CHAPTER 3

NOISE REMOVAL FROM WEB PAGE FOR EFFECTUAL WEB CONTENT MINING

3.1 INTRODUCTION

Analysis and discovery of useful information from World Wide Web poses a phenomenal challenge to the researcher. In this area, the phenomenon of retrieving valuable information by adopting the data mining techniques is called web mining. Web mining is classified into following five sub tasks: 1) Resource finding, 2) Information Selection and Pre-processing, 3) Generalization, 4) Analysis and 5) Visualization. Web mining is divided into three categories: 1) Web Content Mining (WCM), 2) Web Usage Mining (WUM) and 3) Web Structural Mining (WSM).

Web content mining is the process of identifying user specific data from Text, Image, Audio or Video data already available on the web. This process is alternatively called as web text mining, since text content is the most widely researched subjects on the World Wide Web. The technologies that are generally used in web content mining are Information retrieval and Natural language processing. Web Structure mining is another process of using graph theory to analyze the node and connection structure of a web site. Depending upon the type of web structural data, web structure mining has been divided into two fields. The first one is extracting patterns from hyperlinks on the web. The other one is mining the document structure. This involves using the tree-like structure to analyze and describe the HTML or
XML tags within the web page. Web usage mining is to identify user access patterns from Web usage logs. Here we discuss more on web content mining.

3.2 WEB CONTENT MINING (WCM)

Web content mining identifies the useful information from the web contents/data/documents. However, such data in its broader form has to be further narrowed down to useful information. Before narrow down information it is obvious to know that what web content mining is and how web content mining differs from data mining. The web content data consists of structured data such as data in the tables, unstructured data such as free texts, and semi-structured data such as HTML documents. The two main approaches in WCM are (1) Unstructured Text Mining Approach and (2) Semi-Structured and Structured mining approach.

3.2.1 Unstructured Data Mining (Text Mining)

Web content data is much of unstructured text data. The research on unstructured text in data mining includes Knowledge Discovery in Texts (KDT), or Text Data Mining, which also called as Text Mining. Hence one could consider a text mining as an instance of web content mining. To provide effective exploitable results, preprocessing steps for any structured data is done by means of information extraction, text categorization, or applying NLP techniques.

3.2.2 Semi-Structured and Structured Data Mining

Structured data on the web are often very important as they represent their host pages, due to this reason it is important and popular. Structured data is also easier to extract when compared to unstructured texts. Semi-structured data is a point of convergence for the web and database
communities: the former deals with documents, the latter with data. The form of that data is evolving from rigidly structured relational tables with numbers and strings to enable the natural representation of complex real-world objects like books, papers, movies, etc., without sending the application writer into contortions. Emergent representations for semi-structured data (such as XML) are variations on the Object Exchange Model (OEM). In OEM, data are in the form of atomic or compound objects: atomic objects may be integers or strings; compound objects refer to other objects through labeled edges. HTML is a special case of such ‘intra-document’ structure.

3.2.3 Web Content Mining Tasks

Apart from the five tasks enumerated under web mining, another task ‘cleaning’ to be applied in web content mining with the objective of removing redundancy. The detailed explanation about those five tasks is given below.

1. Resource Finding: Resource finding means the process of retrieving the required data from online or offline. Data mining techniques are applied like classification, clustering etc., to extract information.

2. Information Selection and Pre-processing: The pre-processing phase deals with web pages representation. The brief explanation of different representations is given below.

   - Binary: This is called as “Set of words”. The relevance or weight of feature is a binary value \{0, 1\} depending on whether the feature appears in the document or not.

   - Term Frequency (TF): Each term is assumed to have an importance proportional to the number of times it occurs in the text. The weight of a term t in a document d is
given by: \( W(d; t) = TF(d; t) \) is the term frequency of the term \( t \) in \( d \).

- Inverse Document Frequency (IDF): The importance of each term is assumed to be inversely proportional to the number of documents that contain the term. The IDF factor of a term \( t \) is given by: \( IDF(t) = \log_N df(t) \); where \( N \) is the number of documents in the collection and \( df(t) \) is the number of documents that contain the term \( t \).

