

*Republic of Iraq
Ministry of Higher Education
and Scientific Research
Al-Nahrain University
College of Science*



Web Text Mining Using Fuzzy Logic

*A Thesis
Submitted to the College of Science, Al-Nahrain University
In Partial Fulfillment of the Requirements for
The Degree of Master of Science in Computer Science*

By

Huda Abdul Mahdi Taleb

(B.Sc. 2005)

Supervisor

Dr.Sawsan K. Thamer

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

قَالُوا سُبْحَانَكَ لَا عِلْمَ لَنَا إِلَّا مَا عَلَّمْتَنَا إِنَّكَ
أَنْتَ الْعَلِيمُ الْحَكِيمُ

صدق الله العلي العظيم

سورة البقره الاية (32)

Supervisor Certification

I certify that this thesis was prepared under our supervision at the Department of Computer Science/College of Science/Al-Nahrain University, by **Huda Abdul Mahdi Taleb** as partial fulfillment of the requirements for the degree of Master of Science in Computer Science.

Signature:

Name : **Dr. Sawsan K. Thamer**

Title : **Lecturer**

Date : / / **2008**

In view of the available recommendations, I forward this thesis for debate by the examination committee.

Signature:

Name : **Dr. Taha S. Bashaga**

Title : **Head of the department of Computer Science,
Al-Nahrain University.**

Date : / / **2008**

Certification of the Examination Committee

We chairman and members of the examination committee, certify that we have studied this thesis "**Web Text Mining Using Fuzzy Logic**" presented by the student **Huda Abdul Mahdi Taleb** and examined her in its contents and that we have found it worthy to be accepted for the degree of Master of Science in Computer Science.

Signature:

Name : **Dr. Abdul Monem S. Rahma**

Title : **Assist. Prof.**

Date : / /2009

(Chairman)

Signature:

Name: **Dr. Ban N. Dhannoon**

Title : **Assist. Prof.**

Date : / /2009

(Member)

Signature:

Name: **Dr. Jamal Fadil Tawfeeq**

Title : **Lecturer**

Date : / /2009

(Member)

Signature:

Name: **Dr. Sawsan K. Thamer**

Title : **Lecturer**

Date : / /2009

(Supervisor)

Approved by the Dean of the Collage of Science, Al-Nahrain University.

Signature:

Name: **Dr. LAITH ABDUL AZIZ AL-ANI**

Title : **Assist. Prof.**

Date : / /2009

(Dean of Collage of Science)

Dedication

I dedicate my work to ...

My Parents ...

Sisters

Brothers ...

Friends

And

Every one who help and support

me

Huda

Acknowledgement

First of all great thanks are due to Allah who helped me and gave me the ability to achieve this research from first to last step.

I would like to express my sincere appreciation to my supervisor Dr. Sawzan Kamal for guidance, assistance and encouragement during the course of this project.

Grateful thanks for the Head of Department of Computer Science of Al-Mahrain University Dr. Taha S. Bashaga for the continuous support during the period of my studies.

Deep gratitude and special thanks to my family: my parents, sisters and brothers for their encouragements and supporting to succeed in doing this work.

Special thanks to my faithful friends for supporting and giving me advises.

Huda

Abstract

With the explosive growth of the amount of content on the Internet, it has become increasingly difficult for users to find and utilize information and for content providers to classify and catalog documents. Traditional web search engines often return hundreds or thousands of results for a search, which is time consuming for users to browse therefore the searching Web pages similarity is using.

The proposed system (Web Pages Fuzzy Similarity) consists of two phases: Off-line and On-line phases. The Off-line phase constructs Documents Vector DB while On-line phase constructs a Query Document and then gives similar pages to it. Every document should be passed through a set of operations to extract the information that represent it. These operations are: Lexical Text Analyzer, Elimination of Stop Words and other unused words, HTML Document Ranking (HDR) method, weights computation of words by using formula depending on the words frequency and the words attributes (such as font style, font size, position of the words, link text, title, header), and then the documents vector DB is constructed from the largest weights of the document's words.

The On-line phase consists of two steps. The first one takes a query and constructs document vector for it. The second step computes the similarity between query document and documents stored in DB. The similarity measure is done using two methods. The first one is Cosine Similarity Measure and second one is a new suggested formula named Fuzzy Web mining logic equation. Using fuzzy logic enhances and improves the results of search by extracting the most related pages which could be extracted by normal method but with lower relationship.

Table of Contents

Abstract	I
Table of Contents	II
List of Abbreviations	V
Chapter One: Introductions	
1.1 Introduction	1
1.2 The World Wide Web.....	2
1.3 Web Mining.....	2
1.4 Fuzzy Web Mining	4
1.5 Literature Survey	5
1.6 Aim of thesis.....	7
1.7 Thesis outlines.....	8
Chapter Two: Web Mining and Fuzzy Logic	
2.1 Introduction.....	9
2.2 Data Mining.....	9
2.3 Data Mining Systems Classification.....	11
2.4 Kinds of Data that can be Mined.....	12
2.5 Data Mining Tasks.....	14
2.5.1 Classification.....	14
2.5.2 Clustering.....	15
2.5.3 Association Rule.....	16
2.5.4 Neural Networks.....	16
2.5.5 Genetic Algorithms.....	16
2.6 Web characteristics.....	17
2.7 Web Mining	18
2.8 Taxonomy of Web Mining.....	21
2.8.1 Web Usage Mining.....	21
2.8.2 Web Structure Mining.....	23
2.8.3 Web Content Mining.....	24
2.8.3.1 Web Page Content Mining.....	26
2.8.3.2 Web Search Result Mining.....	26

2.8.3.3 Web Content Mining Approaches.....	27
2.9 Information Retrieval	32
2.9.1 Information Retrieval Models.....	35
2.9.1.1. Boolean Model.....	35
2.9.1.2. Vector Space Model	36
2.9.1.3 Statistical Language Model.....	38
2.9.2 Document similarity.....	38
2.10 Benefits of Web Mining.....	39
2.11 Fuzzy Logic	40
2.12 Fuzzy sets.....	41
2.13 Fuzzy Information Retrieval.....	42
2.14 Fuzzy Numbers.....	42

Chapter Three: Web Pages Fuzzy Similarity System Implementation

3.1 Introduction.....	46
3.2 Web Pages Fuzzy Similarity Module	48
3.3 HTML Features Analysis.....	49
3.4 Methodologies of System	51
3.4.1 Data Collection.....	51
3.4.2 Document Preprocessing.....	53
3.4.3 Documents Ranking	61
3.4.3.1 HTML Document feature Ranking	62
3.4.3.2 Html Document Ranking (HDR) Method	64
3.4.4 Term Indexing.....	66
3.4.4.1 Term Weighting Measure	66
3.4.4.2. Constructing Document Vectors	69
3.4.5 Similarity.....	71
3.4.5.1. Construction of Query Document Vector.....	71
3.4.5.2. Normal Similarity	72
3.4.5.3 Fuzzy Similarity	73

Chapter Four: Explanation of Web Pages Fuzzy Similarity

System and Test Results

4.1 Introductions	75
4.2 HTML Document Collections.....	76
4.3 Document Preprocessing Experiments.....	79
4.3.1 Lexical Text Analyzer Experiment	79
4.3.2 Elimination of Stop Words and Useless Words	80

4.4	Document Feature Extraction Experiments.....	84
4.5	DataBase Construction.....	86
4.6	Search Similar Pages Experiments.....	86
Chapter Five: Conclusions and Suggestions for Future Works		
5.1	Conclusions.....	93
5.2	Suggestions for Future Works.....	94
	References	95
	Appendix A	
	Appendix B	

List of Abbreviations

ASCII	American Standard Code for Information Interchange
DB	Data Base
GA	Genetic Algorithms
HDR	Html Document Ranking
HTML	HyperText Markup Language
IR	Information Retrieval
KDD	Knowledge Discovery in Databases
NLP	Natural Language Processing
PDF	Portable Document Format
SGXML	Standard Generalized Markup Language
SQL	Structured Query Language
TFD	Term Frequency Document
TF-IDF	Term Frequency-Inverse Document Frequency
URL	Uniform Resource Locator
W3C	World Wide Web Consortium
WCM	Web Content Mining
WSM	Web Structure Mining
WUM	Web Usage Mining
WWW	World Wide Web
XML	eXtensible Markup Language



Chapter One

Introductions

Chapter One

Introductions

1.1 Introduction

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 order to find, extract, filter, and evaluate the desired information and resources. In addition, with the transformation of the Web into the primary tool for electronic commerce, it is imperative for organizations and companies, who have invested millions in Internet and Intranet technologies, to track and analyze user access patterns. These factors give rise to the necessity of creating server-side and client-side intelligent systems that can effectively mine for knowledge both across the Internet and in particular Web localities.

The Web is a medium for accessing a great variety of information stored in different parts of the world. The rapid expansion of the Web is causing the constant growth of this information, leading to several problems: an increased difficulty of finding relevant information, extracting potentially useful knowledge and learning about consumers or individual users. Web mining is an emerging research area focused with on resolving these problems [Coo97, Wan03].

1.2 The World Wide Web (WWW)

WWW is the most heterogeneous and dynamic repository available. A very large number of authors and publishers are continuously contributing to its growth and a massive number of users are accessing its resources daily. Data in the WWW is organized in inter-connected documents. These documents can be text, audio, raw data, and even applications. Conceptually, the WWW is comprised of three major components: the content of the Web, which encompasses documents available; the structure of the Web, which covers the hyperlinks and the relationships between documents; and the usage of the Web, describing how and when the resources are accessed. A fourth dimension can be added relating the dynamic nature or evaluation of the documents [Osm99].

These documents are created with a spatial language called HyperText Markup Language (HTML). This language allows the full use of hypermedia including text, images, graphics, sounds and other type of multimedia. Because HTML is a spatial language, it requires spatial software to access the Web. This type of access program is known as a Browser [sab02].

Data mining in the WWW, or Web mining, tries to address all these issues and is often divided into Web content mining, Web structure mining and Web usage mining [Osm99, Sha06].

1.3 Web mining

Web mining is the use of data mining techniques to automatically discover and extract information from Web documents and services [Coo97, Wan03, Wes05]. With the phenomenal growth of the Web, there is an ever-increasing

volume of data and information published in numerous Web pages. The research in Web mining aims to develop new techniques to effectively extract and mine useful knowledge/information from these Web pages. Due to the heterogeneity and the lack of structure of Web data, automated discovery of targeted or unexpected knowledge/information is a challenging task. It calls for novel methods that draw from a wide range of fields spanning data mining, machine learning, natural language processing, statistics, databases, and information retrieval [Bin05]. Web mining research can be classified into three categories [Wan03]:

1. Web Content Mining (WCM) refers to the discovery of useful information from Web contents, including text, image, audio, video, etc. Research in Web content mining encompasses resource discovery from the Web, document categorization and clustering, and information extraction from Web pages.
2. Web Structure Mining (WSM) studies the Web's hyperlink structure. It usually involves analysis of the in-links and out-links of a Web page, and it has been used for search engine result ranking.
3. Web Usage Mining (WUM) focuses on analyzing search logs or other activity logs to find interesting patterns.

Figure (1.1) shows the categories of Web mining [Wes05].

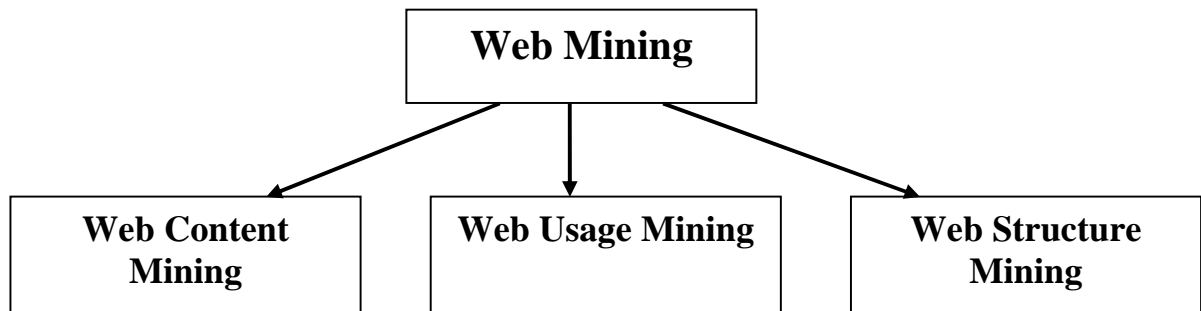


Figure (1.1): Web mining categories.

1.4 Fuzzy Web Mining

The role of fuzzy sets in Web mining holds promise mainly in: (i) document and user clustering, (ii) deduction and summarization, (iii) handling of fuzzy queries involving natural language and/or linguistic quantifiers like almost, about, and (iv) information fusion in multimedia data.

According to Zadeh, fuzzy logic may serve as the backbone of the Semantic Web, an extension of the current Web in which information is given well defined meaning, thereby better enabling computers and people to work in cooperation. Ordinary end-users often face difficulties in formulating a precise representation of their information needs in a Boolean query. This affects the efficiency of the information retrieval process. Hence Web search engines require the use of fuzzy aggregation operators. These are especially suitable in flexible query answering and information retrieval [Dra04].

1.5 Literature Survey

Some of other related works are listed bellow:

1. **“Fuzzy Data Mining for Querying and Retrieval of Research Archival Information” [Mic98].**

This paper proposed the design of FuzzyBase, an information /intelligent system to solve the need of literature search from a wide variety of sources. To facilitate intelligent and fast retrieval of information that is of interest to scientific research communalities with specific needs, such as getting RELEVANT technical information fast. It will involve the intelligent fuzzy retrieval of information, high-end computer and communalities infrastructure, retrieval algorithms.

2. **“An Algorithm for Clustering of Web Search Results” [Sta03].**

This thesis proposed a description-oriented algorithm for clustering of results obtained from Web search engines called LINGO. The key idea of algorithm is to first discover meaningful cluster labels and then, based on the labels; determine the actual content of the groups. It showed how the cluster label discovery can be accomplished with the use of the Latent Semantic Indexing technique. Several factors were also discussed which influence the quality of cluster description, such as input data preprocessing and language identification.

3. “An Efficient Algorithm for Fuzzy Web Mining” [Rui04]

This paper proposed a novel structure, the Frequent Link And Access Tree (FLAAT), and a corresponding efficient algorithm is designed by which mine can accurately frequent preferred paths that users are most interested in. To find more completely frequent fuzzy preferred. In addition, the duration time on Web page was characterized as fuzzy variable. The gained frequent fuzzy preferred paths with fuzzy expected values more accurately disclose the interest of users.

4. “Web Mining: Learning from the World Wide Web” [Jan04].

This paper discussed the use of unsupervised and supervised learning methods for user behavior modeling and content-based segmentation and classification of Web pages. The modeling is based on independent component analysis and hierarchal probabilistic clustering techniques. Text mining is used to categorize text according to topic, to spot new topics, and in broader sense to create more intelligent searches.

5. “Mining Web Sites” [Sha06].

This thesis showed the uses of Web content mining for online news sites. The method applied dynamic schemes for exploring these Web sites and extracting news reports. The most important objective of the system is the discovery of ephemeral associations that can be translated into knowledge about interest of society and social behavior. The discovery of the kind of news trends helps to interpret the society interests and uncover hidden information about the

relationships between the events in social life and to measure the social importance of many events.

6. “Automatic Web Text Classification Using Data Mining” [Sar06].

This thesis gives a description of the design and implementation of a document classification system. By utilizing both high-level features of HTML documents and traditional term frequency information, it is hoped to achieve better classification results than systems that rely solely on text or document structure Web. Also it influence the full amount of information contained within an HTML document in order to classify it, and present approaches and techniques for automatic Web text classification using association rule mining in the data mining area, from pre-mining processes, association rule mining, to post mining process.

1.6 Aim of Thesis

The aim of this work is to find a new method for retrieving information by using web content mining techniques. The proposed method has two new points. The first one is to use a new equation for computing the weights of words in a document instead of using the traditional ones like TF-IDF method. The new equation for computing words' weights takes in consideration the frequency and attributes of these words.

The second point is to use fuzzy logic in page similarity instead of the traditional techniques like Cosine Similarity. This new method tries to prove that using Artificial Intelligent techniques such as fuzzy logic may give better results than the traditional used ones.

1.7 Thesis outlines

This is the summary of the content of the subsequent chapter of this thesis:

- 1. Chapter two:** It involves the description of Data Mining. Also explain Web Mining with their operations and discussed each type of it then describe Information Retrieval with Fuzzy Logic.
- 2. Chapter three:** this chapter presents the proposed system architecture, the relational module that used in this system and the algorithms that used to implement this system.
- 3. Chapter four:** this chapter gives the implementation and interface of the proposed system.
- 4. Chapter five:** this chapter explores conclusions of this work, and the suggestion for future works.



Chapter Two

Web Mining and Fuzzy Logic

Chapter Two

Web Mining and Fuzzy Logic

1.1 Introduction

The aim of this chapter is to define Data Mining and Web Mining with their architecture, techniques and taxonomies. Also gives definitions to Information Retrieval module and type of query with document similarity. Fuzzy Logic is defined with sets. HTML and XML describe with examples. All those give as sufficient knowledge to help for understanding this project.

2.2 Data Mining

Data mining is the analysis of (often large) observational data sets to find unsuspected relationships and to summarize the data in novel ways that are both understandable and useful to the data owner [Dan06].

Data mining is also called Knowledge Discovery in Databases (KDD). It is commonly defined as the process of discovering useful patterns or knowledge from data sources, e.g., databases, texts, images, the Web, etc. The patterns must be valid, potentially useful, and understandable. Data

mining is a multi-disciplinary field involving machine learning, statistics, databases, artificial intelligence, information retrieval, and visualization.

There are many data mining tasks. Some of the common ones are supervised learning (or classification), unsupervised learning (or clustering), association rule mining, and sequential pattern mining.

A data mining application usually starts with an understanding of the application domain by data analysts (data miners), who then identify suitable data sources and the target data. With the data, data mining can be performed, which is usually carried out in three main steps [Bin07]:

1. **Pre-processing:** The raw data is usually not suitable for mining due to various reasons. It may need to be cleaned in order to remove noises or abnormalities. The data may also be too large and/or involve many irrelevant attributes, which call for data reduction through sampling and attribute selection.
2. **Data mining:** The processed data is then fed to a data mining algorithm which will produce patterns or knowledge.
3. **Post-processing:** In many applications, not all discovered patterns are useful. This step identifies those useful ones for applications. Various evaluation and visualization techniques are used to make the decision.

The whole process (also called the data mining process) is almost always iterative. It usually takes many rounds to achieve final satisfactory results, which are then incorporated into real-world operational tasks. Traditional data mining uses structured data stored in relational tables, spread sheets, or flat files in the tabular form. With the growth of the Web

and text documents, Web mining and text mining are becoming increasingly important and popular [Bin07].

2.3 Data Mining Systems Classification [Osm99]

There are many data mining systems available or being developed, some are specialized systems dedicated to a given data source or are confined to limited data mining functionalities, others are more versatile and comprehensive. Data mining systems can be categorized according to various criteria. Among other classifications are the following:

1. Classification according to the type source mined: This classification categorizes data mining systems according to the type of data handled such as, spatial data, multimedia data, time-series data, text data, WWW, etc.
2. Classification according to the data model drawn on: This classification categorizes data mining systems based on the data model involved such as relational database, object-oriented database, data warehouse, transactional, etc.
3. Classification according to the kind of knowledge discovered: This classification categorizes data mining systems based on the kind of knowledge discovered or data mining functionalities, such as characterization, discrimination, association, classification, clustering, etc. Some systems tend to be comprehensive systems offering several data mining functionalities together.

