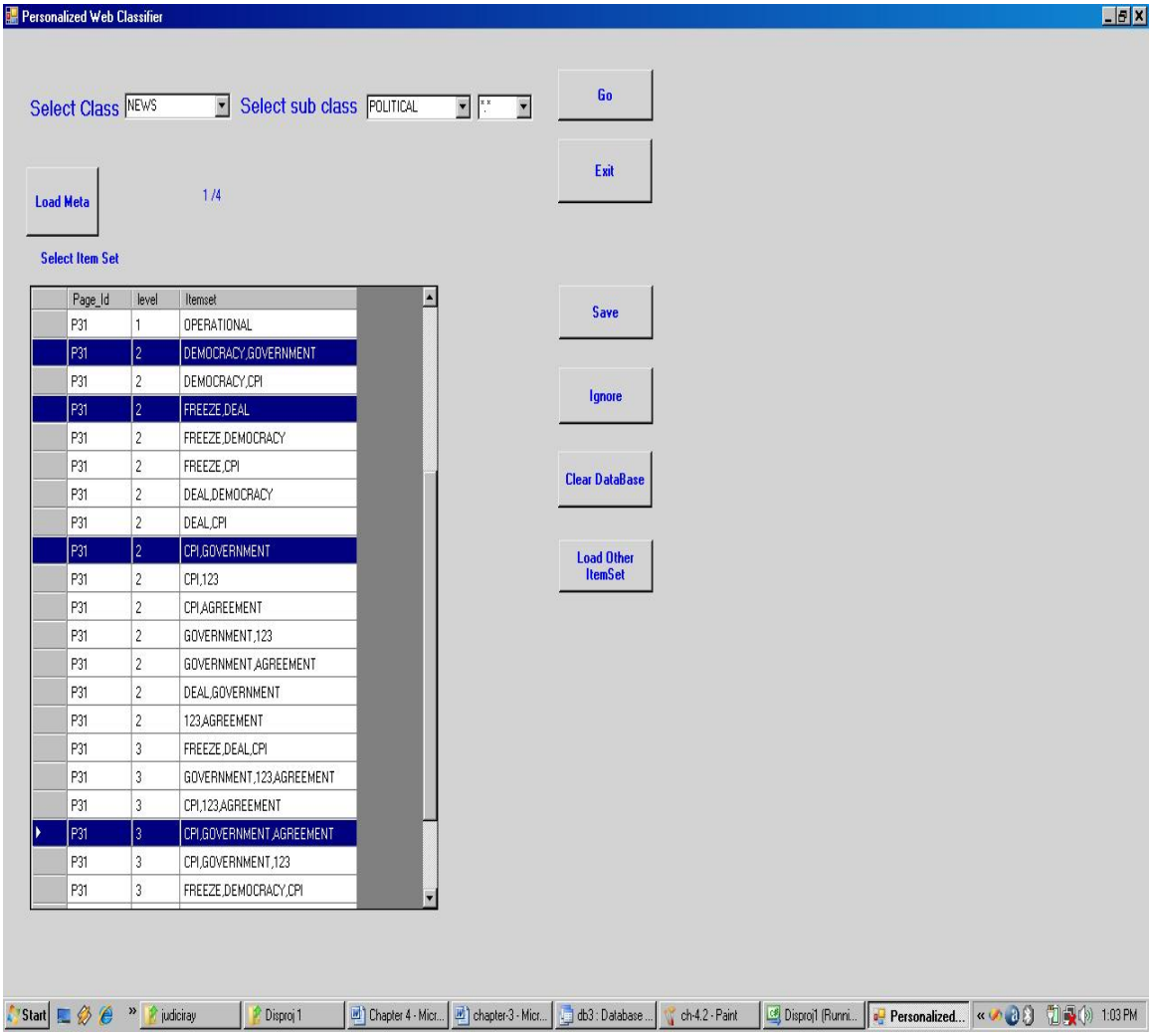


Fig 4.3 Selection of Frequent Itemsets

As shown in figure 4.3, user has to select frequent Itemsets. To achieve better accuracy, select the Itemsets manually. The selected Frequent Itemsets stored in Main Decision table to classify the WebPages in testing mode. Maximum level of itemset depends upon the content of the webpage. In some cases it goes up to 5, 6 depending upon the content.

The above entire process will execute to train the system by one page. One can train only one class of WebPages at one training sessions.