- TF-IDF: Salton (1998) proposed to combine TF and IDF technique that weight terms. Then, the TF-IDF weight of the term \( t \) in a document \( d \) is given by: \( w(d, t) = TF(d, t) \times IDF(t) \).

- Width Inverse Document Frequency (WIDF): It is an extension of IDF to incorporate the term frequency over the collection of documents. The WIDF weight is given by: \( W(d, t) = TF(d, t)/S_d TF(i, t) \).

- Other pre-processing includes Latent Semantic Indexing (LSI) which determines clusters of co-occurring keywords so that the query which uses a particular keyword can then retrieve documents perhaps not containing this keyword, but containing other keywords from the same cluster.

3. Generalization: Pattern evaluation phase is also called as Generalization. Here machine learning or data mining processes is used to identify general patterns in individual web pages or across multiple sites.
4. Analysis: In the analysis phase accuracy of the retrieved pattern is evaluated using accuracy measures.

5. Presentation or Visualization: To decide in which order the discovered knowledge (web pages) has to be presented.

### 3.2.4 Difference between Web Content Mining, Data Mining and Text Mining

Web content mining uses data mining techniques; it differs from data mining because Web data are mostly unstructured and/or semi-structured, while data mining deals mainly with structured data. It is associated with text mining because much of the web contents are texts. Web content mining differs from text mining because of the semi-structure quality of the web, while text mining deals with unstructured texts. Web content mining thus requires inventive applications of text mining and/or data mining techniques and also its own distinct approaches.

### 3.3 WEB CONTENT MINING ISSUES

The quantity of data / information on the web is enormous and rising quickly. The web data’s are simply open.

1. The coverage of web information is wide and diverse. One can find information about almost anything on the Web.

2. Data of all types exist on the web, like structured tables, texts, multimedia data like images and movies and etc.

3. Information on the web is heterogeneous. Multiple web pages may present the same or similar information using
completely different formats or syntaxes, which makes integration of information a challenging task.

4. The web information is semi-structured due to the nested structure of HTML codes and the need of web page designers to present information in a simple and regular fashion to facilitate human viewing and browsing.

5. The web information is linked. There are links between pages within a site, and across different sites. These links serve as an information organization tool and also as indications of trust or authority in the linked pages and sites.

6. The web information is redundant. The same piece of information or its variations may appear in many pages or sites. This property has been explored in many web data mining tasks.

7. The web is noisy. A web page typically contains a mixture of many kinds of information like main content, advertisements, navigation panels, copyright notices, etc., for a particular application only part of the information is useful, and the rest are considered as noises.

8. The web consists of surface web and deep web. Surface web is composed of pages that can be browsed using a normal web browser. Surface web is also searchable through popular search engines. Deep web is mainly composed of databases that can only be accessed through parameterized queries using query forms.
9. The web is also about services. Many web sites and pages enable people to perform operations with input parameters.

10. Above all, the web is a virtual society. It is not only about data, information and services, but also about interactions among people, organizations and automatic systems.

The web is dynamic. Information on the web changes constantly. Keeping up with the changes and monitoring the changes are important issues for many applications.

3.4 A PROPOSED APPROACH FOR SELECTION OF IMPORTANT BLOCKS

This research work proposes an approach for eliminating noise from web pages for the purpose of improving the accuracy and efficiency of web content mining.

The main objective of removing noise from a web page is to improve the performance of the search engine. It is very essential to differentiate important information from noisy content that may misguide user’s interest. In a web page, the irrelevant data such as advertisements, images, audio, etc., are removed using block splitting operation. The result obtained is a set of blocks.
At first, the web page information is divided into various blocks, from which, the unwanted blocks are removed. For each block, three parameters namely Keyword Redundancy ($K_R$), Linkword Percentage ($L_p$) and Titleword Relevancy ($T_R$) are used. The importance of the block is then
calculated using Simhash algorithm. Based on a threshold value the important blocks are selected using, Sketching algorithm. The performance of the proposed approach is evaluated with several web pages and the results ensure the effectiveness of the proposed approach in identifying the important blocks, which are relevant for knowledge extraction from web pages.