4. Classification according to mining techniques used: Data mining systems employ and provide different techniques. This classification categorizes data mining systems according to the data analysis approach used such as machine learning, neural networks, genetic algorithms, statistics, visualization, database oriented or data warehouse- oriented, etc.

2.4 Kinds of Data that can be Mined [Osm99]

In principle, data mining is not specific to one type of media or data. Data mining should be applicable to any kind of information repository. Here are kinds of data that can be mined:

1. **Flat files:** Flat files are actually the most common data source for data mining algorithms, especially at the research level. Flat files are simple data files in text or binary format with a structure known by the data mining algorithm to be applied.
2. **Relational Databases:** Briefly, a relational database consists of a set of tables containing either value of entity attributes from entity relationships. Tables have columns and rows, where columns represent attributes and rows represent tuples.
3. **Data Warehouse:** A data warehouse is a repository of information collected from multiple sources under a unified schema, and which usually resides at a single site. Data warehouses are constructed via process of data cleaning, data transformation, data loading, and periodic data refreshing in order to facilitate decision making, the data in a data warehouse is organized around major subjects such as customer, supplier, and item.

4. **Transaction Databases:** In general, a transactional database consists of a file where each word represents a transaction. A transaction typically includes a unique transaction identify number (trans-ID) and list of the items making up the transaction.
5. **Multimedia Databases:** Multimedia databases include video, images, audio, and text media. They can be stored on extended object relational or object- oriented databases, or simply on a file system.
6. **Spatial Databases:** Special databases that, in addition to usual data, store geographical information, like maps such spatial databases represent new challenges to data mining algorithms.
7. **World Wide Web:** the WWW is the most heterogeneous and dynamic repository available. A very large number of authors and publishers are continuously contributing to its growth and a massive number of users are accessing its resources daily. Data in the WWW is organized in inter-connected documents. These documents can be text, audio, raw data, and even applications. Conceptually, the WWW is comprised of three major components: the content of the Web, which covers the hyperlinks and the relationships between documents; and the usage of the Web, describing how and when the resources are accessed. A Forth dimension can be added relating the dynamic nature or evaluation of the documents. Data mining in the WWW, or Web mining, tries to address all these issues and is often divided into three type mentioned previously.

2.5 Data Mining Tasks

Data mining methods may be classified by the function they perform or according to the class of application they can be used in. The main techniques used in data mining are described in the following sections:

2.5.1 Classification

Classification is the process of finding a set of models from the database, and in this case of supervised learning, this requires the user to define one or more classes. So classification determines whether an object belong to a given class, chosen among a set of predefined classes, based on the value of some object attribute i.e. based on a given classification function. It is referred to as supervised learning as the classes are determined prior to examining the data [Jao00].

A decision tree is a graphical representation of a collection of classification rules. So decision trees are a way of representing a series of rules that lead to a class or value given a data record, the tree directs the record from the root to a leaf. Each internal node of the tree is labeled with a predictor attribute. This attribute is often called a splitting attribute, because the data is ‘split’ based on conditions over this attribute. The outgoing edges of an internal node are labeled with predicates that involve the splitting attribute of the node; every data record entering the node must satisfy the predicate labeling exactly one outgoing edge [Her04].

2.5.2 Clustering

Clustering is an unsupervised process through which objects are classified into groups called clusters. The task of the system is to learn the descriptions of classes in order to be able to classify a new unlabeled object. Clustering is useful in a wide range of data analysis fields, including data mining, document retrieval, image segmentation, and pattern classification.

Clustering Analysis is an unsupervised learning environment, the system has to discover its own classes and one way in which it does this, is to cluster the data in the database. Clustering and segmentation basically partition the database so that each partition or group is similar according to some criteria [Jia00] [Bin07].

A clustering task may include the following components: problem representation, including feature extraction, selection, or both, definition of proximity measure suitable to the domain, Actual clustering of objects, Data abstraction, and Evaluation.

Several different variants of an abstract clustering problem exist. A flat (partitional) clustering produces a single partition of a set of objects into disjoint groups, whereas a hierarchical clustering results in a nested series of partitions. Each of these can either be a hard clustering or a soft one. In a hard clustering, every object may belong to exactly one cluster. In soft clustering, the membership is fuzzy-objects may belong to several clusters with a fractional degree of membership in each [Ron07].

Clustering is also a valuable technique for analyzing the Web. Matching the content-based clustering and the hyperlink structure can reveal patterns, duplications, and other interesting structures on the Web [Zdr07].

2.5.3 Association Rule

Association rule mining finds interesting association or correlation relationships among large set of data items. With massive amount of data continuously being collected and stored, many industries are becoming interested in mining association rules from their database. The discovery of interesting association relationships among huge amount of business transaction records can help in many business decision making processes [Jia00].

2.5.4 Neural Networks

Neural Network are a particular interest because they offer a means of efficiently modeling large and complex problems in which there may be hundreds of predictor variables that have many interactions. A neural network starts with an input layer, where each node corresponds to a predictor variable. These input nodes are connected to a number of nodes in a hidden layer which may be connected to nodes in another hidden layer, or to an output layer. The output layer consists of one or more response variables [Her04].

2.5.5 Genetic Algorithms

Genetic algorithms are not used to find patterns, but rather to guide the learning process of data mining algorithms such as neural nets. Essentially, genetic algorithms act as a method for performing a guided search for good models in the solution space. They are called genetic algorithms because they loosely follow the pattern of biological evaluation in which the

members of one generation (of models) compete to pass on their characteristics to the next generation (of models), until the best (model) is found [Her99].

2.6 Web Characteristics [Bin05] [Bin07]

The rapid growth of the Web it makes the largest publicly accessible data source in the world. The Web has many unique characteristics, which make mining useful information and knowledge a fascinating and challenging task. Some of these characteristics are:

1. The amount of data/information on the Web is huge and still growing. The coverage of the information is also very wide and diverse. One can find information on almost anything on the Web.
2. Data of all types exist on the Web, e.g., structured tables, semi-structured Web pages, unstructured texts, and multimedia files (images, audios, and videos).
3. Information on the Web is heterogeneous. Due to the diverse authorship of Web pages, multiple pages may present the same or similar information using completely different words and/or formats. This makes integration of information from multiple pages a challenging problem.
4. A significant amount of information on the Web is linked. Hyperlinks exist among Web pages within a site and across different sites. Within a site, hyperlinks serve as information organization mechanisms. Across different sites, hyperlinks represent implicit conveyance of authority to the target pages. That is, those pages that

- are linked (or pointed to) by many other pages are usually high quality pages or authoritative pages simply because many people trust them.
5. The information on the Web is noisy. The noise comes from two main sources. First, a typical Web page contains many pieces of information, e.g., the main content of the page, navigation links, advertisements, copyright notices, privacy policies, etc. Second, due to the fact that the Web does not have quality control of information, i.e., one can write almost anything that one likes, a large amount of information on the Web is of low quality, erroneous, or even misleading.
 6. The Web is dynamic. Information on the Web changes constantly. Keeping up with the change and monitoring the change are important issues for many applications. Also The Web is a virtual society. The Web is not only about data, information and services, but also about interactions among people, organizations and automated systems. One can communicate with people anywhere in the world easily and instantly, and also express one's views on anything in Internet forums, blogs and review sites.

All these characteristics present both challenges and opportunities for mining and discovery of information and knowledge from the Web.

2.7 Web mining

The WWW servers is a huge, widely distributed, global information service center for news, advertisement, consumer information, financial management, education, government, e-commerce, and many other information services. The Web also contains a rich and dynamic collection

of hyperlink information and Web page access and usage information, providing rich sources for data mining.

Only a small portion of the information on the Web is truly relevant or useful. Although this may not seem obvious, it is true that a particular person is generally interested in only a tiny portion of the Web while the rest of the Web contains information that is uninteresting to the user and may swamp desired search results [Jia00].

The causation that appears is how can a search identify that portion of the Web that is truly relevant to one user's interests? And how can a search find high quality Web pages on a specified topic?

Currently, users can choose from two major approaches when accessing information stored on the Web [Eti96, Ray00]:

1. **Keyword-based search** or topic directory browsing with search engines such as Google or Yahoo, which use keyword indices or manually built directories to find documents with specified keywords or topics;
2. **Querying deep Web sources**: where information, such as Amazon.com book data, hides behind searchable database query forms that, unlike the surface Web, cannot be accessed through static URL links; and basically, Web mining is concerned with the use of data mining techniques to automatically discover and extract information from WWW documents and services, which is categorized in three areas of interest: Web Usage Mining, Web Structure Mining and Web Content Mining.

WUM finds access patterns from Web sites while WSM provides structured information about Web documents and sites, and WCM is resource discovery on the Internet is still frustrating and sometimes even useless when simple keyword searches can convey hundreds of thousands of documents as results, knowledge discovery and data mining from the Web is a new promising result topic that is attracting tremendous interest [Osm04].

There are a suggest decomposing Web mining task into four subtasks: [Ray00, Sou03, Joh06]

1. **Resource finding:** This is the process of retrieving data, which is either online or offline, from the multimedia sources on the Web, such as electronic newsletters, electronic newswire, news groups, and the text content of HTML documents obtained by removing the HTML tags.
2. **Information selection and preprocessing:** This is the process by which different kinds of original data retrieved in the previous subtask are transformed. These transformations could be either a kind of preprocessing such as removing stop words, stemming, etc. or a preprocessing aimed at obtaining the desired representation, such as finding phrases in the training corpus, representing the text ... etc.
3. **Generalization:** Generalization is the process of automatically discovering general patterns within individual Web sites as well as across multiple sites. Different general-purpose machine-learning techniques, data mining techniques, and specific Web-oriented methods are used.

4. **Analysis:** This is a phase in which validation and/or interpretation of the mined patterns is performed.

2.8 Taxonomy of Web Mining [Cha03, Wan03, Wes05]

Taxonomy of Web mining is based on which part of the Web to be mined, and it consists of three areas, as show in Figure (2.1).

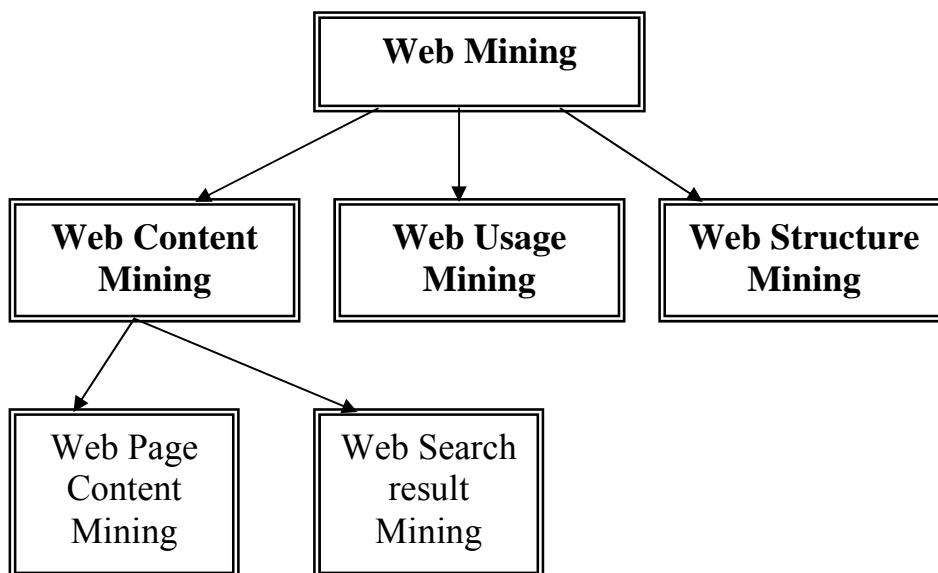


Figure (2.1): Taxonomy of Web Mining

2.8.1 Web Usage Mining

WUM is the application of data mining techniques to discover interesting usage patterns from Web data, in order to understand and better serve the needs of Web-based applications [Wes05]. Web usage mining

refers to the automatic discovery and analysis of patterns in clickstream and associated data collected or generated as a result of user interactions with Web resources on one or more Web sites. The goal of WUM is to capture, model, and analyze the behavioral patterns and profiles of users interacting with a Web site. The discovered patterns are usually represented as collections of pages, objects, or resources that are frequently accessed by groups of users with common needs or interests [J0h06].

Usage data captures the identity or origin of Web users along with their browsing behavior at a Web site. Web usage mining itself can be classified further depending on the kind of usage data considered [Wes05]:

1. **Web Server Data:** The user logs are collected by Web server. Typical data includes IP address, page reference and access time.
2. **Application Server Data:** Commercial application servers, e.g. Web logic, StoryServer, etc. have significant features in the framework to enable e-commerce applications to be built on top of them with little effort. A key feature is the ability to track various kinds of business events and log them in application server logs.
3. **Application Level Data:** New kinds of events can always be defined in an application, and logging can be turned on for them generating histories of these specially defined events.

The overall Web usage mining process can be divided into three interdependent tasks: data preprocessing, pattern discovery, and pattern analysis or application. In the preprocessing stage, the clickstream data is

cleaned and partitioned into a set of user transactions representing the activities of each user during different visits to the site. In the pattern discovery stage, statistical, database, and machine learning operations are performed to obtain possibly hidden patterns reflecting the typical behavior of users, as well as summary statistics on Web resources, sessions, and users. In the final stage of the process, the discovered patterns and statistics are further processed, filtered, and used as input to applications such as recommendation engines, visualization tools, and Web analytics and report generation tools [Joh06].

The usage data can also be split into three different kinds on the basis of the source of its collection: on the server side, the client side, and the proxy side. The key issue is that on the server side there is an aggregate picture of the usage of a service by all users, while on the client side there is complete picture of usage of all services by a particular client, with the proxy side being somewhere in the middle [Weso5].

2.8.2 Web Structure Mining

The interconnections between hypertext documents, the WWW can reveal more information than just the information contained in document indicates the popularity of the document, while links coming out of a document indicate the richness or perhaps the Variety of topics covered in the document [Osm99].

The structure of a typical Web graph consists of Web pages as nodes, and hyperlinks as edges connecting related pages. Web Structure Mining is the process of discovering structure information from the Web. This can be

further divided into two kinds based on the kind of structure information used [Jia00] [Kha03] [Wes05]:

1. **Hyperlinks:** A Hyperlink is a structural unit that connects a location in a Web page to different location, either within the same Web page or on a different Web page. A hyperlink that connects to a different part of the same page is called an Intra-Document Hyperlink, and a hyperlink that connects two different pages is called an Inter-Document Hyperlink.
2. **Document Structure:** In addition, the content within a Web page can also be organized in a tree-structured format, based on the various HTML and XML tags within the page.

Web structure mining is a research field focused on using the analysis of the link structure of the Web, and one of its purposes is to identify more preferable documents. Data of structure Web mining represent the way content is organized. They can be either data entities used within a Web page, such as HTML or XML tags or data entities used to put a Web site together, such as hyperlinks connecting one page to another [Mag03].

2.8.3 Web Content Mining

WCM is an automatic process that goes beyond keyword extraction. The other definition of content mining is the process of extracting useful information from the contents of Web documents [Joh06]. It may consist of text, images, audio, video, or structured records such as lists and tables. Application of text mining to Web content has been the most widely

researched. Issues addressed in text mining are, topic discovery, extracting association patterns, clustering of Web documents and classification of Web Pages. Research activities on this topic have drawn heavily on techniques developed in other disciplines such as Information Retrieval (IR) and Natural Language Processing (NLP) [Wes05]. Since the content of a text document presents no machine readable semantic, some approaches have suggested restructuring the document content in a representation that could be exploited by machines. Since the lack of structure that permeates the information sources on the WWW makes automated discovery of Web based information difficult [Coo97].

Others consider the Web structured enough to do effective Web mining. (nevertheless, in either cases an intermediary representation is often relied upon and built using known structure of a limited type and set of documents (or sites) or using typographic and linguistic properties. The semi-structured nature of most documents on the Internet helps in this task [Coo97].

Content data correspond to the collection of facts a Web page was designed to convey to the users. Web content mining can take advantage of the semi-structured nature of Web page text. The HTML tags or XML markup within Web pages bear information that concerns not only layout but also the logical structure and semantic content of documents [Joh06].

There are two groups of Web content mining strategies; those that directly mine the content of documents and those that improve on the content search of other tools like search engines [Coo97].

2.8.3.1 Web Page Content Mining

There has been a lot of research in retrieving information from structured data, hypertext or semi-structured documents. However, most of the suggested approaches are limited to a known set of documents and use custom-made wrappers to map the content of these documents to an internal representation. Shopping agents, furthermore, learn to recognize document structures of online catalogs and e-commerce Web sites and extract price lists and special offers. This agent can compile information retrieval from different sites and discover bargains. Mysimon.com is an example of such a service. A major obstacle for efficient information extraction from text documents is the absence of reliable HTML-based metadata and the lack of a standard way to describe, manipulate and exchange data in electronic documents [Osm99].

2.8.3.2 Web Search Result Mining

The heterogeneity of the WWW and the absence of structure have led some researchers to mine subsets of known documents or data from documents known to pertain to a given topic. One subset can be a search result of a query sent to a search engine like Yahoo or AltaVista.

A system uses search engines to retrieve relevant documents and collects information either from within the documents or data provided by servers like the URL, title, content type and modification type.

Search result mining relies on information provided in search results like URLs and snippets to induce clusters and categorize the retrieved documents in these discovered clusters. The clusters, which can present overlapping, represent a higher-level view on top of the list of retrieved

documents and facilitate the sifting through the often very large search engine result list [Osm99].

2.8.3.3 Web Content Mining Approaches

Two different points of Web content mining research can be viewing: IR views and DB views. The goal of Web content mining from the IR view is mainly to assist or to improve the information finding or filtering the information to the users usually based on either inferred or solicited user profiles, while the goal of Web content mining from the DB view mainly tries to model the data on the Web and to integrate them so that more sophisticated queries other than the keywords based search could be performed.

The appropriate taken in Web content mining could be categorized into three main groups:

1. Information retrieval from unstructured data source.
2. Information retrieval from Simi-structure data sources.
3. Database oriented modeling of the Web.

In the following will look at some of the work done in each of the above categories.

1. Information retrieval from unstructured data sources [Gay03]

Unstructured is mainly free text where a scheme or meta data describing the text is not provided, they include textual description found in Web site such as news stories; feature articles etc.

2. Information retrieval from semi-structured data sources [Sve04]

In order to understand the methods used in extracting information from Semi-structured data it is important to define what exactly semi-structured data means.

Semi-structure data refer to data with some of the following characteristics:

1. A schema is not given in advance and may be implicit in the data.
2. The schema is relatively large and many be changing frequently.
3. The schema is descriptive rather than prescriptive, i.e., it describes the current state of data.
4. Data is not strongly typed.

The aims of Web content mining is easier said than done. The challenges are mainly due to the nature of the data available on the Web. More and more data formats and services are open to the Web continuously.

A good measure of the data format heterogeneity is to see how many different data format are supported by the newly released Web browser such as Netscape 7.0 these include data format such as text(e.g.: HTML, XML etc.), image, video, and other binary format such as PDF.

Further with many enterprises Web enabling their business processes, most of the new sites are data driven (dynamic) as opposed to classic static HTML model. As result many search engines indexes are of no use against dynamically generated data.

However Web content mining is limited to three main categories based on the type of data being processed. These three categories are:

(Unstructured data (includes free text), Semi-structured data (such as HTML) and Structure data (such as XML).

As seen clearly the multimedia data is not present in the majority of the Web content mining work. The inherent difficulties in mining multimedia data and text dominant of the Web are two main reasons for this [Gay03].

A good example of semi-structured data is HTML. HTML documents contain a loose structure inside with tags and have a global structure based on the hyper links.

Mining techniques applied to semi-structure data utilize the structural data in providing richer knowledge extraction techniques.

Application of the nature includes hypertext classification and clustering, learning relationships between Web documents and finding patterns in HTML documents.