4.2.2 Test mode

After giving appropriate training, next phase would be testing. Here 200 pages of each class used to train the system. Initially user has to provide the pages that are going to train by the system. As discussed earlier, Folder-Browser control will used to select the WebPages which are being tested.

In test mode the following phase will be carried out without user interaction:

I. Metadata Extraction
II. Generation Of Itemsets
III. Generation of frequent Itemsets using Apriori algorithm

The above phase is same as are in train mode. Now decision tree is constructed using C4.5 algorithm to classify the web pages. Here binary classification is used to classify the WebPages. C4.5 algorithm is explained in previous chapter.

In this mode, set of unknown pages is given as input to the system. Rest of the entire sub-phases carried out by system automatically without user interaction. Data-Gridview control is used to display the result of the classification.

Apart from this, logs all the pages that are used for train or testing purpose is kept to avoid the duplication of WebPages during the training or testing. In other word one can say that a page can not be trained or tested more than one times. Path of local directory and file name is used to avoid the duplication.

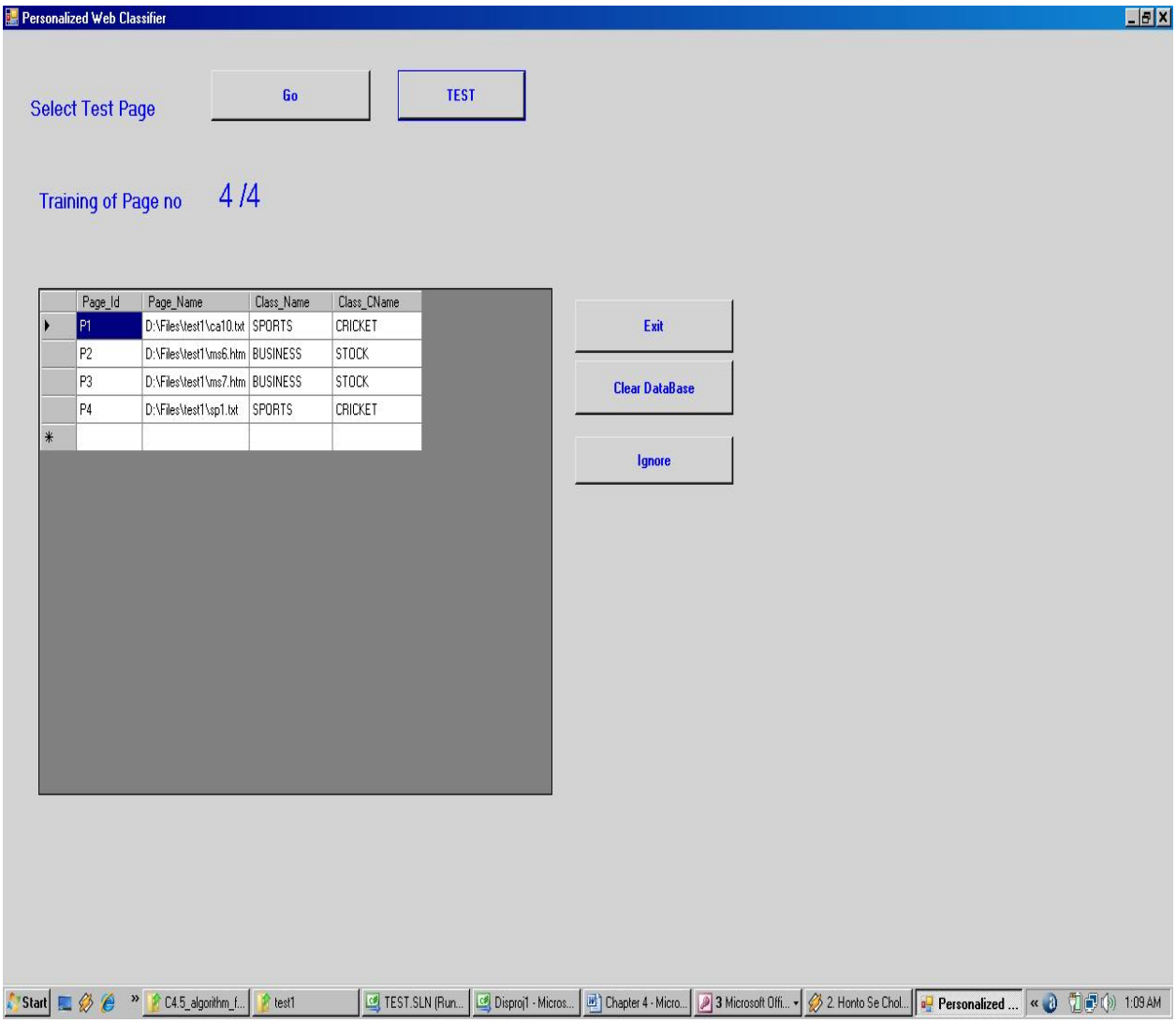


Fig 4.4 Result of Classification

Fig 4.4 displays the result of the classification done by the PWS (Personalized Web classifier).