The Figure 3.1 explains the overall procedure of noise removal, selection of important blocks and extracting keywords from a web page for web mining. The different stages of proposed work are as follows:

1. Noise Elimination and Block Splitting
2. Selection of Distinct Blocks using Simhash Algorithm
3. Importance Blocks Calculation using Simhash Algorithm
4. Selection of Important Blocks using Sketching Algorithm
5. Keywords Extraction for Web Content Mining.

3.5 WEB PAGE NOISE

Nowadays, a large number of web pages contained useful information is often accompanied by a large amount of noise such as banner advertisements, navigation bars, copyright notices, etc. These noise data can seriously harm for web miners by extracting whole document rather than the informative content and also retrieve non-relevant results. It is also important to distinguish valuable information from noisy data within a single web page. The web pages are constructed not only main contents information like product information in shopping domain, job information in a job domain but also advertisements bar, static content like navigation panels, copyright sections, etc. When web documents are processed, the main content is surrounded by noise in the retrieved data. Therefore, without removing such
data, the efficiency of feature extraction and finally text classification is certainly degraded. Web noise can be classified as global noises and local noise (Yi et al 2003). Global noises include mirror sites; legal/illegal duplicated web pages, old versioned web page with advertising segments, unnecessary images, or navigation links, etc.

Here the new approach is presented which describes and removal of noisy data to extract main content information as fast and accurate. Many studies are based on information extraction or information retrieval and also try to discover informative content from a set of web documents. Extraction of “useful and relevant” content from web pages has many applications, including cell phone and PDA browsing, speech rendering for the visually impaired, and text summarization. Most approaches are removing clutter or making content more readable involve changing font size or removing HTML and data components such as images, which takes away from a webpage’s inherent look and feel. However, it is relatively little work has been done on eliminating noisy data from web pages in the past. Hence, this research work mainly focus on efficiently and automatically detecting and removing noisy data from web pages to extract and select only important information.

In many web pages, the main content information exists in the middle block and the rest of page contains advertisements, navigation links, and privacy statements as noisy data. Web pages are often cluttered with distracting features around the body of an article that distract the user from the actual content they are interested in. These “features” may include pop-up advertisement, flashy banner advertisements, search and filtering panel, unnecessary images, or links scattered around the screen. However, these noisy data formed in various patterns in different web sites. When extract, only relevant information, such items are irrelevant and should be removed.
Therefore, the mechanism proposed in this work is to eliminate multiple noise patterns in web pages to reduce irrelevant and redundancy data. The HTML `<DIV>`, `<Table>`, and `<TD>` tag is used to detect multiple noise patterns in current web page.

3.5.1 Noise Identification

Information in a web page is not uniformly significant. For example, consider the web page in Figure 3.2, the caption in a news website is much more attractive to users than the navigation bar. And users only just pay attention to the advertisement or the copyright when they browse a web page.

Figure 3.2 Sample Webpage Containing Multiple Regions with Different Importance
Therefore, dissimilar information in a web page has dissimilar importance weight according to its location, occupied area, content, etc. Thus, it is to assign importance to a region in a web page, and need to segment a web page into a set of blocks.

Each resultant group should have similar web pages, while web pages from different resultant groups should be dissimilar. In recent times, most of the available web documents are bound to contain noise and irrelevant information in addition to the significant information. Moreover, a worrying number of documents available on World Wide Web are exact or near duplicates of others. Web content mining includes the task of organizing and clustering the documents and search engines for gaining access to the different documents by keywords, categories, contents etc. The two main research focuses on web content mining have been paid more attention in the past few years. One is information extraction; the other is a web page classification or clustering. At the dawn of the World Wide Web, finding information was done mainly by scanning through lists of links collected and ordered by humans according to some criteria.