3. Data base oriented modeling of the Web

Basically the Data Base (DB) view tries to infer the structure of the Web site or to transform a Web site to become a database so that better information management and querying on the Web become possible and tries to model the data on the Web and to integrate them so that more sophisticated queries search could be performed [Ray00]. Also the database oriented modeling of the Web is mainly concerned with managing data on the Web in a manner as is done in conventional data bases [Dan05].

In the DB view, Web documents are considered to be much more structure than in IR view. Documents on the Web are defined in different ways the most commonly uses are HTML, XML:

A. HyperText Markup Language (HTML) Documents

Documents on the Web are defined in the HyperText Markup Language (HTML) [Sou03]. When mining inside an HTML document, the structure of the document as indicated by the HTML tags will be exploited. However, the structure imposed by HTML is purely for presentation purposes. Indeed, HTML only provides tags to specify the title of the document, to partition the document into paragraphs, to indicate lists, tables, hyperlinks, and so on. The HTML file, for instance, displays the page in figure (2.2) and could be part of a Web page of a computer vendor where each separate page contains the data of each offered computer. HTML tags are the words between brackets and determine how the text in between should be display. For instance, the text Laptop 44X3D between the start-tag `<TITEL>` and the end-tag `</TITEL>` is the text display in the title bar of the Web browser. Two matching tags to gather with the text in between is called an element. Further, `<BODY>` specifies the content of the THML file, this content is enclosed with in `<BODY>` and `</BODY>`, each `<L1>` determines a list item and `<H1>` `<H2>` are used for heading, `<A>` is used for link, etc. An example of the HTML file is given below [Pie03] [Sou03].

Although HTML, based on tags, is an excellent mechanism to provide platform independent browsing, it hardly any semantics. Clearly, ABT, grey, and 2000 are properties of the laptop 44X3D and a human can defer their meaning, but not a computer program. Additionally, it could be possible that different lap-top models have different properties and there is no way to specify this in HTML, while keeping the structure and the content of the document separated. See appendix B.

```
<HTML>
< HEAD>
<TITEL> Laptop 44x3D </TITEL>
</HEAD>
<BODY>
<L1> ABT </L1>
<L1> gray</L1>
<L1> 2000</L1>
</BODY>
</HTML>>
```

Figure (2.2): HTML source file.

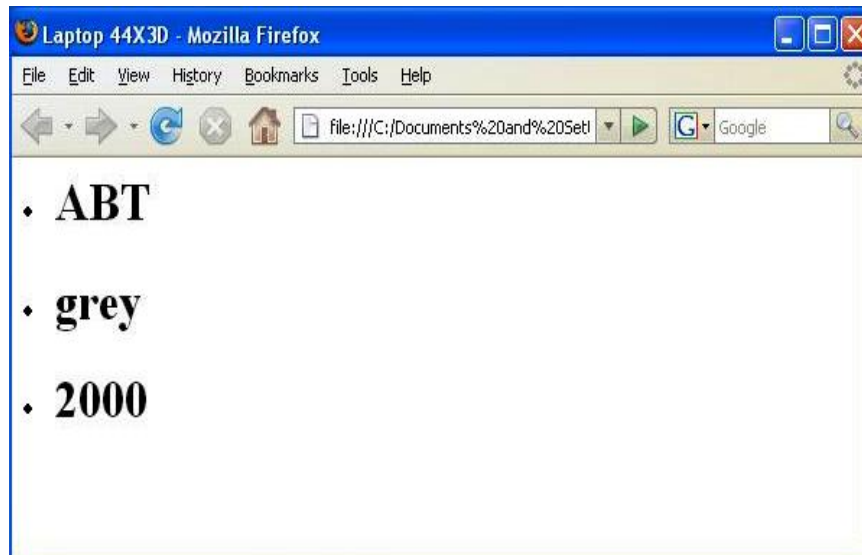


Figure (2.3): Screen grab of browser displaying.

B. eXtensible Markup Language (XML) Documents

eXtensible Markup Language (XML) is a meta-language for describing markup languages. XML provides a facility to define tags and the structural relationships between them. It specifies neither semantics nor a tag set [Aut05].

Currently there is ongoing work in the area of what is called a “semantic Web”. The idea here is to make the Web more understandable to computer by providing semantic tags in documents. An important impulse in this direction is given by the use of XML instead of HTML for Web documents. XML is a new standard for the specification of structure documents developed by the WWW Consortium (W3C) and is essentially a cleaned up version of the Standard Generalized Markup Language (SGXML).

2.9 Information Retrieval

Information Retrieval (IR) is the automatic retrieval of all relevant documents while at the same time retrieving as few of the non-relevant as possible. IR has the primary goals of indexing text and searching for useful documents in a collection and nowadays research in IR includes modeling, document classification and categorization, user interfaces, data visualization, filtering, etc. Web mining is part of the (Web) IR process [Ray00]. Also IR helping users to find information that matches their information needs [Bin07]. Technically, IR studies the acquisition, organization, storage, retrieval, and distribution of information. Historically,

IR is about document retrieval, emphasizing document as the basic unit. Figure (2.5) gives a general architecture of an IR system.

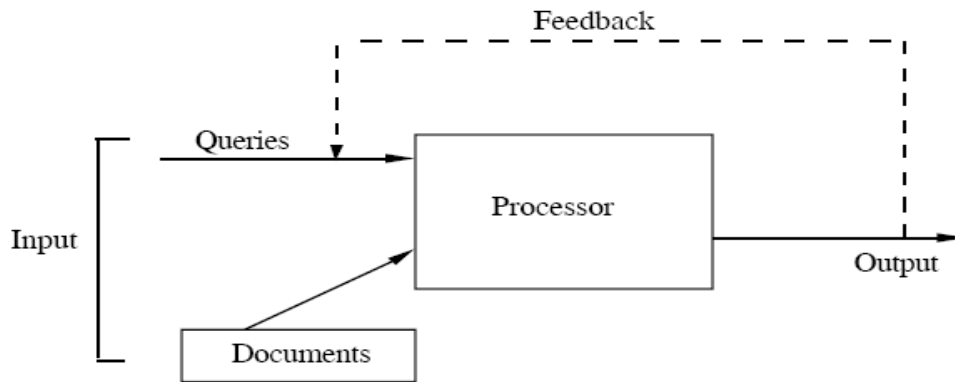


Figure (2.5): A general IR system architecture.

In figure above shows components of IR system. The main problem here is to obtain a representation of each document and query suitable for a computer to use. The processor, that part of the retrieval system concerned with the retrieval process. The process may involve structuring the information in some appropriate way, such as classifying it. It will also involve performing the actual retrieval function that is, executing the search strategy in response to a query. When the retrieval system is on-line, it is possible for the user to change his/her request during one search session in the light of sample retrieval, thereby; it is hoped, improving the subsequent retrieval run. Such a procedure is commonly referred to as feedback. The output is usually a set of citations or document numbers. Document representation is illustrating in Information Retrieval Models below [Cor79].

A user query represents the user's information needs, which is in one of the following forms: [Bin07]

1. **Keyword queries:** The user expresses his/her information needs with a list of (at least one) keywords (or terms) aiming to find documents that contain some (at least one) or all the query terms.
2. **Boolean queries:** The user can use Boolean operators, AND, OR, and NOT to construct complex queries. Thus, such queries consist of terms and Boolean operators.
3. **Phrase queries:** Such a query consists of a sequence of words that makes up a phrase. Each returned document must contain at least one instance of the phrase. In a search engine, a phrase query is normally enclosed with double quotes.
4. **Proximity queries:** The proximity query is a relaxed version of the phrase query and can be a combination of terms and phrases. Proximity queries seek the query terms within close proximity to each other. The closeness is used as a factor in ranking the returned documents or pages.
5. **Full document queries:** When the query is a full document, the user wants to find other documents that are similar to the query document. Some search engines (e.g., Google) allow the user to issue such a query by providing the URL of a query page. Additionally, in the returned results of a search engine, each snippet may have a link called “more like this” or “similar pages.” When the user clicks on the link, a set of pages similar to the page in the snippet is returned. In this thesis will has been used this query to retrieve similar pages.
6. **Natural language questions:** This is the most complex case, and also the ideal case. The user expresses his/her information need as a natural language question. The system then finds the answer.

However, such queries are still hard to handle due to the difficulty of natural language understanding.

2.9.1 Information Retrieval Models

An IR model governs how a document and a query are represented and how the relevance of a document to a user query is defined. There are four main IR models: Boolean model, vector space model, language model and probabilistic model. They all treat each document or query as a “bag” of words or terms. A term is simply a word whose semantics helps remember the document’s main themes. Each term is associated with a weight. Vector representation, a collection of documents is simply represented as a relational table. Each term is an attribute, and each weight is an attribute value.

2.9.1.1. Boolean Model

The Boolean model is one of the earliest and simplest information retrieval models. It uses the notion of exact matching to match documents to the user query. Both the query and the retrieval are based on Boolean algebra. Documents and queries are represented as sets of terms.

The weight w_{ij} ($\in \{0, 1\}$) of term t_i in document \mathbf{d}_j is 1 if t_i appears in document \mathbf{d}_j , and 0 otherwise, i.e.

$$W_{ij} = \left. \begin{array}{l} 1 \quad \text{if } t_i \text{ appears in } \mathbf{d}_j \\ 0 \quad \text{otherwise.} \end{array} \right\} \dots\dots\dots (2.1).$$

Query terms are combined logically using the Boolean operators AND, OR, and NOT, which have their usual semantics in logic. Given a Boolean query, the system retrieves every document that makes the query logically true. Document is either relevant or irrelevant.

2.9.1.2. Vector Space Model

Text documents can be conveniently represented in a high-dimensional vector space where terms are associated with vector components. More precisely, a text document d can be represented as a sequence of terms, $d = (\omega(1), \omega(2), \dots, \omega(|d|))$, where $|d|$ is the length of the document and $\omega(t) \in V$. Vector-based representations are sometimes referred to as a “bag of words” [Pie03]. A document in the vector space model is represented as a weight vector, in which each component weight is computed based on some variation of TF or TF-IDF scheme. The weight w_{ij} of term t_i in document d_j can be any number between $\{0, 1\}$.

A. Term Frequency (TF) Scheme: In this method, the weight of a term t_i in document \mathbf{d}_j is the number of times that t_i appears in document \mathbf{d}_j , denoted by f_{ij} . Normalization may also be applied (see Equation (2.2)). The shortcoming of the TF scheme is that it does not consider the situation where a term appears in many documents of the collection. Such a term may not be discriminative [Bin07].

$$tf_{ij} = \frac{f_{ij}}{\max\{f_{1j}, f_{2j}, \dots, f_{|V|j}\}} \dots\dots\dots (2.2)$$

B. TF-IDF Scheme: This is the most well known weighting scheme combines Term Frequencies (which are relative to each document) with an ‘absolute’ measure of term importance called Inverse Document Frequency (IDF). IDF decreases as the number of documents in which the term occur increases in a given collection. So terms that are globally rare receive a higher weight. The TF stands for the **Term Frequency** and IDF the **Inverse Document Frequency**.

Formally, let $D = \{d_1, \dots, d_n\}$ be a collection of documents and for each term ω_j let n_{ij} denote the number of occurrences of ω_j in d_i and n_j the number of documents that contain ω_j at least once. Then define

$$\text{TF}_{ij} = n_{ij} / |d_i| \quad (2.3)$$

$$\text{IDF}_j = \text{Log } n/n_j \quad (2.4)$$

Here the logarithmic function is employed as a damping factor. The TF-IDF weight of ω_j in d_i can be computed as [Pie03] [Chr08]:

$$X_{ij} = \text{TF}_{ij} \cdot \text{IDF}_j \quad (2.5)$$

A query \mathbf{q} is represented in exactly the same way as a document in the document collection. The term weight w_{iq} of each term t_i in \mathbf{q} can also be computed in the same way as in a normal document, or slightly differently. For example, Salton and Buckley

2.9.1.3 Statistical Language Model

Statistical language models (or simply language models) are based on probability and have foundations in statistical theory. The basic idea of this approach to retrieval is simple. It first estimates a language model for each document, and then ranks documents by the likelihood of the query given the language model. Similar ideas have previously been used in natural language processing and speech recognition.

2.9.2 Document similarity [Pie03, Bin07, Chr08]

Can define similarity between two documents d and d' as a function $s(d, d') \in \mathbb{R}$. The documents are ranked according to their degrees of relevance to the query. One way to compute the degree of relevance is to calculate the similarity of the query \mathbf{q} to each document \mathbf{d}_j in the document collection D . There are many similarity measures. The most well known one on the vector space representation and the metric defined is the **cosine similarity**, which is the cosine of the angle between two documents, \mathbf{x} and \mathbf{x}' (the query vector \mathbf{q} and the document vector \mathbf{d}_j). See figure (2.6).

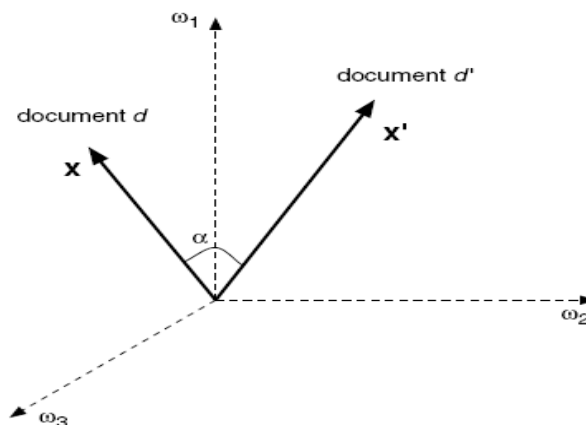


Figure (2.6): cosine measure of document similarity.

where the superscript ‘T’ denotes the ‘transpose’ operator and $\mathbf{x} \cdot \mathbf{y}$ indicates the dot product or inner product between two vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^m$, defined as

$$\cos(\mathbf{x}, \mathbf{x}') = \frac{\mathbf{x}^T \mathbf{x}'}{\|\mathbf{x}\| \cdot \|\mathbf{x}'\|} = \frac{\mathbf{x}^T \mathbf{x}'}{\sqrt{\mathbf{x}^T \mathbf{x}} \cdot \sqrt{\mathbf{x}'^T \mathbf{x}'}} \quad \dots\dots\dots (2.6)$$

Note that in the case of two sparse vectors \mathbf{x} and \mathbf{y} associated with two documents d and d' , the above sum can be computed efficiently in time $\Omega(|d| + |d'|)$.

2.10 Benefits of Web Mining

There are many benefits of Web mining and some of these are:

1- Understand customer behavior:

- A. Companies can optimize e-business sites for maximum commercial impact by understanding the dynamic behavior of visitors to their Web sites.
- B. E-tailors can know gain knowledge on the individual tastes and preferences of the visitors to their sites.
- C. Determine the conversion rate of visitors to buyers on your site.
- D. Determine the repeat frequency of existing buyers (I.e., the likelihood of customers repurchasing your brand).
- E. Calculate the rate of new customer acquisition.
- F. Discover actionable browsing and buying patterns of customers.
- G. Learn who is buying what from your site.

- H. Discover cross relationships between clients in your e-commerce sites.

2- Determine Web Site Effectiveness:

- A. Discover high and low impact area of your e-commerce site.
- B. Web administrators no longer have to rely on intuition when designing the layout of a Web site.
- C. E-tailors can now develop the look and feel of the Web site and personalize online content.

3- Measure the success of marketing efforts:

- A. In the physical world it is difficult to get reliable feedback on marketing campaigns. But, on the Internet you can get segment reel measurements, of the success of a marketing campaign.
- B. Companies can cluster customers with similar patterns, and the Web site can adapt to recognized customers. Segments can then be targeted with campaigns and special offers [Jia02].

2.11 Fuzzy Logic [Jam05]

Fuzzy logic is a superset of conventional (Boolean) logic that has been extended to handle the concept of partial truth- truth values between "completely true" and "completely false". As its name suggests, it is the logic underlying modes of reasoning which are approximate rather than exact. The importance of fuzzy logic derives from the fact that most modes of human reasoning and especially common sense reasoning are

approximate in nature. The essential characteristics of fuzzy logic as founded by Zadeh Lotfi are as follows:

1. In fuzzy logic, exact reasoning is viewed as a limiting case of approximate reasoning.
2. In fuzzy logic everything is a matter of degree.
3. Any logical system can be fuzzified
4. In fuzzy logic, knowledge is interpreted as a collection of elastic or, equivalently, fuzzy constraint on a collection of variables
5. Inference is viewed as a process of propagation of elastic constraints.

2.12 Fuzzy Sets

Fuzzy sets are sets whose elements have degrees of membership. Fuzzy sets have been introduced by Lotfi A. Zadeh (1965) as an extension of the classical notion of set. In classical set theory, the membership of elements in a set is assessed in binary terms according to a bivalent condition; an element either belongs or does not belong to the set. By contrast, fuzzy set theory permits the gradual assessment of the membership of elements in a set; this is described with the aid of a membership function valued in the real unit interval $[0, 1]$. Fuzzy sets generalize classical sets, since the indicator functions of classical sets are special cases of the membership functions of fuzzy sets, if the latter only take values 0 or 1.

In fuzzy sets, each elements is mapped to $[0, 1]$ by membership function. Where $[0, 1]$ means real numbers between 0 and 1 (including 0, 1). Consequently, fuzzy set is ‘vague boundary set’ comparing with crisp set [Lee05].

$$\mu_A : X \rightarrow [0, 1]$$

2.13 Fuzzy Information Retrieval

Fuzzy information retrieval model shows how various fuzzy set techniques can be successfully applied to information engineering problems. Some of the problems that can be successfully addressed by fuzzy logic are i) the clarification and interpretation of information ii) the retrieval of information by querying and reasoning and iii) the utilization of information in decision making, designing and optimization tasks. Fuzzy query is also known as knowledge query and using this query imprecise data such as opinions, judgments and values can be expressed in linguistic terms, can be queried from the database. Thus to make fuzzy queries against a relational database one need to decompose the domain of database column into their underlying term sets. Information retrieval, fuzzy theory and database technology are still regarded as research domains, which concerns the representation and relevance of queries and data [Mru05].

2.14 Fuzzy Numbers [Wil05] [Lee05]

Fuzzy Numbers represent a number of whose values are somewhat uncertain. They are a special kind of fuzzy set whose members are numbers from the real line, and hence are infinite in extent. The function relating member number to its grade of membership is called a membership function.

A fuzzy number is a convex, normalized fuzzy set $\tilde{A} \subseteq \mathbb{R}$ whose membership function is at least segmentally continuous and has the functional value $\mu_A(x) = 1$ at precisely one element. It represents a real number interval whose boundary is fuzzy.

Fuzzy number is expressed as a fuzzy set defining a fuzzy interval in the real number R . Since the boundary of this interval is ambiguous, the interval is also a fuzzy set. Generally a fuzzy interval is represented by two end points a_1 and a_3 and a peak point a_2 as $[a_1, a_2, a_3]$ as shown in figure (2.7). It contains two types:

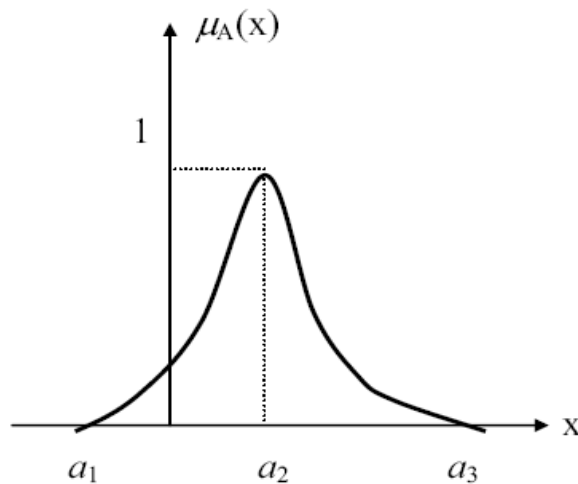


Figure (2.7): Fuzzy number $A = [a_1, a_2, a_3]$.