Function

The following function is used for classification

I. Createtree (): this function will create decision tree.
II. Decide(): this function will perform the test at each level
III. FillGrid():This function will used to fill up the grid.

4.2.3 Namespaces

In .Net framework classes are encapsulated in the namespaces. Following namespaces have been used for PWS.

I. System
II. System.Collections.Generic
III. System.ComponentModel
IV. System.Data
V. System.Collections
VI. System.Drawing
VII. System.Text
VIII. System.Windows.Forms
IX. System.Data.OleDb
X. System.IO

4.3 Performance Measure

The following 4 terms are used for performance measure quantity.

I. TP (True Positive): the number of webpage correctly classified to that class.
II. TN (True Negative): the number of webpage correctly rejected from that class.
III. FP (False Positive): the number of webpage incorrectly rejected from that class.
IV. FN (False Negative): the number of webpage incorrectly classified to that class.

The most important performance parameter Error rate can be find out as below formula:

$$Error\ rate = \frac{the\ number\ of\ all\ testing\ examples\ classified\ erroneously}{the\ number\ of\ all\ testing\ examples} \quad \text{.....(1)}$$

The other important peformance parameter can be define as follow

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad \text{.....(2)}$$

$$precision = \frac{TP}{TP + FP} \quad \text{.....(3)}$$

$$\text{Re}\,call = \frac{TP}{TP + FN} \quad \text{.....(4)}$$

The result of the system will be discussed in the result section.

When distributions are highly skewed, as those in text categorization often are, using Accuracy or Error rate= 1 − Accuracy may be inappropriate. Instead, Recall, Precision, and F1 measure are commonly used as classification performance measures. There is a trade off relationship between Precision and Recall.

The above performance parameter heaveily used Information Retreival area(IR area).

4.4 Experimental Result

In this phase, 100 pages have been tested in 10 pages group. In testing Minsup (minimum support) for frequent Itemset is kept variable. Here pages are tested on diiferent minsup value like 15%, 20%, 30%.

$$\text{Minsup}(\%) = \frac{\text{No of occurrence of Itemset in webpage}}{\text{No of senetences in Webpage}}$$

Results are shown in below table. It will heavily depend upon the content of webpage. Table 4.1 and 4.2 display the result in terms of accuracy,precision, recall.

Minsup	Sports			Business		
(%)	Accuracy	Precision	Recall	Accuracy	Precision	Recall
15	94.54	93.33	96.55	92.72	88	95.65
20	92.72	90	96.42	87.27	86.95	90.9
25	89.1	86.66	92.85	85.45	82.6	90.47

Table 4.1 Results of sports and business class (%).

Minsup	News			Entertainment		
(%)	Accuracy	Precision	Recall	Accuracy	Precision	Recall
15	82.22	76	90.47	86.66	80	88.9
20	75.55	68	85	82.22	75	83.3
25	68.88	60	78.94	77.77	70	77.4

Table 4.2 Results of News and Entertainment class(%).

As shown in above table we found that value of precision is less than recall and value of the Recall is higher. So one can conclude that wrong classification is very less. As observing value of precision in news and Entertainment class is low. The reason for this low precision is that news has vast domain. So we get lower precision value.

After examining classification of all four class, We conclude that as decrease minsup (minimum support for selecting frequent Itemsets), accuracy increase. Apart from this, time required to classify one page is also increased. This is the trad-off one has to make.

In table 4.3 the list of itemsets is displayed which will be used to classify the class.

Table 4.3 common itemsets of the different class.