A. Triangular Fuzzy Number [Lee05]

Triangular Fuzzy Number is a fuzzy number represented with three points as follows:

$$A = (a_1, a_2, a_3)$$

this representation is interpreted as membership functions as shown in figure (2.8). Among the various shapes of fuzzy number, triangular fuzzy number (TFN) is the most popular one.

$$\mu_{(A)}(x) = \begin{cases} 0, & x < a_1 \\ \frac{x - a_1}{a_2 - a_1}, & a_1 \leq x \leq a_2 \\ \frac{a_3 - x}{a_3 - a_2}, & a_2 \leq x \leq a_3 \\ 0, & x > a_3 \end{cases} \dots\dots\dots (2.7)$$

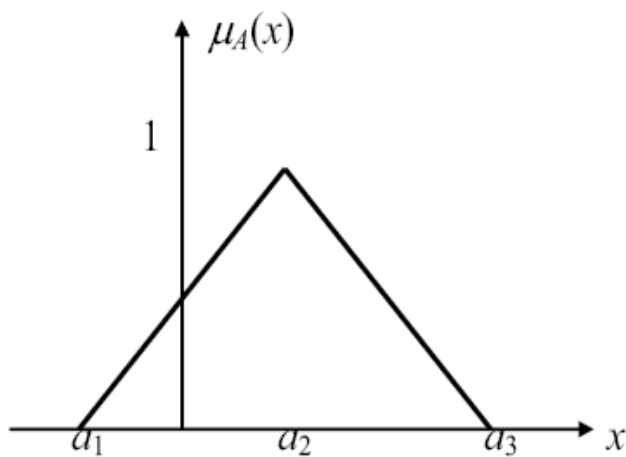


Figure (2.8): Triangular fuzzy number $A = (a_1, a_2, a_3)$.

B. Trapezoidal Fuzzy Number [Lee05]

Trapezoidal fuzzy number is a fuzzy number represented with the follows points:

A as

$$A = (a_1, a_2, a_3, a_4)$$

the membership function of this fuzzy number will be interpreted as follows.

Figure (2.9) shown the Trapezoidal fuzzy number.

Another shape of fuzzy number is trapezoidal fuzzy number. This shape is originated from the fact that there are several points whose membership degree is maximum ($\alpha = 1$).

$$\mu_A(x) = \begin{cases} 0, & x < a_1 \\ \frac{x - a_1}{a_2 - a_1}, & a_1 \leq x \leq a_2 \\ 1, & a_2 \leq x \leq a_3 \\ \frac{a_4 - x}{a_4 - a_3}, & a_3 \leq x \leq a_4 \\ 0, & x > a_4 \end{cases} \quad \dots\dots\dots (2.8)$$

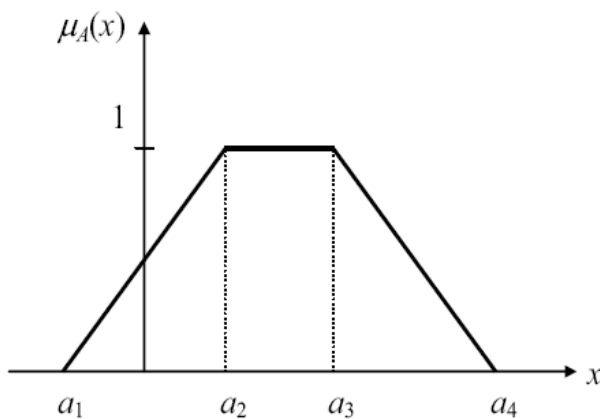


Figure (2.9): Trapezoidal fuzzy number $A = (a_1, a_2, a_3, a_4)$.



Chapter Three

*Web Pages
Fuzzy Similarity
System
Implementation*

Chapter Three

Web Pages Fuzzy Similarity

System Implementation

3.1 Introduction

In this chapter, the Web Content (Text) Mining is used to find the similarity between query page and the stored database that consists from a large number of documents (pages). In this thesis, a Web pages fuzzy similarity system is implementation. The similarity process is accomplished by using normal method and another by using fuzzy method. The work has been partitioned into two phases, Off-Line phase and On-Line phase. The Off-Line phase consists of many steps that help in building the document vector for the stored pages such as: Document Collection, Document Preprocessing, Ranking, Feature Extraction, Indexing, and Constructing Document Vectors. While On-Line phase consists of two steps, first read the query page and construct a query vector for it, and second perform the similarity between query page and DB document, then display the similar pages. Figure (3.1) shows the Web pages fuzzy similarity System.

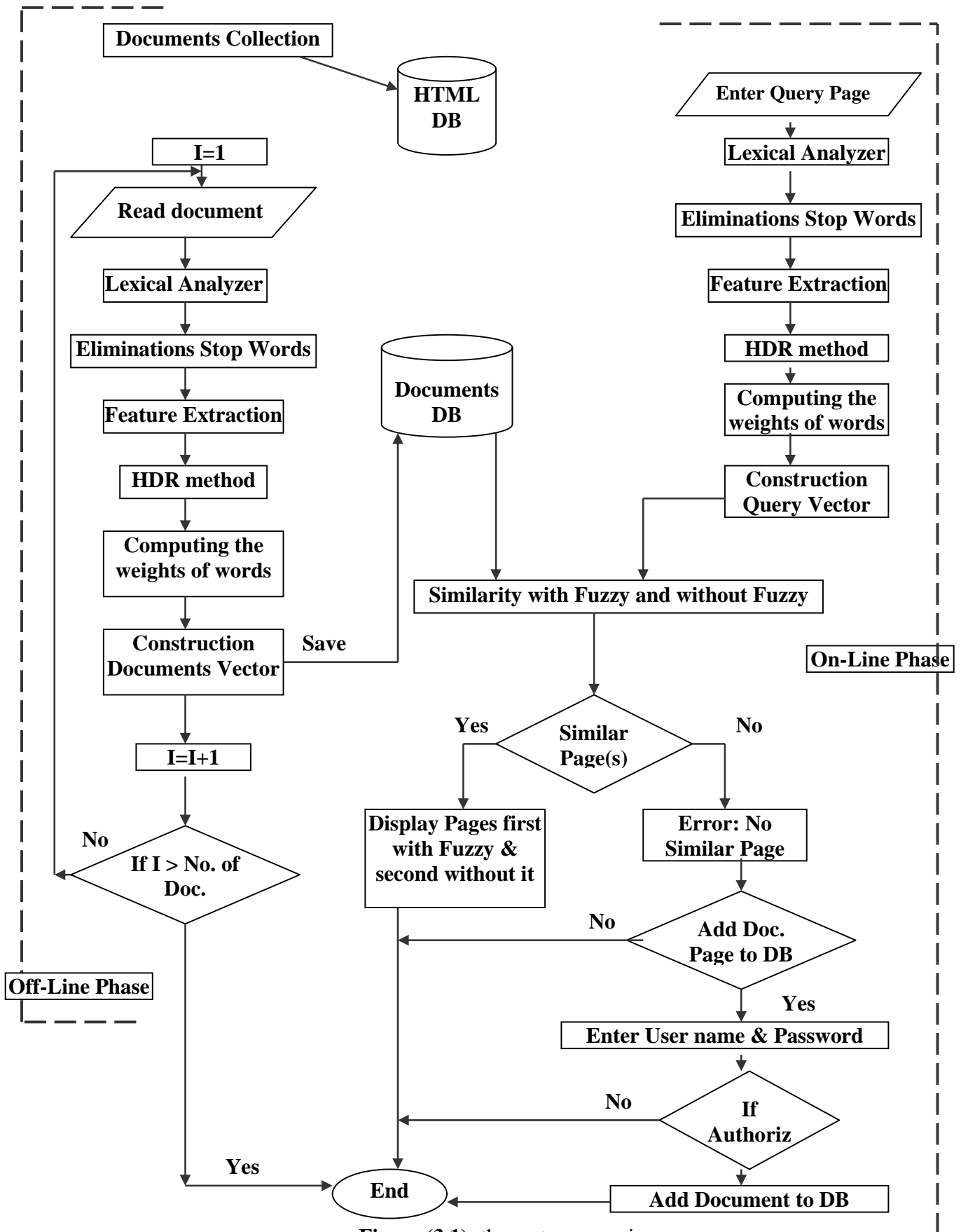


Figure (3.1): the system overview.

3.2 Web pages fuzzy similarity Module

Web pages fuzzy similarity module consists of two phases: **Off-Line** phase and the **On-Line** phase.

The **Off-Line** phase consists of:

1. Data Collection: download a number of HTML pages from WWW and store them in a DB (HTML DB).
2. Preprocessing of Web Page: which consists of:
 - A. Lexical Text Analysis: convert stream of characters in HTML documents to sets of words.
 - B. Elimination Stop Words, Special Character, Digits, and Unused Words.
3. Ranking: which consists of
 - A. HTML Document Feature Ranking.
 - B. Html Document Ranking (HDR) Method.
4. Term Indexing: which consists of:
 - A. Weights for Terms: compute the weight for all words by using certain formula.
 - B. Construction of Documents' Vector: construct documents' vectors from words and its weights.

The On-Line phase consists of:

1. Input Query Page: a user can enter the query page address in a text box and click on the search button to start searching for similar pages.
2. Similarity: take query page and construct the query vector to it. Then use similarity formula to compute membership between query page and documents in the database, first with fuzzy and second without fuzzy similarity.
3. Displaying results (Similar Pages): display the result of similarity step. The result is either displaying similar pages found sorted in descending order according to the membership between them and query pages, or no similar page is found to the query page, so, a note is displayed about that.

3.3 HTML Features Analysis

To extract a useful text features from existing HTML and natural language properties, a strategy must be set for taking the requirements of similarity task into consideration.

In order to describe the nature of the content similarity, a quantitative analysis was performed. The results are obtained by analyzing a collection of nearly (150) HTML documents (pages). According to the tags of HTML documents, the best quality and amount of features are extracted. The selection of rich feature location in the document is a very important matter for similarity process. A set of examples of Web documents are given.

Table (3-1) shows the percentage of Web HTML documents with a certain number of words for each type of HTML tags.

Table (3-1): Word Frequency Distribution in HTML tags.

Tag name	No. of Words			
	0	1-20	21-50	51-
<TITLE>	6%	93%	1%	0%
<META=Description>	70%	10%	15%	5%
<META=Keywords>	85%	2%	10%	3%
<BODY>	10%	7%	17%	66%

A number of words are counted in the content attributes of the <META NAME="keywords"> and <META NAME = "description"> tags as well as <TITLE> tags, also counting free text found within the <BODY> tag, without taking other tags in our consideration because of text weakness.

A 93% of the tested documents contain (1-20) words in the <TITLE>, and 6% of documents have no title. The <Meta =description> tag exists in 30% of documents, 70% do not have <Meta =description>, 15% contains (21-50) words. The <Meta=Keywords > tag exists in 15% of documents, 85% do not have <Meta=Keywords > tag, 10% contains (21-50) words. As seen there is no such dependency on <Meta=Keywords> tag and <Meta=description> tag, Because of smaller word percentage exists on HTML documents. The main amount of text can be extracted, is laid on the body tag <BODY>, 17% of the words are in the <BODY> tag which contains (21-50) words, and about 66% of the words are in the <BODY> tag which contains

above (51-) words. As shown the body tag can be the main text source in HTML documents. These results reflect the analysis of the experiment samples, which are obviously small and may not be one hundred fifty percent reflections of other Web document formats or types, but this example can be generalized since it is text based Web documents for similarity purpose.

3.4 Methodologies of System.

As shown in figure (3.1) similarity of pages using Web content mining consists of many tasks. These tasks are:

1. Data Collection.
2. Data Preprocessing.
3. Ranking.
4. Term Indexing.
5. Similarity.

3.4.1 Data Collection

Data Collection Consists of the following steps:

- A. Download Web Document.
- B. Cleansing.
- C. Text extraction.

- A. Download Web Document:** downloading number of pages because the work is done on a personal PC not on an internet server. This step involves downloading Web document files from different domains. Web document files are downloaded from known Yahoo! Directory and Google search engine Websites.
- B. Cleansing:** Cleansing the files by removing non-HTML files, since the implementation system works only on HTML pages.
- C. Text storing process:** storing downloaded documents text into database (HTML DB). Figure (3.2) explains the HTML text extraction process from downloaded files.

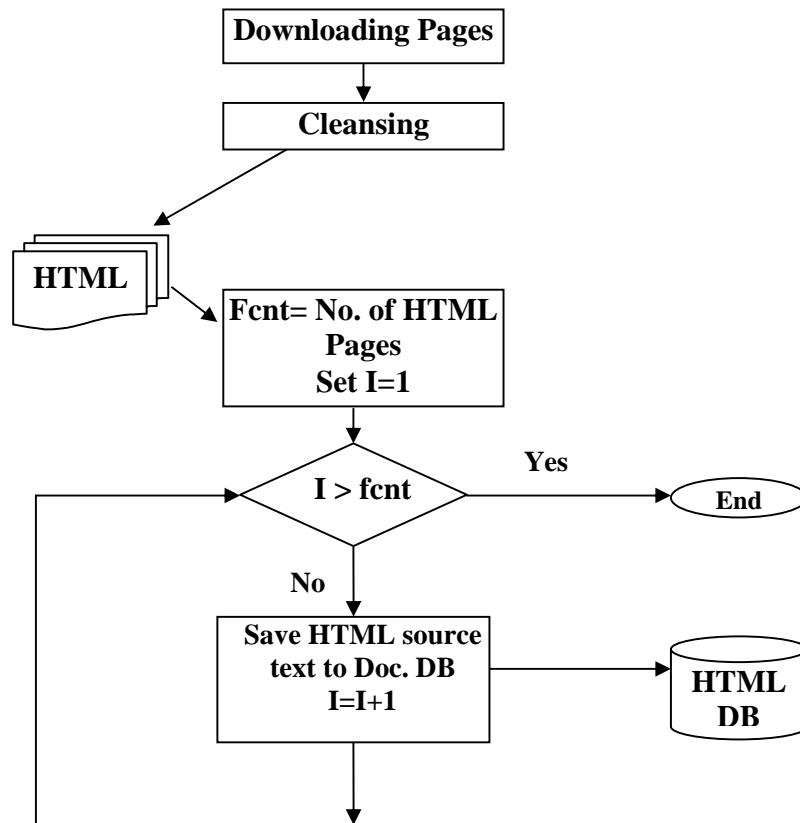


Figure (3.4): documents storage unit.

3.4.2 Document Preprocessing

Web HTML document preprocessing pass through the following processes:

- A. Lexical Text Analysis.
- B. Stop words Elimination.

A. Lexical Text Analyzer

Lexical Text Analysis is the process of converting a stream of characters (text of document) into a stream of words, to be adopted as an index terms. The major objective of lexical analysis phase is the identification of the words in the text document.

First step, there is a process of parsing, which means that HTML page is parsed to a plain text by removing HTML tags like “/”, <HTML>, <HEAD>, <TITLE>, </TITLE>, <H1>, <P>, </P>, ...etc. A tag is a string used to mark the beginning or ending of structural elements in a text. These tags richly exist in HTML source code, as mentioned in chapter two. These tags labels are now considered as a noisy data but what is referred by a tag is certainly useful data which the program looks for.

To implement that, some of algorithms are constructing before removing tags. First algorithm takes the beginning of tags and the texts that follow them and then removes ending of tags. It takes the beginning of tags because from these can specify the attributes for each text followed. The following example of HTML sentence explains that:

<TITLE> WEB CONTENT MINING </TITLE>

<TITLE> represents the beginning of tag,

</TITLE> represents the ending of tag, and,
WEB CONTENT MINING represents the text document.

Algorithm (3.1) describes the lexical text analyzer for an HTML file.

Algorithm (3.1): Read HTML file then take beginning of tags with text and remove ending of tags.
Input: HTML file. Output: list of all file words with its attributes.
Step1: read character from file in sequence Step2: check if character = '<' then repeat put character in buffer of characters read character from file in sequence until character = '</' Call algorithm (3.2) (buffer of characters as input, list of words with it's attributes as output). Step3: Check if character = '</' then Repeat read character from file in sequence Until character = '>' Step4: Repeat step1, step2, step3 until end of HTML file.

Second algorithm separates tags from text, put tag in Token Tags Buffer and put text in Token Text Buffer. After that sends Token Tags Buffer to algorithm (3.3) to discover the attributes for each word and sent

Token Text Buffer to algorithm (3.4) to convert stream of characters to stream of words. Then store all words with their attributes. Algorithm (3.2) describes the above process.

Algorithm (3.2): Separate tags from text.
Input: buffer of characters Output: list of words with its attributes.
<p>Step1: Repeat Read character from buffer and put it in tags token Until character = '>'</p> <p>Step2: Call algorithm (3.3) (tags token as input, list of attributes as output).</p> <p>Step3: Read character and check if it is '<' then repeat step1, step2 Else repeat Put character in text token Read character from buffer Until no character in buffer</p> <p>Step5: Call algorithm (3.4) (text token as input, list of words as output).</p> <p>Step6: Save list of words from step5 with it's attributes from step2 in list.</p>

Algorithm (3.3) takes Token Tags Buffer and discovers the attributes of words from tags. From these attributes, weights of words will be computed to construct documents vectors.

Algorithm (3.3): Gives the text attributes.
Input: token of tags Output: attributes of words
Step1: Set Flag_Size = 0 Set Flag_Bold = 0 Set Flag_Italic = 0
Step2: Case 1: Check If token = 'title' or 'TITLE' then Flag-Size value = 13 Case 2: Check If token = 'subtitle' or 'SUBTITLE' then Flag-Size value = 9 Case 3: Check If token = 'b' or 'B' or 'strong' or 'STRONG' or 'BLINK' or 'blink' then Flag-Bold value = 1 Case 4: Check If token = 'i' or 'i' or 'u' or 'U' or 'em' or 'EM' then Flag-Italic value = 1 Case 5: Check If token = 'big' or 'BIG' then Flag-Size value = 5 Case 6: Check If token = 'h1' or 'H1' then Flag-Size value = 10 Case 7: Check If token = 'h2' or 'H2' then Flag-Size value = 8 Case 8: Check If token = 'h3' or 'H3' then Flag-Size value = 7 Case 9: Check If token = 'h4' or 'H4' then Flag-Size value = 5 Case 10: Check If token = 'h5' or 'H5' or 'h6' or 'H6' then Flag-Size value = 4 Case 11: Check If token = 'font' or 'FONT' then check size and change Flag-Size by no. of size. Case 12: Check If token = 'a' or 'A' then change Flag-Size value = 7
Step2: Return attributes of words (Flag_Size, Flag_Bold and Flag_Italic).

The last step of data preprocessing is converting stream of characters to stream of words or (Terms), and ignoring the unnecessary special characters like (! , #, @, \$, %, &, *, ^, ”, ...etc), hyphens, comma's, also the words which begin with digits (0.9). The remaining words begin with letters (a ...z), (A ...Z); any other ones are ignored. Algorithm (3.4) explains the above paragraph.

Algorithm (3.4): Convert stream of character to stream of words.

Input: text token (stream of characters).

Output: list of words (stream of words).

Step1:

Repeat

Read character from text token

Check if character \notin number and character \notin special characters then put character in buffer

Check if character = ' ' or character = numbers or character = special character then delete it and append buffer in list

Until no character in text token

Step2:

Return list of words.