Sports	Entertainment	Business
{IPL,PLAY}	{BOLLYWOOD,GLAMOUR}	{SENSEX ,UP}
{PLAY,CRICKET}	{ACTOR,DEBUT}	{NIFTY,DOWN}
{ICL,PLAY}	{FILM,DIRECTOR}	{COMPANY,JOINT,VENTURE}
{BOWLING}	{SET,HOLLYWOOD}	{COMPANY,TAKE, OVER}
{GOAL,FIRST}	{MUSIC,REALEASE}	{SHORT, SELLING}
{ICC,ANTI,CURRUPTION}	{ALBUM,SONG}	{RUPPES,INVESTMENT}
{EPL,RONALDO}	{HIT,TRACK}	{HIGHER,INFLATION}
{REAL,MADRID}	{MUSIC,DIRECTOR}	{BUY,STOCKS}
{PLAY,SOCCER}	{FILM,FLOP}	{PORTFOLIO,MANAGER}
{MANCHESTER,UNITED}	{BOX,OFFICE}	{GDP,GROWTH}
{ICC,SHOAIB}	{SUPER,HIT,FILM}	{F&O}

From above table 4.1 and 4.2 one can easily show the result using graph.

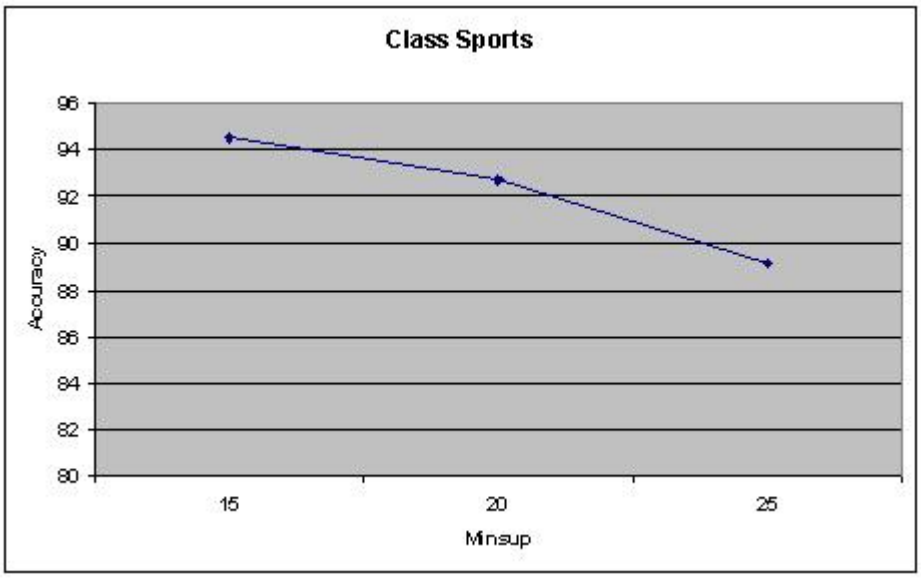


Fig 4.5 Accuracy vs Minsup in Sports class

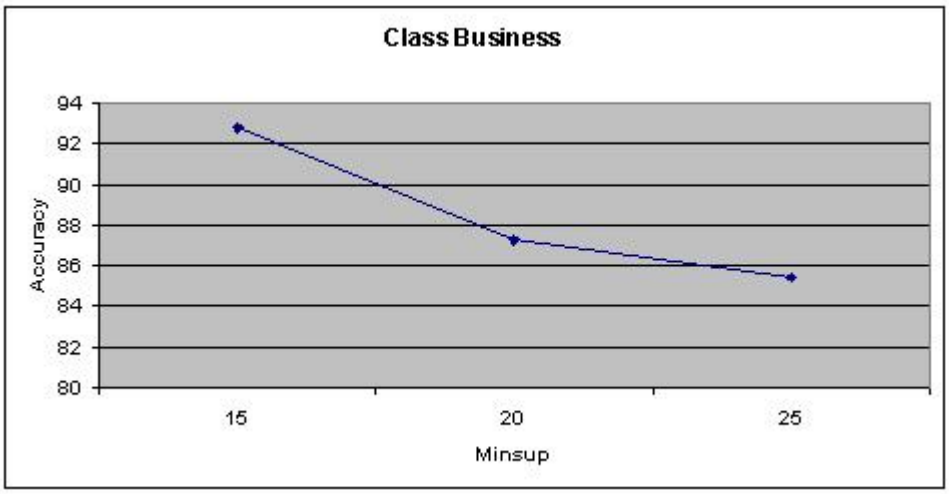


Fig 4.6 Accuracy vs Minsup in Business class

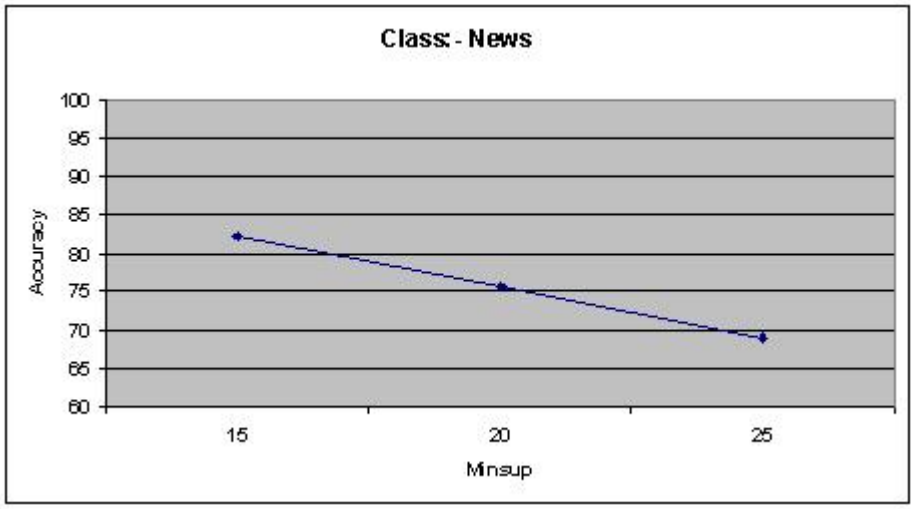


Fig 4.7 Accuracy vs Minsup in News class

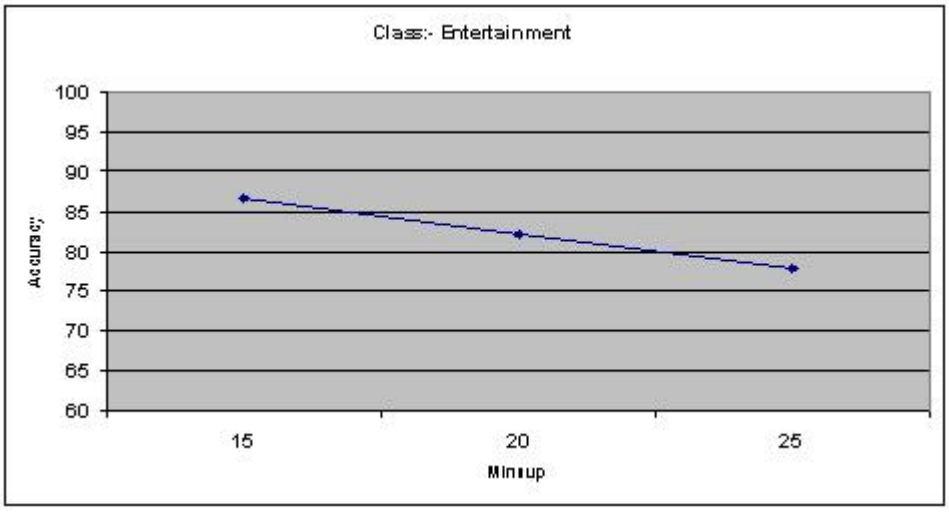


Fig 4.8 Accuracy vs Minsup in Entertainment class

As shown in above fig. 4.5 to 4.8 accuracy is derease as increase the minsup. In sports highest accuracy one can find, while in news accuracy is lower comapre to other class. Now combining above four graph in one bar graph shown in next figure.