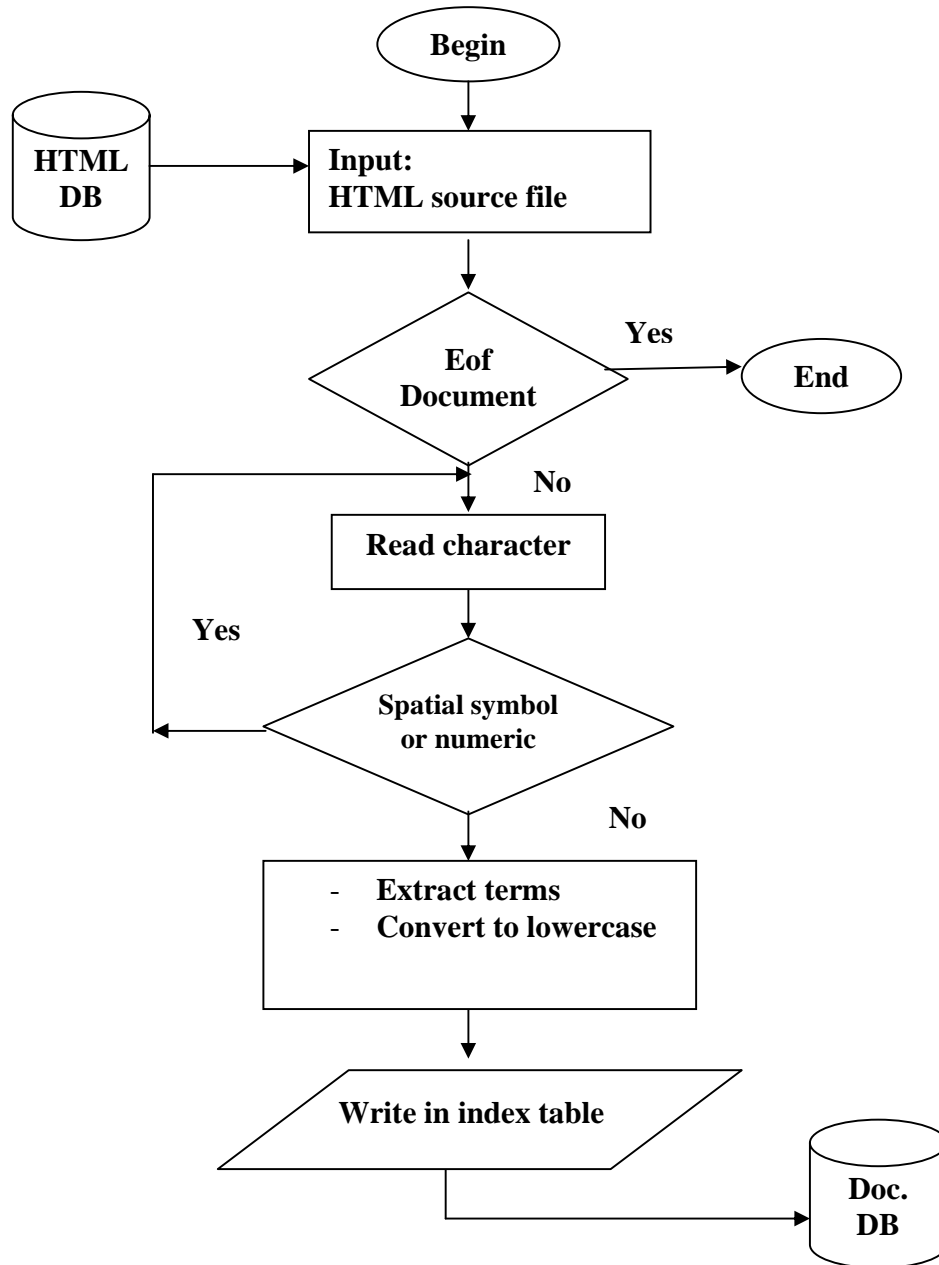


Figure (3.3): Lexical Text Analyzer Flowchart.

B. Stop Words Elimination

A stop word is a word that has little semantic content such as a preposition. It also refers to words that have a high frequency across a collection. Since a stop word appears in many documents, and not helpful for retrieval, these terms are usually removed from the internal text of a document. However, stop words could depend on context. For example, the word “Computer” would probably be a stop word in a collection of computer science newspapers articles, but not in a market list.

After initial indexing, it will be discovered that the document index contained useless terms, to decrease the number of terms in the index; it is desired to be filtered by removing stop words. A list of stop words is prepared in this research, a number of (1500) words is suggested as stop words, including the ordinary stop words similar to “the”, “which”, “is”, ...etc. Also an extracted or suggested stop words similar to “repeat”, “high”, “width”, “second”, “first”, “h1”, “h2” ..,etc. Table (3-2) contains some of the stop words and a complete list is shown on appendix A. A collection of (110) HTML documents were tested, the experiment shared that (46%) of the words are stop words; this practical example proves the need for stop words elimination process.

Figure (3.4) describes the process of eliminating stop words. The last step is converting all remaining words to lowercase.

Table (3-2): List of Stop Words.

ability	before	d	easy	good
after	been	days	f	got
all	best	description	face	greater
afterwards	between	did	fact	grouping
allow	better	dilation	feed	generally
almost	browse	differ	false	i.e.
along	back	does	far	it
among	came	done	found	it's
anybody	can	down	four	itself
anything	case	up	free	i've
as	caption	each	from	I'd
ask	can't	early	full	know
generally	cannot	e.g.	general	last
back	copy	eight	give	list
based	copyright	edu	go	least
je	language	re	search	side
jiawel	large	reader	secondary	sides
jm	largely	really	seconds	sign
jo	larger	recent	section	simple
join	last	recently	see	since
jp	later	record	seem	small
just	latest	related	seemed	site
an	latter	remainder	seeming	six
kamber	laughter	remember	seems	sixteen
hundred	made	these	thier	want
or	main	they	thing	wanted
ordered	make	they'd	things	wanting
order	man	they'll	think	wants
	many	they're	thinks	was

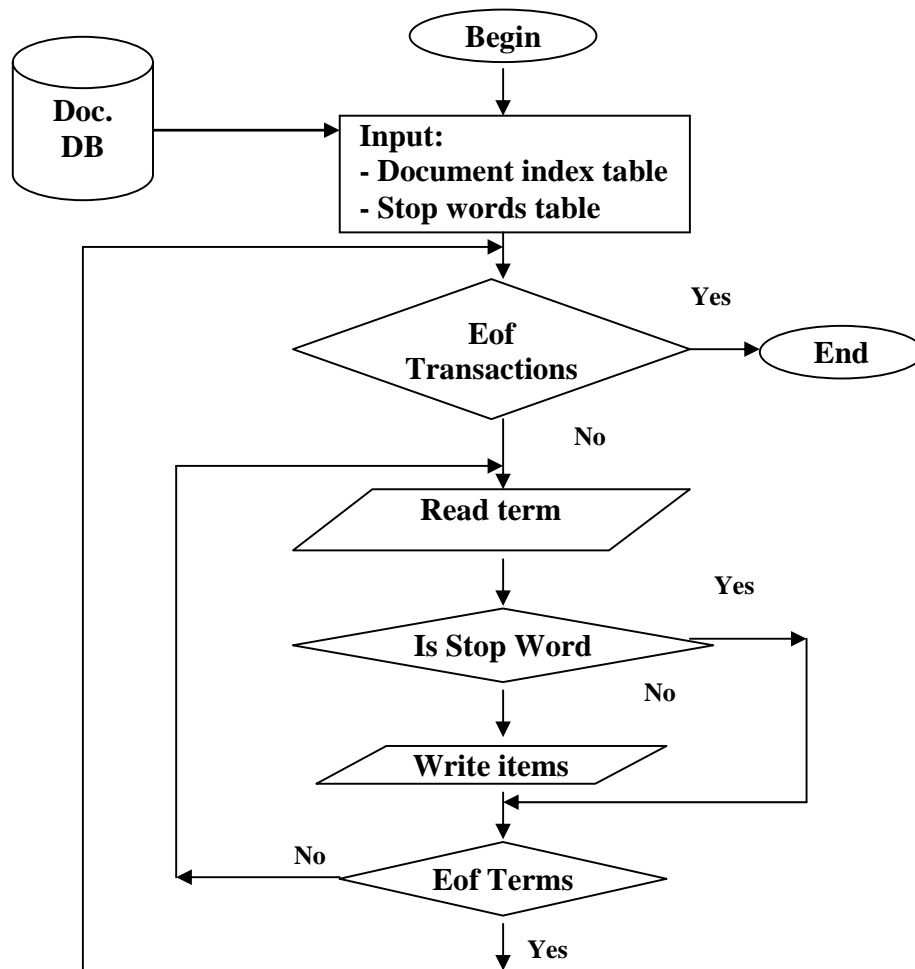


Figure (3.4): Stop Word Elimination Flowchart.

3.4.3 Documents Ranking

The Documents Ranking process involves the following steps:

- A. HTML Document feature Ranking.
- B. Html Document Ranking (HDR) Method.

3.4.3.1 HTML Document feature Ranking

Explaining the important HTML document features depends on the special structure of the language. In this research, the analysis is done to detect the influence of the HTML tags, and include text properties in the similarity consideration. As a result of this analysis, the following tags and text properties are suggested to support the document features; this must be extracted before similarity process. The useful effect of determining these features on accuracy of similarity process will be clear in chapter four. Some of these tags are:

1. HTML title — this is the text that is located within the <TITLE> tags of the HTML document.
2. HTML keywords — a text that located within the <Meta name = keywords"> tag of HTML document.
3. Link text — a text located within the anchor (<A>) tags.
4. Heading text — a text located within <H1> through <H4> tags.
5. Bold text — a text located between tags.
6. Italic text — a text located between tags.

The significant features of HTML document are extracted and added to document vector, which is designed to be used for looking up document information by significant terms. After many experiments, a weight is suggested for each feature. The tags and description with Term Ranks are shown in table (3-3).

Ranks are assigned to the important features of HTML document mentioned in table (3-3). Ranking is performed by multiplying a term by the

number of times assigned to each feature it belongs to. For example, a word in the <title> tag is entered into the index (13) times for that document. The suggested term rank is desired to be a basic values, it may be changed after evaluation process.

Table (3-3): Document Feature and its Description.

Tags (HTML Feature)	Description
B, STRONG, BLINK I, U, S, STRIKE, CODE, SAMP, VAR, EM, BLOCKQUOTE, TT, CITE, ADDRESS, SUB, SUP, KBD	Change the word style
BIG	Change the word style and size
TITLE	Change the word to a title
FONT	Change the word font style, color, or size
A	Change the word to a link
H1, H2, H3, H4, H5, H6	Change the word to a heading

Concerning the proposed document features extraction, and term rank as shown in table (3-3), it is important to discover the effectiveness of the implementation for such features. Therefore, it is desired to evaluate the suitability and effectiveness of the suggested features and weight in the similarity process. This evaluation is done by suggesting four classes of document index, see the next section.

3.4.3.2 Html Document Ranking (HDR) Method

HDR method contains four classes. The definitions of each class in HDR method for alternative representation of HTML documents are:

- 1- ALLTR, ALL document term, significant Terms Ranking - here all terms in a document are taken in consideration, with the feature of significant term ranking weights explained on section (3.4.3.1).
- 2- ALLTNR, ALL document Terms No Ranking - here all terms in document taken without the feature rank weights are taken in consideration.
- 3- SIGTR, just SIGnificant Terms Ranking - here just significant terms taken with their features weights ranking are taken, while the rest of terms is ignored.
- 4- SIGTNR, just SIGnificant Terms No Ranking - are taken here just significant terms taken without ranking weights. The rest of terms are ignored.

Figure (3.5) depicts the general process of proposed HTML Document feature Ranking (HDR) method.

Many advantages are obtained with implementation of this method, it greatly simplifies the construction of documents vectors, by ranking a term in the index, and it increases its term frequency index document weighting value, which will be explained later. The important feature values are computed with the rest of the terms in other HTML tags.

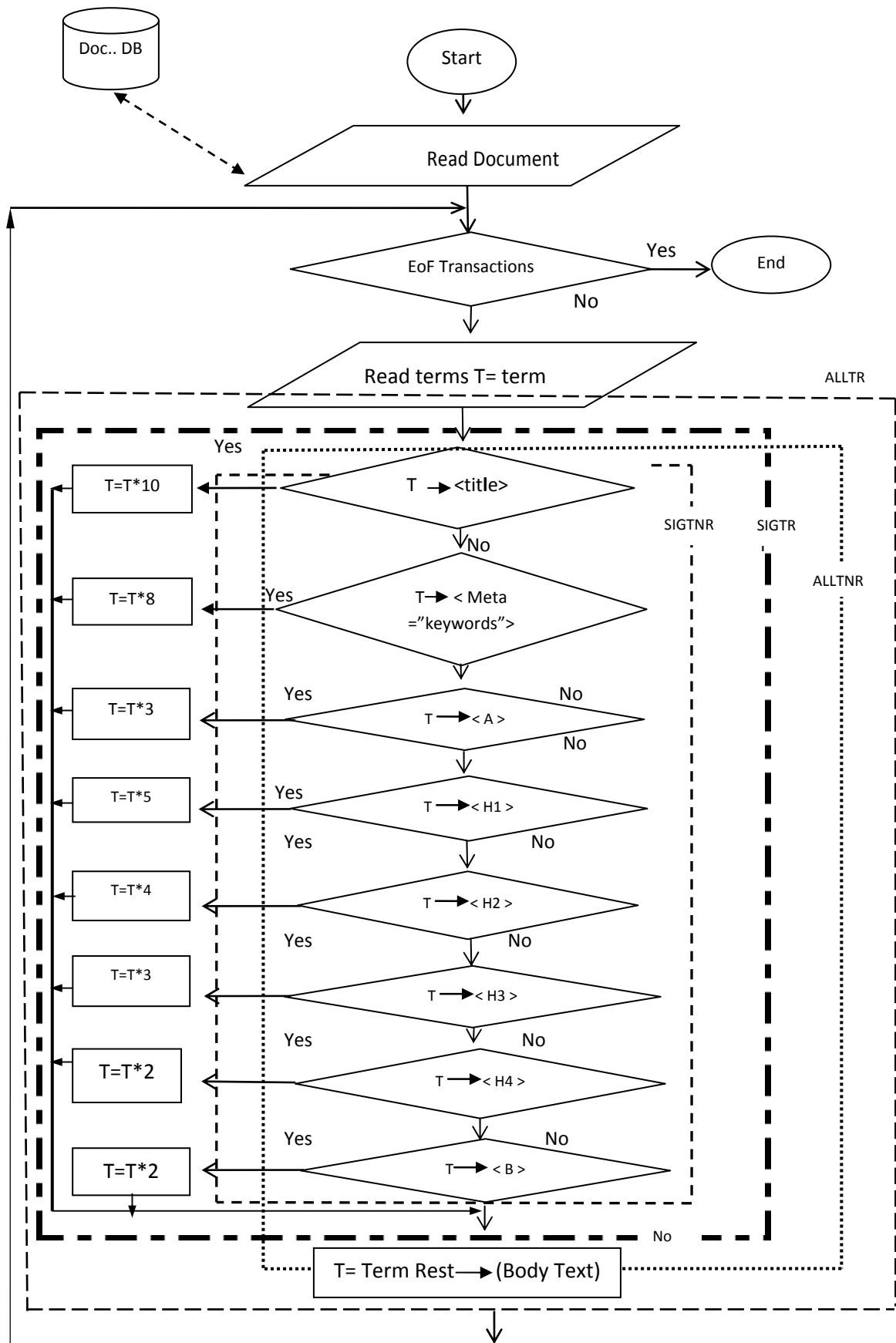


Figure (3.7) depicts the general process of proposed HTML Document features Ranking (HDR) Method

3.4.4 Term Indexing

This process involves the following steps:

- A. Term Weighting Measure.
- B. Constructing Document Vectors.

3.4.4.1 Term Weighting Measure

After HTML text has been parsed, Lexical Analysis is done, Stop Words is deleted, and Ranking of Terms is computed. Then the weight of each word is computed from the attribute which it gets from documents ranking and frequency (occurrence), to compute weights of words some processes are used.

First process involves computing the occurrence (repetition) for each term in document. Occurrence represents the number of words repetition in a document.

Algorithm (3.5) describes the process of computing repetition of all terms in each document.

Algorithm (3.5): compute the occurrence of each word in the file.
Input: list of words. Output: list of words with its occurrence.
<p>Step1: Read a word from word list.</p> <p>Step2: Check word with all words in list, increment the counter by one for each word similar to the searched one.</p> <p>Step3: Save the counter.</p> <p>Step4: Repeat step1, step2, step3 until last word in the list.</p>

Second step, the weight of each word in word list is computed. In this thesis, a formula is suggested to compute the weight of terms depending on occurrence and the attributes that it gets from Documents Ranking (see section 3.4.4). This formula is:

$$\text{Weight} = f(S, P, R, B, T) \quad (3.1)$$

where: S stands for the size of term.

R stands for the repetition (occurrences) of term.

P stands for the position of term in page.

B stands for the term bold or not.

T stands for the term italic or not.

$$\text{Weight} = \sum_{i=1}^n ((\alpha_1 * P_i + \alpha_2 * S_i + \alpha_3 * R_i + \alpha_4 * B_i + \alpha_5 * T_i) * \beta) \quad \dots (3.2)$$

where $i=1, 2, 3, \dots, n$ ($n = \text{total no. of terms}$),

α stands for a random number between [0, 1], the summation of all α is one i.e. ($\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 + \alpha_5 = 1$) and

$$\alpha_1 > \alpha_2 ,$$

$$\alpha_2 > \alpha_3 ,$$

$$\alpha_3 > \alpha_4 ,$$

$$\alpha_4 > \alpha_5 .$$

α gives an importance and priority to the term which it belongs to in order to increase the weight of that term.

$$\beta = \begin{pmatrix} 1 & \text{New term} \\ 0 & \text{Garbage} \end{pmatrix} \quad (3.3)$$

Each parameter in equation (3.2) is computed according to an equation related to it. The details of these computations are:

1. Position can be obtained from equation (3.4).

$$P = \frac{\text{No. of total lines in page} - \text{line No. includes word}}{\text{No. of total lines}} \quad (3.4)$$

2. Size can be obtained from equation (3.5).

$$S = \frac{\text{Size of word} - \text{Minimum existing size}}{\text{Maximum existing size} - \text{Minimum existing size}} \quad (3.5).$$

3. Repetition can be obtained from equation (3.6).

$$R = \frac{\text{No. of word occurrence}}{\text{total of words}} \quad (3.6).$$

4. Bold is either [0, 1]

If B=1 then word is Bold else it's not.

5. Italic is either [0, 1]

If T=1 then word is Italic else it's not.

The computed weights will be used to construct words' vector (Documents' Vectors), these weights represent membership between words and its document.

3.4.4.2. Constructing Document Vectors

First step in constructing the documents' vectors is to put all "useful" terms from the index into a database table. Our original goal is to use all the terms that represent the document clearly as a keywords for the document, for the document vector construction, but even with the stop word elimination as described above, the index is still filled with useless terms, misspellings, and garbage. Three heuristics have been taken in order to decrease the length of our document vectors to a manageable size. The first is that it only uses terms that match the regular expression [a...z], that is, all terms that start with letters. The second heuristic is removing stop words and then converts all the terms or document words to lowercases.

The third heuristic is to only include terms that are found in frequency above a threshold of document words. By experiment, (10 %) of the total number of text document terms, are the best setting of lower bound, a regular number of terms, or keywords represent a document vector. Third heuristic involves a step to reduce the number of terms.

Because not all terms in document are important and not all have high weights or high priority, therefore, small number of terms are used as keywords point to document's domain. In this thesis, only few ratios from total number of highest weights terms of the document are used to represent

the keywords of it. Document Vector will be constructed from these keywords.

After reducing the number of terms and taking the highest weights for all documents, the words' vectors (Documents' Vectors) will be built and stored in a DB as declared in algorithm (3.6).

Algorithm (3.6): Collect each HTML files or pages in one database.

Input: HTML files.

Output: database of all terms and its weights.

Step1:

Read file from database of HTML files

Step2:

Call algorithm (3.1)

Call algorithm (3.5)

Compute weights of all words.

Specify the keywords from words' list.

Build document vector by using keywords.

Step3:

Repeat step1, step2 for all HTML file.

3.4.5 Similarity

Similarity process involves the following steps:

1. Construction of Query Document Vector.
2. Similarity without fuzzy.
3. Similarity with fuzzy.
4. Construction Query-Document Similarity Vector (membership vector).

3.4.5.1. Construction of Query Document Vector:

The query page passes through all processes that DB pages went through, so that, a document vector could be built for query page similar to the stored one in DB. This query vector could be used later for similarity process.

1. Cleansing the file by checking if it is an HTML file then go to the next step, else gives an error message “Enter HTML file”.
2. Lexical Analyzer: removes tags and converts stream of characters to a list of words (see figure (3.3)).
3. Elimination of Stop Words, special characters and all words started with digits. Then takes all words with (a...z) or (A...Z). (See figure (3.4)).
4. Document Ranking: extracting features from tags and apply (HDR) Method.
5. Apply weighting formula (3.2) to compute the weight for each word.
6. Sort words in descending order according to their weights and then take ratio of highest weight from total number of words.

7. Construct Query Vector from remaining words (Terms) with its weights.

3.4.5.2. Normal Similarity

In vector space, the documents are ranked according to their degrees of relevance to the query. One way to compute the degree of relevance is to calculate the similarity of the query \mathbf{q} to each document \mathbf{d}_j in the document collection D . There are many similarity measures. The most well known one is the **Cosine Similarity** (mentioned in chapter two), which is the cosine of the angle between the query vector \mathbf{q} and the document vector \mathbf{d}_j (see chapter two). The formula of cosine similarity is:

$$\text{cosine}(\mathbf{d}_j, \mathbf{q}) = \frac{\langle \mathbf{d}_j \bullet \mathbf{q} \rangle}{\|\mathbf{d}_j\| \times \|\mathbf{q}\|} = \frac{\sum_{i=1}^{|V|} w_{ij} \times w_{iq}}{\sqrt{\sum_{i=1}^{|V|} w_{ij}^2} \times \sqrt{\sum_{i=1}^{|V|} w_{iq}^2}} \dots\dots\dots (3.9)$$

where:

w_{iq} represents the term weight of each term t_i in \mathbf{q}

w_{ij} represents the term weight of each term t_i in \mathbf{d}_j

$|V|$ represents the vocabulary size of the documents collection.

The result of this step is the Query-Document Vector. This vector represents the membership between the query document and documents collection (Document DB)

3.4.5.3 Fuzzy Similarity

Similarity with fuzzy is used to make the system more flexible and gives efficient results than normal Web mining. In this thesis, an equation is suggested to compute the membership (fitness) depending on fuzzy logic rule. Figure (3.6) shows the membership scale between query weights and documents' weights in fuzzy logic. The formula of fuzzy Web page similarity mining can be seen in formulas (3.10) and (3.11).

$$\mu (i) = \left\{ \begin{array}{ll} \frac{W_q (i)}{W_d (i)} & \text{If } W_q (i) < W_d (i) \\ 1 & \text{If } W_q (i) \geq W_d (i) \end{array} \right\} \dots (3.10)$$

$$\text{Fuzzy} = 1/n \sum_{i=1}^{|v|} \mu (i) \dots\dots(3.11)$$

where:

W_q represents the weight of term in a query vector.

W_d represents the weight of term in a document vector.

v represents the number of terms in a vector.

n represents the number of documents in the document DB.

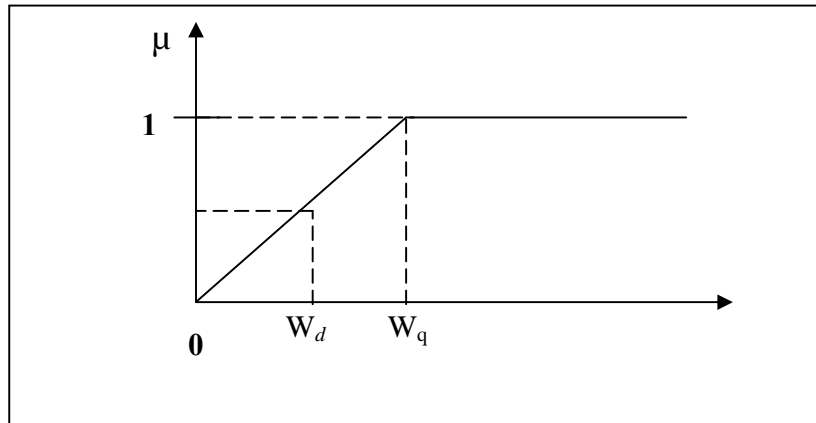
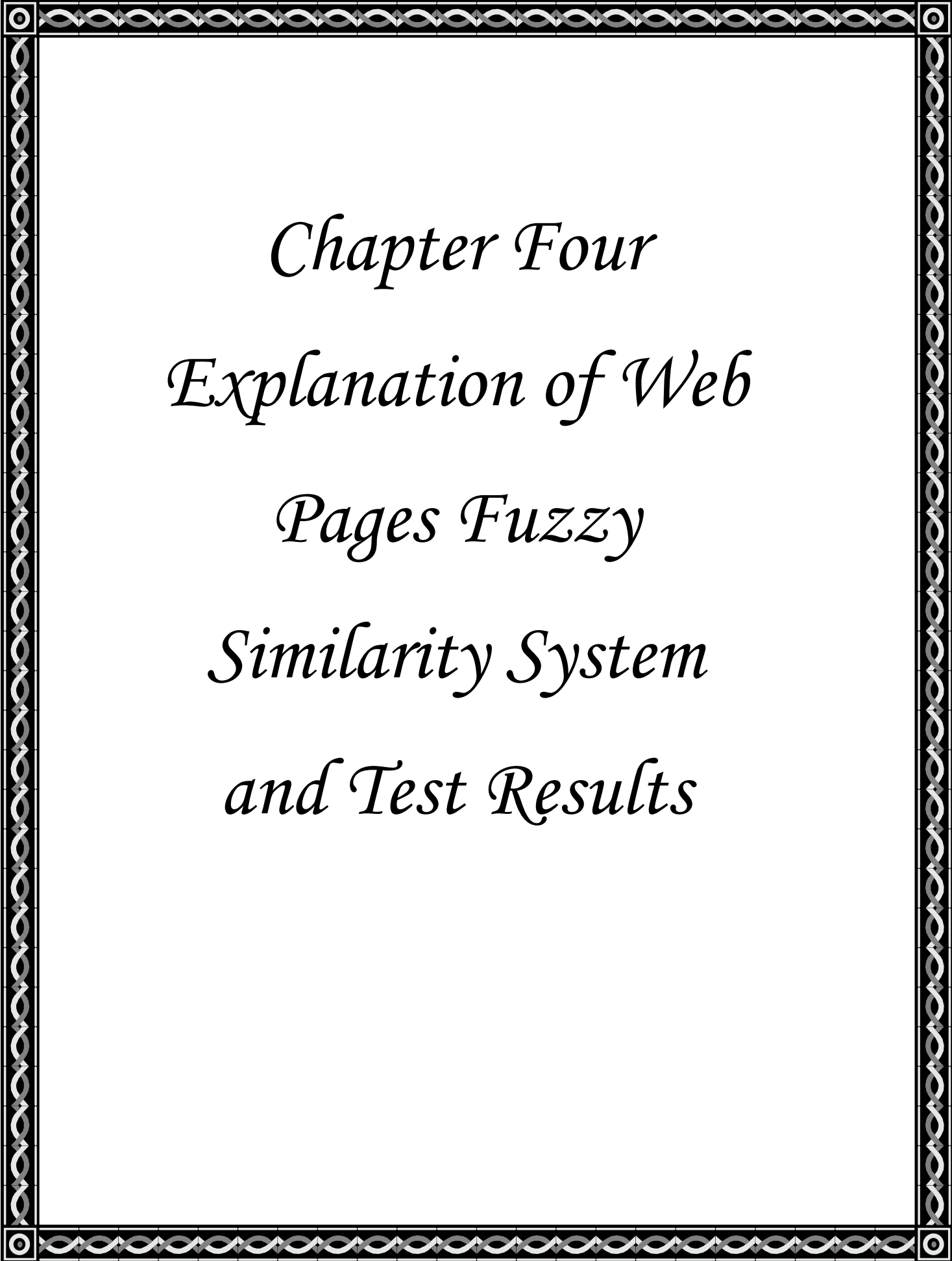


Figure (3.6): scale of similarity using fuzzy rule.

After performing similarity process either a similar page(s) can be found and displayed to the user, or no similar page is found, and in this case, the system can suggest adding the query vector to the set of documents DB if the user wishes that.



Chapter Four
Explanation of Web
Pages Fuzzy
Similarity System
and Test Results

Chapter Four

Explanation of Web Pages

Fuzzy Similarity System and Test

Results

4.1 Introductions

In this chapter the implementation the Web pages fuzzy similarity mining is execution. The Off-line phase is management by administrator only; some results of analyzer documents are explained in tables.

On-line phase represents the search about pages. When enter page query and click on search button, the system takes query and analyze it. Then done similarity first without fuzzy and second with fuzzy, and display the two results, if its find page(s) similar the query. If not find any page, the system displays three options: first ignore the search, second retry the search, third adding query page to DB. In this case, the user must enter the password and user name if wish adding query page.

Microsoft Visual Basic 6.0 is used to execute this system, Microsoft Office access is used to store DB, and HTML language.

4.2 HTML Document Collections

HTML documents are collected by downloading from Web resources, for determined categories, for example “data mining”, “fuzzy logic”, “neural networks”, “geography”, “history”, “sports”. In order to build the document sets, Web HTML files are collected using Yahoo’s Directory and Google search engine Websites. The number of HTML Web documents are nearly (200) documents.

As declared in chapter two and three, HTML Web documents are built with a special format including texts and image, hyperlinks, tags, special characters,....., etc. for more explanation, a sample of HTML view as obtained from Web browser and sample of its source code, show the tags, special structure and links, noisy data, which are major source data to be processed. Figure (4.1) depicts a Web page and (4.2) shows its source code. To understand this example see simple example in appendix B.

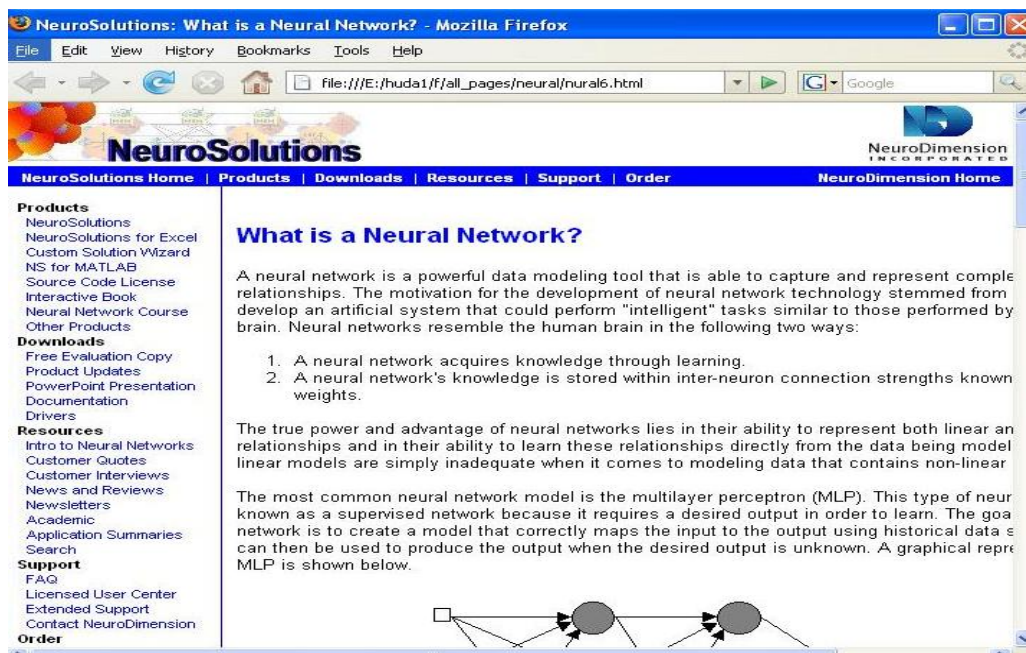


Figure (4.1): HTML Document as obtained from Web Browser.