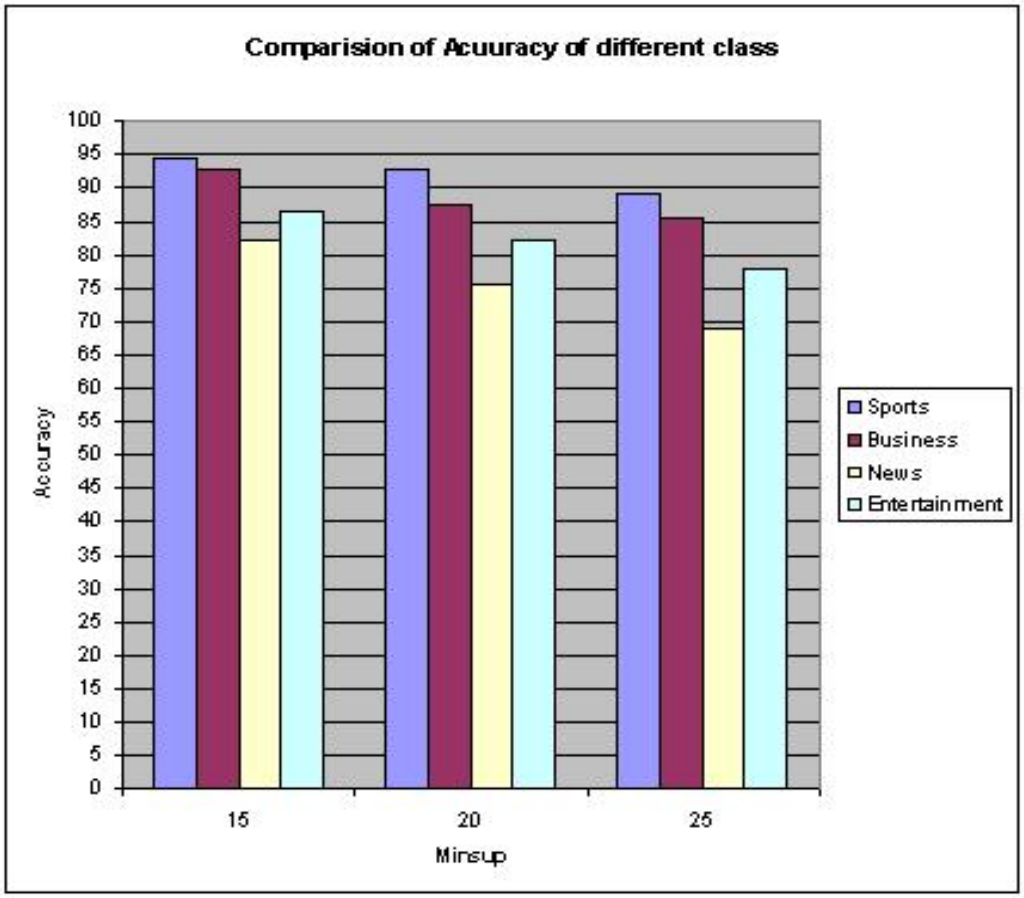


Fig 4.8 Comparision of Accuracy vs Minsup in different classes

5. Conclusion

5.1 Concluding Remarks

After reviewing web classification research with respect to its features and algorithms, one can conclude this report by summarizing the lessons that have learned from existing research and pointing out future opportunities in web classification.

Web page classification is a type of supervised learning problem that aims to categorize web pages into a set of predefined categories based on labeled training data. Classification tasks include assigning documents on the basis of subject, function, sentiment, genre, and more. Unlike more general text classification, web page classification methods can take advantage of the semi-structured content and connections to other pages within the Web.

Future web classification efforts will certainly combine content and link information in some form.

5.2 Suggestion for Webpage Architecture

In Webpage design mechanism, <Body> tag is used for specifying the content of webpage. Nowadays Webpage become so much noisy. Nowadays Internet becomes the new media to for the marketing of the product. Due to this, it contains many objects like images, banner, ads, logo which may be irrelevant with respect to main content. So, it is very difficult to extract the main content of the web page.

So in web page design, if one can define one new tag which defines the Main content of webpage, then main content can be easily retrieved. Also efficiency of the PWS would be increased.

5.2 Future Work

I. Apply Ant-Colony Optimization method and compare the results with current method.

II. Implement the Page relevancy checking in Google search engine's result.

III. Design Firewall using PWS.

REFERENCES

GENERAL

Literature from various books and research papers has been carried out to support present work. The literature is summarized as below.

BOOKS

[1]. Chakrabarti, S. (2003). Mining the Web: Discovering Knowledge from Hypertext Data. San Francisco, CA: Morgan Kaufmann

[2]. Data Mining Techniques by P.M.Gupta and N.krishnamurthy

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

PAPERS

[1]. Toshiko Wakaki, Hiroyuki Itakura: “Rough Set-Aided Feature Selection for Automatic Web-Page classification”.

[2]. R.Agrwal and R.Srikant: First algorithms for mining association rules. In Proceedings of the 20th VLDB Conference, pp.487-499, 1994.

[3]. Daniele Riboni: Feature Selection for Web Page Classification.

[4]. Xiaoguang Qi and Brian D. Davison : Web Page Classification: Features and Algorithms

[5]. Makoto Tsukada, Takashi Washio : Automatic Web-Page Classification

[6]. Raymond Kosala , Hendrik Blockeel : Web Mining Research: A Survey, ACM SIGKDD Explorations Newsletter archive Volume 2, Issue 1(JUNE 2000)

[7]. Jaroslav Pokorny , Jozef Smizansky : Page Content Rank: An Approach to Web Content Mining.