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<!-- saved from url=(0038)http://www.nd.com/welcome/whatisnn.htm -->
<HTML><HEAD><TITLE>NeuroSolutions: What is a Neural Network?</TITLE>
<META http-equiv=Content-Type content="text/html; charset=windows-
1256"><LINK
href="nural6_files/style.css" type=text/css rel=stylesheet>
<META content="MSHTML 6.00.2900.3199" name=GENERATOR></HEAD>
<BODY leftMargin=0 topMargin=0 MARGINWIDTH="0" MARGINHEIGHT="0"><!--Start
Top SSI -->
<TABLE cellSpacing=0 cellPadding=0 width="100%" border=0>
  <TBODY>
    <TR>
      <TD><IMG height=60 src="nural6_files/NS_Busy_Blue.jpg" width=250>
</TD>
      <TD vAlign=bottom align=right>
        <TABLE cellSpacing=0 cellPadding=0 border=0>
          <TBODY>
            <TR>
              <TD align=middle><A href="http://www.nd.com/"><IMG
                src="nural6_files/ND_Spin.gif" align=center border=0></A>
</TD></TR>
            <TR>
              <TD align=middle><A href="http://www.nd.com/"><IMG
                src="nural6_files/Logo_NDI.jpg" align=center border=0></A>
</TD></TR></TBODY></TABLE></TD></TR></TBODY></TABLE>
<TABLE cellSpacing=0 cellPadding=0 width="100%" border=0>
  <TBODY>
    <TR>
      <TD bgColor=#0003fc>&nbsp;&nbsp;&nbsp;<A class=NavTop
        href="http://www.nd.com/">NeuroSolutions Home</A> <A
        class=NavTop>&nbsp;&nbsp;&nbsp;</A> <A class=NavTop
        href="http://www.neurosolutions.com/products/">Products</A> <A
        class=NavTop>&nbsp;&nbsp;&nbsp;</A> <A class=NavTop
        href="http://www.neurosolutions.com/downloads/">Downloads</A>

```

Figure (4.2): Sample of HTML Document Source Code.

Table (4-1) illustrates a sample of an HTML document noise characters and there frequency in descending order. Table (4-2) illustrates an example of HTML terms after noise deleting.

Table (4-1): Sample of Noise Terms after HTML Document Parsing.

Document ID	Character	Frequency
DM15.HTML	>	423
DM15.HTML	<	395
DM15.HTML	.	319
DM15.HTML	,	246
DM16.HTML	;	246
DM15.HTML	/	214
DM15.HTML	:	109
DM16.HTML	&	109
DM15.HTML	;	108
DM15.HTML	=	102
DM15.HTML	-	101
DM15.HTML	‘	101
DM16.HTML	“	87
DM15.HTML	“	52
DM15.HTML	&	41
DM14.HTML	_	32
DM15.HTML	0	28
DM16.HTML	#	20
DM14.HTML)	19
DM15.HTML	1	18

Table (4-2): example of HTML terms after noisy data removal.

Term	Term
HTML	page
head	title
meta	head
title	body
introduction	networks
the	bgcolor
top	ffffff
ten	link
steps	vlink
to	alink
perfect	page
neural	title

4.3 Document Preprocessing Experiments

As explained in chapter three, document preprocessing includes Lexical Text Analysis, Elimination Stop Words and useless words.

4.3.1 Lexical Text Analyzer Experiment

On lexical text analysis, a stream of document characters is converted into a stream of terms to represent a document. An example is presented to make a random collection of (225) HTML Web documents to test the percentage of characters in it. It contains total of (5,150,884) characters. Any of HTML tags simple and other non-alphabetic terms removed which are considered as noises, a total of (1,264,762) character were noise, the table below shows the percentage of noise characters to alphabetic characters is (24.5 %) that will be a conclusion of the importance of cleaning the text documents from noise. Table (4-3) shows the test results.

Table (4-3): Noise Percentage to Total HTML Terms.

HTML Documents (225)	Characters Total	Char. per Documents	Percentage to Total
Characters of 200 HTML text	5150884	22893	100%
Noise	1264762	5621	24.5%
Char. after Noise Cleaning	3886122	17272	75.5%

4.3.2 Elimination of Stop Words and Useless Words

For examining the effects of stop words being resident within the term corpus, stop words elimination is done for the same experiment documents used. The result illustrates that after HTML tag and noise cleaning and removing non alphabetic terms. These documents contain a total of (1,921,080) words, (1,056,920) are stop words with percentage of 55%. The same conclusion can be said about stop words elimination advantages, for the reason of text mining, keeping the useful terms for the task. Table (4-4), illustrates the test values result after stop words elimination.

Table (4-4): Test values results after stop words elimination.

HTML Documents (225)	Term Total	Term per Documents	Percentage to Total
Term after Noise Cleaning	1921080	8538	100%
Stop Words	1056920	4697	55%
Term after Stop Words Elimination	864160	3838	45%

Some examples of documents with total numbers of terms, total number of stop words and the percentage to words can be shows in table (4-5).

Table (4-5): Documents with percentage of Stop Words.

No.	Total No. of Terms in Doc.	No. of Stop Words	Percentage to Words
1	4334	2299	54%
2	3448	1800	53%
3	2824	1470	53%
4	2804	1559	56%
5	2551	1344	53%
6	1613	950	60%
7	1450	722	50%
8	1249	713	58%
9	1088	630	58%
10	1036	593	57%
11	774	420	54%
12	638	362	56%
13	239	120	51%
14	147	80	54%

Table (4-6) shows a document's terms before remove stop words and duplication, while table (4-7) shows the same document's terms after stop words eliminations.

Table (4-6): Document's terms before remove Stop Words.

Term	Term	Term	Term
ability	as	data	nbsp
ability	as	data	nbsp
able	backpropagation	data	nbsp
academic	backpropagation	data	nd
acquires	be	desire	network
advantage	because	desired	network
again	being	desired	network
algorithm	below	develop	network
an	block	development	network
an	book	diagram	network
an	both	directly	network
and	brain	documentation	network
and	brain	downloads	network
and	by	downloads	network
and	by	drivers	network
and	by	evaluation	networks
and	by	excel	networks
and	by	extended	networks
and	by	faq	networks
application	called	fed	neural
are	can	finally	neural
are	capture	first	neural
are	center	first	neural
artificial	characteristics	first	neural
as	code	following	neural
as	com	for	neural
as	comes	for	neural
as	common	for	neural
backpropagation	complex	free	neural

Table (4-7): Document’s terms after removing Stop Words.

Term	Term	Term
academic	attaching	models
academic	automatically	motivation
acceptable	automatically	motivation
acquires	automatically	movement
acquires	automatically	mr
adjust	backpropagated	multilayer
adjust	backpropagation	multiplayer
ads	backpropagation	multiplied
advantage	banner	multiplied
advantage	behavior	multiplied
algorithm	binary	nd
allocate	binary	nd
analyze	brain	nd
analyzing	brain	nd
appears	brain	network
application	brain	network
application	breaks	network
application	broad	network
application	called	network
application	camera	network
applications	campaign	network
applied	capture	network
artificial	capture	network
artificial	card	network
ascii	character	network
ascii	character	network
assets	character	network
assigning	character	network
assisting	character	network
assisting	neural	neural
asss	neural	neural

4.4 Document Feature Extraction Experiments

Chapter three explains the proposed (HDR) method, which is responsible for features extraction. In order to implement (HDR) method, experiments were done on the training document set, for proposed classes of term ranking: Significant Term Ranking, All Document Term No Ranking, Jest Significant Terms Ranking, and Jest Significant Terms No Ranking. Also to compute the terms weight, formula (3.2) will be used, which depends on the frequency of the word in the page (occurrence) and the attributes of the terms such as (font size, font style, position of the word in the pages, title, header, and link text).

$$Weight = \sum_{i=1}^n ((\alpha_1 * P_i + \alpha_2 * S_i + \alpha_3 * R_i + \alpha_4 * B_i + \alpha_5 * T_i) * \beta)$$

Size (title, header,..)	position	Frequency	Bold	Italic
--------------------------------	-----------------	------------------	-------------	---------------

After calculating the weight for each term, delete duplication term and take only largest weight from duplicate term. The percentage of total removing duplication is nearly 50% from total number of terms in documents. This step reduces the number of terms which represent the document keywords. Table (4-8) shows the terms after removing duplications and useless words, and table (4-9) shows the percentage of duplication in same document.

Table (4-8): Document Terms after removed Duplications and useless words.

Term	Term
neural	presentation
neurosolutions	documentation
products	drivers
network	academic
interactive	application
license	support
code	extended
source	information
matlab	data
evaluation	development

Table (4-9): Documents Term with percentage of Duplication.

Doc. No.	No. of Terms in Doc.	No. of terms after delete duplication	Percentage to Words
1	2035	1095	52%
2	1648	857	52%
3	1354	802	45%
4	1245	661	57%
5	1207	654	51%
6	663	523	45%
7	728	364	50%
8	536	281	60%
9	458	214	66%
10	443	316	47%
11	354	250	41%
12	276	210	42%
13	119	81	33%
14	67	55	32%

4.5 DataBase Construction

To construct documents DB, Documents Vectors must be built from the terms and their weights after taking a ratio of largest weights from each document to represent the KeyWords of that document, which is used later to compare similarity between them. Table (4-10) shows a document vector sorted in descending order.

Table (4-10): Document Vector.

Term	Weight
neural	0.910
network	0.892
neurosolutions	0.776
neurodimension	0.739
data	0.663
process	0.521
products	0.489
machine	0.421
code	0.407
interactive	0.336
intelligent	0.321

4.6 Searching Similar Pages Experiments

The On-Line phase of this system performs a search operation for similar pages. The similarity search is done in two ways; first by using normal Web Mining and second by using Fuzzy Web Mining. Figure (4.3) shows the main form of Web Pages Similarity system and figure (4.4) shows the search form.

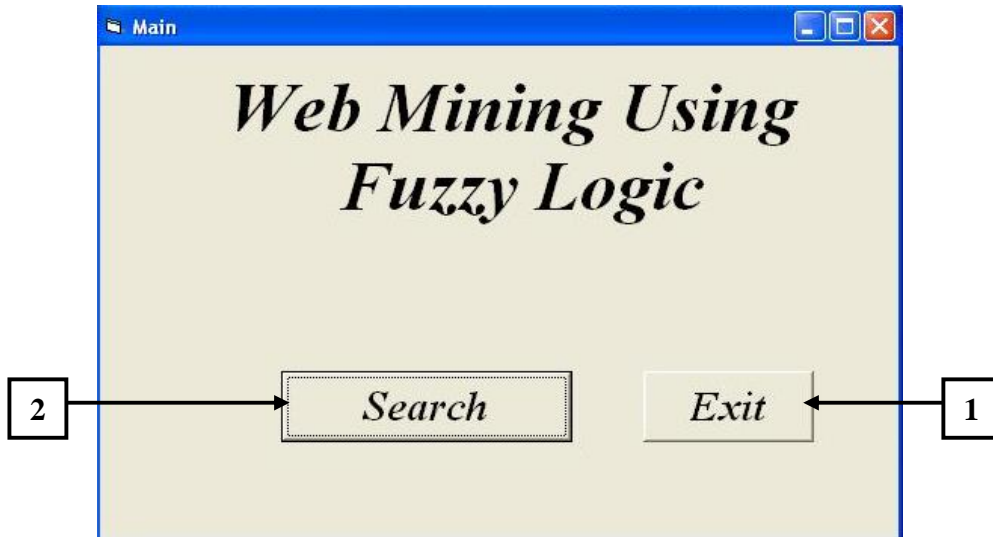


Figure (4.3): the main form.

1. *Exit*: ends the program execution.
2. *Search*: the search page will be display in the search form as show in figure (4.4).

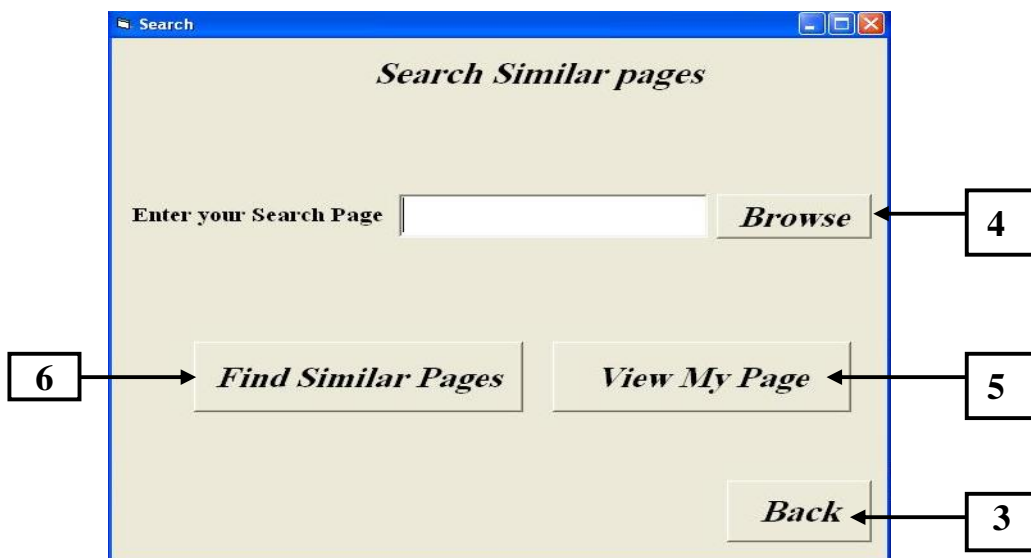


Figure (4.4): search main form.

3. **Back:** return to main form.
4. **Browse:** by this button a user can select one page from a database to find the similar pages to it because the program is executed on a PC computer.
5. **View My Page:** display the page that will be chosen to search similar pages to it.
6. **Find similar pages:** After entering the query page and click Find Similar Pages button, the system will check if the page is an HTML or not, if yes then it will perform the following tasks:

A. Construction Query Vector: such as previously.

B. Normal Web Mining: this step uses the most commonly used equation, Cosine Similarity measure to compute the membership between documents vectors (Documents DB) and Query document vector. The value of membership vector range between [0, 1], if the value is zero then no membership between them, i.e., no similar pages. After that, it displays the results of membership sorted in descending order, see table (4-11), or displaying figure (4.5) if no membership between query vector and documents DB.

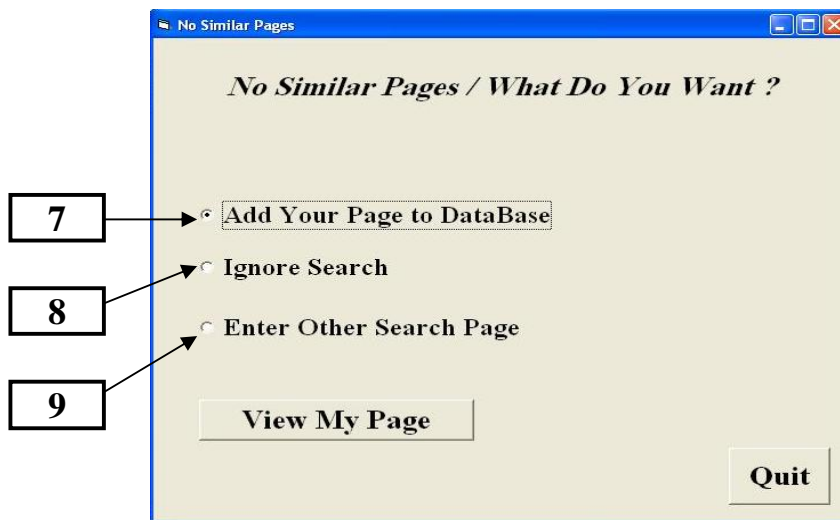


Figure (4.5): no similar page form.

7. **Add your Page to Database:** this option will be displaying sign in form see figure (4.6)
8. **Ignore Search:** ignore searching about page and return to displaying the main form.
9. **Enter Other Search Page:** displaying the search form to enter another page and found similar pages to it, see figure (4.4).
10. **Add New Document:** adding new HTML file to the document database and update it. This option required password and user name in order to update DB because only authorized persons can add a new document to DB.



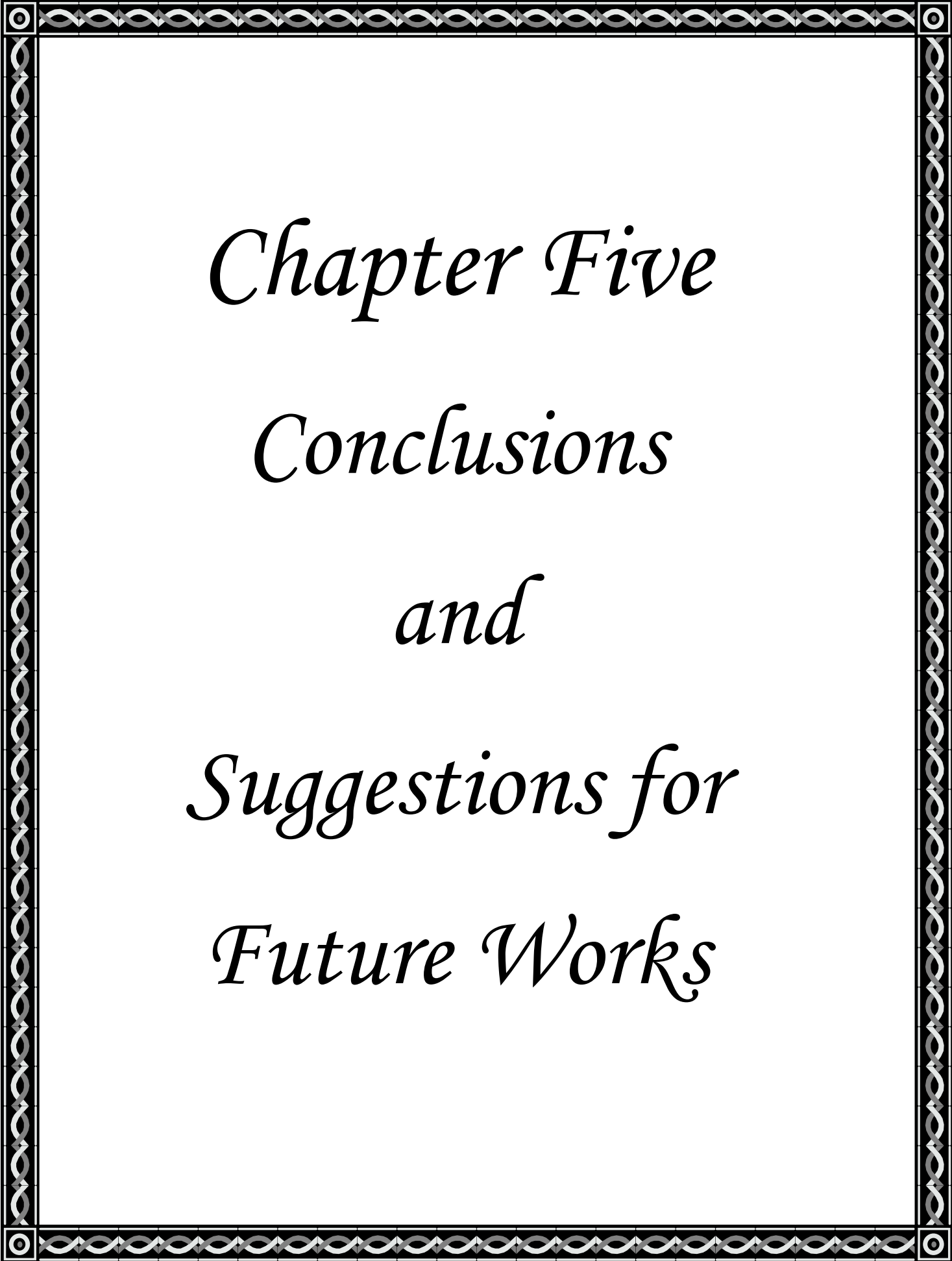
Figure (4.6): Add new document.

This program gives the users ability to compare between results first by using normal Web mining and second by using fuzzy Web mining, see table (4-11). Using fuzzy mining make the database more flexible and improves the efficiency of similarity results because the results of similar pages are more nearly to the query page. Table (4-11) shows the difference between two results.

C. Fuzzy Web Mining: this step uses a fuzzy formula maintained previously to compute the fitness between documents DB and query document. The value of fitness range between [0, 1]. The results of fitness are displayed in descending order as shown in table (4-11).

Table (4-11): the Membership and Fitness between DB and query document.

Without Fuzzy Similarity		With Fuzzy Similarity	
Doc. ID	Memberships	Doc. ID	Fitness
DM5	1	DM10	1
DM11	0.717779	DM2	1
DM12	0.6880519	DM5	1
DM13	0.6660858	DM6	1
DM9	0.6444142	DM15	1
DM3	0.6240057	DM14	1
DM4	0.6155047	DM13	1
DM14	0.6031903	DM8	1
DM15	0.6002548	DM9	1
DM2	0.5949122	DM3	0.998847
DM10	0.5946029	DM4	0.9978946
DM7	0.5817119	DM7	0.9978259
DM1	0.5467456	DM12	0.9940159
DM6	0.5423400	DM1	0.99234
DM8	0.5418300	DM11	0.98892



Chapter Five

Conclusions

and

Suggestions for

Future Works

Chapter Five

Conclusions and Suggestions for Future Works

5.1 Conclusions

The conclusions that can be drawn from this work are listed as follows:

1. The performed search operation using page contents instead of keywords yields to retrieve less number of pages, more accurate to find similar pages, and this will be more useful for the users to decrease the time for finding the desired information.
2. The suggested formula which was used to compute the words' weights depends on the words' frequency and words' attributes such as (font style, font size, position of the words in a document, text hyper link, title, and header). This makes the weights more precise to represent the words.
3. The use of HDR method with its first class (ALLTR class) gives the words useful and efficient weights because it takes all words' attributes in computing word's weight and it gives the word an importance in the document better than ignoring part of word's attributes.

4. The suggested fuzzy formula takes in consideration all the word's attributes so, the resulted weight gives a number which represents the relationship between the word and its document. The word's weights of the query page also computed in the same way. The results of similarity from using fuzzy equation is better than normal equation (cosine similarity) and this was obvious in the results, because in the fuzzy results many pages were retrieved and with high order which are related to query page and did not appear, or appears with low order in cosine results.

5.2 Suggestions for Future Works

There are several ideas for developing the proposed Web Pages Fuzzy Similarly system such as:

1. Develop ranking algorithm by using another formula to compute weights of words by using META tags (Description and Keywords) rather than using BODY and TITLE tags to enhance information retrieving process.
2. Using another fuzzy number logic function to improve the system results.
3. Using other intelligent techniques such as neural network and genetic algorithm, to check the efficiency of using intelligent algorithms with Web mining.
4. Other Web documents types can be used, such as XML Web documents which their use increased rapidly.



References

References

- [Ant05] Anthony Scime, “**Web Mining Applications and Techniques**”, State University of New York College at Brockport USA, Idea Group Reference, 2005.
- [Bin05] Bing Liu, Kevin Chen-Chuan Chang, ”**Editorial: Special Issue on Web Content Mining**”, SIGKDD Explorations, Vol. 6, PP. 1-4, 2005.
- [Bin07] Bing Liu, “**Web Data Mining**”, Springer, 2007.
- [Bus06] Bushra K. Al-Saidi, “**A Proposed Genetic Algorithm for Clustering Web Search Engine Results**”, Ph.D. thesis, Iraqi Commission for Computers and Informatics, Informatics Institute for Postgraduate Studies, 2006.
- [Chr08] Christopher D. Manning, Prabhakar Raghavan & Hinrich Schütze, “**An Introduction to Information Retrieval**”, Cambridge University Press, 2008.
- [Coo97] Cooley R., Mobasher B., and Srivastava J., “**Web Mining: Information and Pattern Discovery on the World Wide Web**”, Department of Computer Science and Engineering, University of Minnesota, U.S.A., 1997.
- [Cor79] Cornelis Joost van Rijsbergen, “**Information Retrieval**”, the University of Glasgow, 1979.

- [Dan06] Daniel T. Larose, **“Data Mining Methods and Modules”**, Department of Mathematical Sciences Central Connecticut State University, WILEY, 2006.
- [Dan98] Daniela Florescu, Alon Levy, Alberto Mendelzn **“Database Techniques for the World Wide Web: A Srvey”**, University of Washington, 2005.
- [Dra04] Dragos Arotaritei, Sushmita Mitra, **“Web Mining: asurvey in the fuzzy framework”**, ELSEVIER, PP. 5-19, 2004.
- [Eti96] O. Etioini, **“The World Wide Web: Quagmire or gold mine”**, Communications of the ACM, Vol. 39, No. 11, 1996.
- [Gay03] Gaya Buddhinath, **“Knowledge Discovery on the Web”**, Department of Computer Science and Software Engineering University of Melbourne, 2003.
- [Her04] Herb Edelstein, **“Building Profitable Customer Relationships with Data Mining”**, Two Crows Corporation, 2004.
- [Her99] Herbert A. Edelstein **“Introduction to Data Mining and Knowledge Discovery”**, Two Crows Corporation, 1999.
- [Jam05] James J. Buckley, **“Fuzzy Probabilities New Approach and Applications”**, Springer, 2005.
- [Jao00] Joachim Herbst, **“A Machine Learning Approach to Workflow Management”**, Springer-Verlag, Vol. 1810, PP. 183—194, 2000.
- [Jia00] Jiwei Han, Micheline Kamber, **“Data Mining: Concept and Techniques”**, New Jersey London Simon Fraser University, Morgan Kaufmann Publishers, 2000.

-
- [Jia02] Jiawei Han Kevin, Chen Chuan Chang, “**Data Mining for Web Intelligence**”, University of Illinois at Urbana-Champaign, IEEE, 2002.
- [Joh01] John R. Punin, Mukkai S. Krishnamoorthy, Mohammed J. Zaki, “**Web Usage Mining: Languages and Algorithm**”, Computer Science Department Rensselaer Polytechnic Institute, Troy NY 12180, 2001.
- [Joh06] John Wang, “**Encyclopedia of Data Warehousing and Mining**”, the United States of America by Idea Group Reference, 2006.
- [Kha03] Khaki A. Sedigh, Mehdi Roudaki, “**Identification of the Dynamic of Google Ranking Algorithm**”, University of Technology Department of Electrical Engineering, K.N, 2003.
- [Lee05] Kwang H. Lee, “**First Course on Fuzzy Theory and Applications**”, Springer, 2005.
- [Mag03] Magdalini Eirinaki, Michalis Vazirgiannis, “**Web Mining for Web Personalization**”, Athens University of Economics and Business, ACM, Vol. 3, No 1, PP. 1 – 27, 2003.
- [Mag05] Magdalini Eirinaki, “**Web mining: A Roadmap**”, Athens University of Economics and Business, Department of Informatics, 2005.
- [Mic98] Michael H. Smith, Stuart Rubin, Ljiljana Trajkovic, “**Fuzzy Data Mining for Querying and Retrieval of Research Archival Information**”, IEEE, Vol. 0-7803-4453, PP. 140-145, 1998.

-
- [Mru05] Mrutyunjaya Swain, J. A. Anderson, N. Swain, Raghu Korrapati, “**Study of Information Retrieval Using Fuzzy Queries**”, 2005, 0-7803-8865 IEEE.
- [Osm98] Osmar R. Zaiane, “**From Resource Discovery to Knowledge Discovery on the Internet**”, Simon Fraser University, Burnaby School of Computing Science, Canada, 2004.
- [Osm99] Osmar R. Zaiane, “**Resource and knowledge discovery from the Internet and Multimedia Repositories**”, Simon Fraser University Burnaby, BC, Canada, 1999.
- [Pie03] Pierre Baldi, Paolo Frasconi, Padhraic Smyth, ”**Modeling the Internet and the Web**”, WILEY, 2003.
- [Ray00] Raymond Kosala, Hendrik Blockeel, “**Web Mining Research: A Survey**”, ACM SIGKDD, Vol. 2, 2000.
- [Ron07] Ronen Feldman, James Sanger, “**The Text Mining Handbook**”, Cambridge University Press, 2007.
- [Rui04] Rui Wurui, Wansheng Tang, Ruiqing Zhao, “**An Efficient Algorithm for Fuzzy Web Mining**”, IEEE, Vol. 0-7803-8819, PP. 576-581, 2004.
- [Sab02] Saba Abdul Khaliq Al-Khadady, M.Sc. thesis, “**Internet and Arabic Search Engine**”, 2002.
- [Sar06] Saran A. AL-Chawishli, “**Automatic Web Text Classification Using Data Mining**”, Iraqi Commission for Computers and Informatics, Informatics Institute for Postgraduate Studies, Ph.D thesis, 2006.

-
- [Sha06] Shaimaa A. Bahaa AL-Deen, “**Mining Web Sites**”, Iraqi Commission for Computers and Informatics, Informatics Institute for Postgraduate Studies, Ph.D thesis, 2006.
- [Sou03] Soumen Chakrabarti, “**Mining the Web: Discovering Knowledge from Hypertext Data**”, Indian Institute of Technology, Bombay, Morgan Kaufmann Publishers, U.S.A, 2003.
- [Sta03] Stanislaw Osinski, “**An Algorithm for Clustering of Web Search Results**”, University of Technology, Poland, M.Sc., 2003.
- [Sve98] Svetlozrr Nestorov, Serge Abiteboul, Rajeev Motwaniz, “**Extracting Schema from Semi-structured Data**”, Stanford University, USA, 2004.
- [Wan03] WangBin, LiuZhijing, “**Web Mining Research**”, Computer Society IEEE, Vol. 0-7695, 2003.
- [Wes05] Wesley Chu, Tsau Young Lin, “**Foundations and Advances in Data Mining**”, Springer, 2005.
- [Wil05] William Siler, James J. Buckley, “**Fuzzy expert systems and fuzzy reasoning**”, JOHN WILEY & SONS, INC., 2005.
- [Zdr07] Zdravko Markov, Daniel T. Larose, “**Data Mining the Web**”, Central Connecticut State University, WILEY, 2007.



Appendix

Appendix A

Some of Stop Words

a	and	been	buy
ability	another	before	bv
able	any	beforehand	bw
about	anybody	began	by
above	anyhow	begin	bz
absolute	anyone	beginner	c
abstract	anything	beginning	ca
according	anywhere	begun	calle
across	ao	behind	came
act	aq	being	can
actually	ar	beings	cannot
ad	are	below	Canon
adj	area	beside	can't
advance	areas	besides	caption
advances	aren	best	case
ae	aren't	better	cases
af	arial	between	cbook
after	around	beyond	cc
afterwards	arpa	bf	cd
ag	as	bg	center
again	asin	bgcolor	certain
against	ask	bh	certainly
ai	at	bi	cf
al	attribute	big	cg
align	au	bigger	cglance
alink	author	billion	ch
all	available	bj	chapter
allcatpop	aw	black	checkbox
allow	away	block	ci
allowing	az	blockquote	ck
allows	b	bm	cl
almost	ba	bn	clean
alone	back	bo	clear
along	backed	body	clearly
already	backing	bold	click

also	backs	book	clos
alt	based	both	cm
although	bb	boy	cn
always	bd	br	co
am	be	browse	co.
amabot	became	bs	color
amazon	because	bt	column
among	become	bucket	columns
amongst	becomes	build	com
an	becoming	but	come
computer	digital	er	finding
concept	dilation	erosion	finds
cond	display	error	first
connect	div	errors	five
connectivity	dj	es	fj
contact	dk	et	fk
contain	dm	etc	fm
containing	do	even	fo
contains	do does	evenly	fobido
content	document	ever	follwo
contents	does	everchanging	follow
continue	doesn	every	font
contour	doesn't	everybody	for
copy	don	everyone	form
copyright	done	everything	former
could	don't	everywhere	formerly
couldn	down	example	forty
couldn't	downed	except	found
count	downing	exec	four
course	downs	exist	fourteen
cr	dr	f	fourty
crank	dt	face	fr
create	during	faces	free
cs	dz	fact	from
cu	e	facts	full
current	each	false	fully
customer	early	far	function
cv	easy	fdetail	fundamenrtal
cx	ec	feed	further
cy	edu	feedback	furthered
cz	ee	felt	furthering
d	eg	fencode	furthermore
day	eh	few	further
days	eight	fexec	futrelle
dd	eighteen	ff	future
de	eighty	ffff	fx
defin	either	fffff	g

define	eleven	fi	ga
description	else	field	gave
detail	elsewhere	fifteen	gb
did	end	fifty	gd
didn	ended	figure	ge
didn't	endind	file	general
differ	ending	files	generally
different	ends	filter	get
differently	enough	find	gets
gf	has	ht	is
gg	hasn	htm	isn
gh	hasn't	html	isn't
gi	have	http	it
gif	haven	hu	its
girrl	haven't	hundred	it's
give	having	i	itself
given	he	i.e.	i've
gives	head	id	j
gl	heading	I'd	javascript
glance	headings	identification	je
gm	he'd	identifier	jiawel
gmt	height	identifiers	jm
gn	he'll	identify	jo
go	help	ie	join
going	helpful	if	jp
goes	helvetica	iframe	just
gone	hence	ii	k
good	her	il	kamber
goods	here	i'll	kaufmann
got	hereafter	illustrate	ke
gov	hereby	im	keep
gp	herein	i'm	keeps
gq	here's	img	kg
gr	hereupon	important	kh
great	hers	in	ki
greater	herself	inc	kind
greatest	he's	inc.	km
group	hieght	include	kn
grouping	high	includes	knew
groups	higher	including	know
gs	highest	indeed	known
gt	him	initializer	knows
gu	himself	input	kostoff
guide	his	inside	kp
gw	hk	instead	kr
gy	hm	int	kw
h	hn	interest	ky

h1	home	interested	kz
h2	homepage	interesting	I
h3	hover	interests	Ia
h4	how	into	language
h5	however	introduction	large
h6	hr	io	largely
had	href	iq	larger
handle	hspace	ir	last
later	mc	mx	now
latest	md	my	nowhere
latter	me	myself	np
laughter	meantime	mz	nr
lb	meanwhile	n	nu
lc	member	na	null
le	members	name	number
least	men	namely	numbers
left	meta	navspacer	nz
length	method	nbsp	o
less	mg	nc	obido
let	mh	ne	of
lets	micheline	necessary	off
let's	microsoft	need	office
level	might	needed	official
levels	mil	needing	offset
li	million	needs	often
like	miss	neither	old
likely	mk	net	older
line	ml	netscape	oldest
link	mm	never	om
links	mn	nevertheless	on
list	mo	new	once
lk	model	newer	onclick
ll	month	newest	one
long	months	news	one's
longer	more	next	only
longest	moreover	nf	onmouseout
lr	morgan	ng	onmouseover
ls	morpholo	ni	onto
lt	morphological	nine	open
ltd	morphology	ninety	opened
lu	most	nineteen	opening
lv	mostly	ninty	opens
ly	mp	nl	operation
m	mq	no	option
ma	mr	nobody	or
made	mrs	noframe	order
main	ms	non	ordered

make	msie	none	ordering
makes	mt	nonetheless	orders
making	mu	noone	org
man	much	nor	other
many	must	not	others
matter	mv	nothing	otherwise
maybe	mw		our
ours	problem	said	sign
ourselves	problems	same	simple
out	product	sample	since
over	prototype	San	sitb
overall	provide	save	site
own	provides	saw	six
p	providing	say	sixteen
pa	pt	sb	sixty
page	put	Sc	size
pap	puts	scr	sj
paper	pw	scrollbar	sk
papers	py	sd	skeleton
part	q	Se	skin
parted	qa	search	sl
parting	quit	secondary	Sm
parts	r	seconds	small
pe	raster	section	smaller
per	rat	see	smallest
perform	rather	seem	sn
performs	re	seemed	snappy
perhaps	reader	seeming	so
pf	really	seems	solid
pg	recent	sees	some
ph	recently	serif	somebody
phase	record	sery	somehow
phrase	related	seven	someone
pk	remainder	seventeen	something
pl	remember	seventy	sometime
place	repeat	several	sometimes
places	report	sg	somewhere
please	reserved	sh	span
pm	resizable	shall	Sr
pn	result	she	st
point	return	she'd	star
pointed	review	she'll	state
pointing	right	she's	states
points	ring	shipping	status
popover	ro	should	step
pos	room	shouldn	steps
possible	rooms	shouldn't	still

pr	row	show	stop
present	rows	showed	store
presented	ru	showes	structure
presenting	rw	si	studies
presents	s	side	studing
price	sa	sides	study
style	thereafter	tops	uy
su	thereby	toward	uz
submit	therefore	towards	v
subst	therein	tp	Va
such	there'll	tr	valign
suggestion	there's	tree	value
summary	thereupon	trillion	values
sure	thirteen	true	variable
sv	these	U	variables
swanson	they	tudeft	vc
sy	they'd	tudelft	ve
system	they'll	turn	verdana
systems	they're	turned	very
sz	they've	turning	vg
t	thier	turns	vi
tab	thing	tv	via
table	things	tw	vlaign
tag	think	twelve	vlink
tags	thinks	twenty	vn
tahoma	thirty	two	vote
take	this	type	vspae
taken	those	types	vu
taking	though	tz	w
target	thought	u	want
tbody	thoughts	ua	wanted
tc	thousand	ug	wanting
td	three	uk	wants
teal	through	ul	was
technology	throughout	urn	wasn
ten	thru	under	wasn't
test	thus	unless	we
text	time	unlike	web
tf	tiny	unlikely	webpage
tg	title	until	website
th	tj	up	we'd
than	tk	upon	wedth
that	whom	you'll	
that'll	whomever	young	
that's	who's	younger	
the	whose	youngest	
their	why	your	

them	width	you're
themselves	will	yours
then	window	yourself
thence	with	yourselves
there	within	you've
wf	without	un
what	womwn	us
whatever	won	use
what'll	won't	used
what's	word	useful
whatsoever	work	users
when	worked	user
whence	working	uses
whenever	works	using
where	would	utility
whereafter	tn	
whereas	tn	
whereby	to	
wherein	today	
whereupon	together	
wherever	tokenizer	
whether	too	
welcome	took	
well	tool	
we'll	top	
wells	wouldn	
went	wouldn't	
were	write	
we're	ws	
weren	www	
weren't	x	
we've	y	
which	ye	
while	year	
white	years	
whither	yes	
who	yesno	
who'd	yesterday	
whoever	yet	
whole	you	
who'll	you'd	

Appendix B

Simple Example for the HTML

Language

HTML Tags

HTML markup tags are usually called HTML tags

- HTML tags are keywords surrounded by **angle brackets** like `<html>`
- HTML tags normally **come in pairs** like `` and ``
- The first tag in a pair is the **start tag**, the second tag is the **end tag**
- Start and end tags are also called **opening tags** and **closing tags**.

HTML Documents - Web Pages

- HTML documents **describe web pages**
- HTML documents **contain HTML tags** and plain text
- HTML documents are also **called web pages**

The purpose of web browsers (like Internet Explorer) is to read HTML documents and display them as web pages. The browser does not display the HTML tags, but uses the tags to interpret the content of the page:

```
<html>
<body>
<h1>My First Heading</h1>
<p>My first paragraph</p>
</body>
</html>
```

Example Explained

- The text between <html> and </html> describes the web page
- The text between <body> and </body> is the visible page content
- The text between <h1> and </h1> is displayed as a heading
- The text between <p> and </p> is displayed as a paragraph

Figure (1) describe simple example of HTML language to learn all example in this thesis.

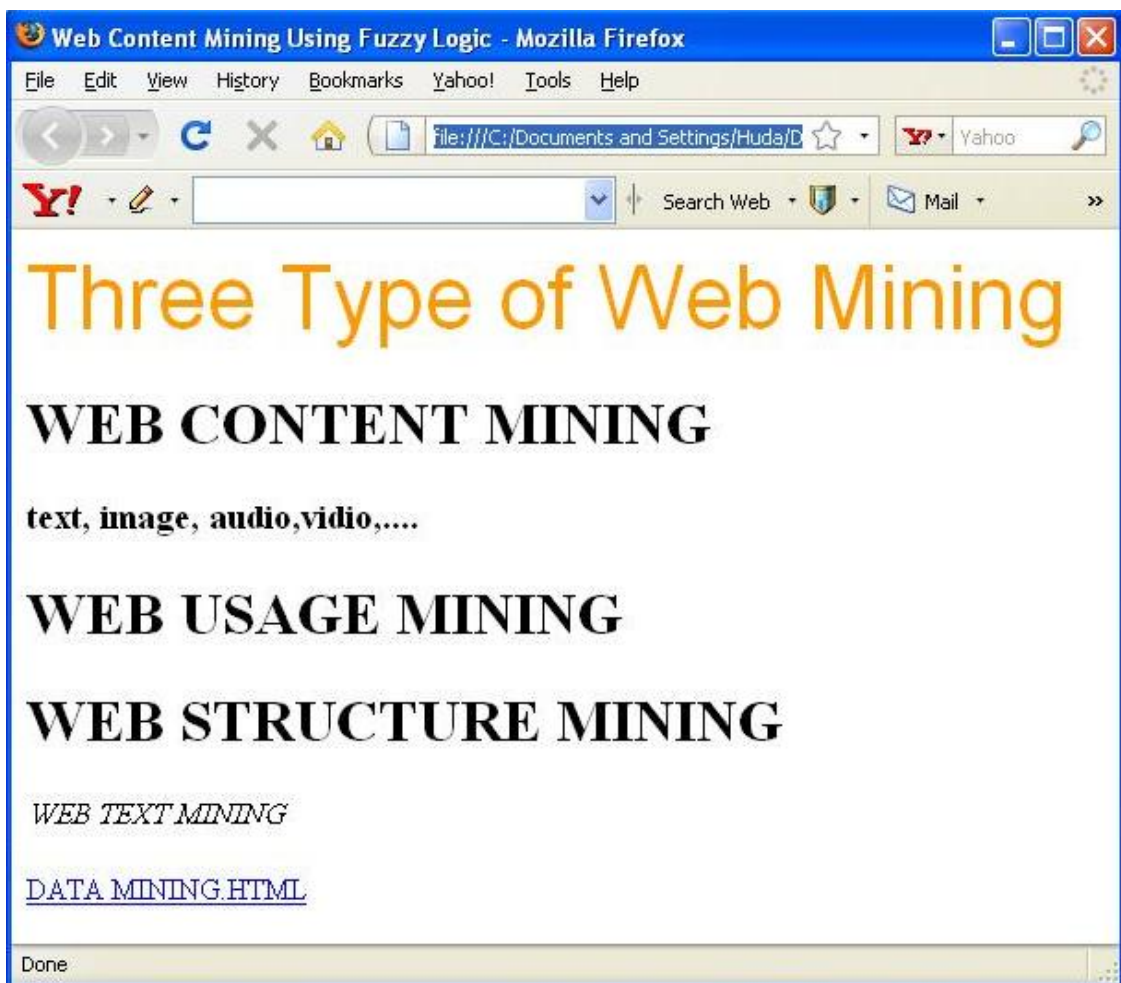


Figure (1): simple page in HTML language.

Figure (2): describe the source file of HTML pages in figure (1) that contents tags and text.

```
<HTML>
<HEAD>
<TITLE> Web Content Mining Using Fuzzy Logic </TITLE> </HEAD>
<BODY>
<FONT SIZE= "7" FACE="Arial" color = "#ff990"> Three Type of Web
Mining </FONT>
<P>
<H1> WEB CONTENT MINING</H1></P>
<STRONG> <H3> text, image, audio,vidio,... </H3> </STRONG>
<H1> WEB USAGE MINING </H1>
<H1> WEB STRUCTURE MINING </H1>
<P> <EM> WEB TEXT MINING </EM> </P>
<H1> </H1>
<P> <A HREF ="DATA MINING.HTML"> DATA MINING.HTML </A>
</P>
</BODY>
</HTML>
```

Figure (2): source file of HTML page.

مع النمو المتزايد في كمية المعلومات على شبكة الإنترنت، أصبح من الصعب جداً على المستخدمين ايجاد وإستعمال المعلومات وكذلك على المجهزين لتصنيف وتعداد الوثائق. وان محرّكاتُ البحث التقليدية عن مواقع الويب تسترجع في أغلب الأحيان المئات أو آلاف النتائج للبحث، الذي يؤدي الى إستهلاك وقت المستخدمين للتجول، لذلك استخدم البحث عن طريق تشابه صفحات الويب.

ان النظام المقترح (التشابه الضبابي لصفحات الويب) يتكون من مرحلتين: مرحلة مقطوعة الاتصال ومرحلة على الاتصال. مرحلة مقطوعة الاتصال تبني متجه الوثائق لقاعدة البيانات بينما المرحلة الثانية تبني وثيقة السؤال ومن ثم تعطي الصفحات المشابه لها. كل وثيقة يجب ان تمر من خلال مجموعة من العمليات حتى نستخلص البيانات التي تمثلها. وهذه العمليات هي محلل النص المعجمي , وازالة كلمات التوقف والكلمات الغير مرغوب بها , تصنيف رتب وثيقة الاثش تي ام ايل , حساب الاوزان للكلمات باستخدام معادلة تعتمد على تردد الكلمات وخواصها كذلك (مثل حجم الخط , نوع الخط , موقع الكلمة داخل الصفحة , نص ارتباطي , عنوان , عنوان خاص) , ثم تبني قاعدة بيانات الوثائق باستخدام الاوزان الكبيرة للكلمات الموجودة في الوثائق.

المرحلة الثانية تتكون من خطوتين: الخطوة الاولى تأخذ السؤال وتبني متجه الوثيقة منه. الخطوة الثانية تحسب نسبة التشابه بين وثيقة السؤال وقاعدة بيانات الوثائق, وهذه الخطوة تنفذ مره باستخدام اجراء التشابه بواسطة معادلة الجيب تمام ومره اخرى بواسطة المنطق المضبيب. ان استخدام المنطق الضبيب يعزز ويحسن نتائج البحث.



جمهورية العراق
وزارة التعليم العالي و البحث العلمي
جامعة النهرين
كلية العلوم

رسالة
مقدمة إلى كلية العلوم, جامعة النهرين
كجزء من متطلبات نيل شهادة الماجستير في علوم الحاسوب

من قبل
هدى عبد المهدي طالب

بكالوريوس
2005

المشرفة
د. سوسن كمال ثامر