3.5.2 Overview of Noise Elimination Process

In this noise elimination process, two mechanisms that determine which region of current web page contain noise or mixture (data and noise region) is illustrated. Then another mechanism is devised on matching to determine how to process the three classes (noise, data and mixture) based on cases. Last, the various noise patterns are removed in current web page and show the extracted main content data. Figure 3.3 describes the detailed architecture of the noise deducting process.
3.5.3 Types of Noise

Noise data of web documents can be categorized into two groups such as global noise and local noise. Global noises are redundant web pages over the Internet such as mirror sites and legal or illegal duplicated web pages. Local noises, only related intra-page redundancy and exist in the web page. This research work focuses on the local noise elimination method. There are at least four different known categories of noise pattern within Web pages of any web sites including banners with links including search panels, advertisements, navigational panel (directory list) and copy right and privacy notice in each web site. It can be seen that many web pages contain these four noise categories together but most of noise patterns are structured by using sectioning tags such as <TABLE> and <DIV> and sectioning separating tags like <FRAMESET>, and interactive tags like <SELECT>, <FIELDSET>.
Input moreover, anchor tag <A> and <IMG> tag are most commonly used to link another web page or another web site. However, these four noise categories can be structured by using various noise patterns (or tags).

### 3.5.4 Fixed Noise

Fixed noise is usually descriptive the information on a webpage or a website. It consists of three sub-types:

1. Decorating noise like site logos and decorative graphics or text, etc.
2. Statement noise, such as copyright notices, privacy statements, license notices, terms and conditions, partners or sponsors statement and etc.
3. Page description noise like date, time and visit counters of the current page and etc.

![Example of Fixed Noise](image)

**Figure 3.4 Example of Fixed Noise**

Figure 3.4 show some examples of fixed noise description of an actual web page. The fixed noise description is usually found either in the form or content.
### 3.5.5 Web Service Noise

Many web pages contain useful service blocks by the way to page content or to manage the server to communicate. This web service blocks are known as noise. There are three types of web service noise illustrated in the Figure 3.5,

1. **Page noise**, as the management of this page and page relocation, etc. services to print the current page and e-mail, or services to jump to other parts of the current page.

2. **Little information board**, like the weather reports and stock board / council reporting market, etc.

3. **Interactive noise service** for users to put their needs. These include inputs based services such as search bars, sign forms, subscription forms, etc., and the selection based services such
as the rating form, quiz forms, voting forms and option menus, etc.

Similar to solid description of noise, noise web service often fixed format and content. However some web sites can implement noise service in java blocked, hence the technique with java scripts in HTML files needed for a full exploration of web service noise.

3.5.6 **Navigational Assistance**

Navigation Tool leadership is common in large websites as it helps users to the sites. It usually serves as intermediate guidance or a shortcut to the pages in a website. There are two main types of navigation guidance and leadership directory recommended guidance.

1. **Guide line** is usually a list of hyperlinks that lead to important index / portal page within a website. It usually reflects the subject categorization and / or topic hierarchies. The three guidance in styles are
   
   i. Global directory guidance shows the main topic categories of the current sites.
   
   ii. Hierarchical leadership guide shows the hierarchical concept space of the current page within a given location.
   
   iii. The Hybrid combines a guide leading the worldwide leading directory and guide hierarchical leadership.

2. **Recommendation guidance** set web users with some potentially interesting sites. It comes in three styles
   
   i. Advertise recommendation is usually a block of hyperlinks that lead to hot items for Web users. This is
shown for commercial purposes. Those hot items are usually advertisements, offers and promotions.

ii. Site recommendation surfers set a few links to other potentially.

![Directory guidance](image1.png)

![Recommendation guidance](image2.png)

**Figure 3.6 Examples of Navigational Guidance Noise**

iii. Page surfer's recommendation set some links related to web's topics is in any way related to the current page. For example, it recommends the pages under the same category of the current page. It can also recommend some pages with similar or related topics.

### 3.5.7 Block Splitting

The rapid expansion of the Internet has made the World Wide Web a popular place for disseminating and collecting information. The innovation of the web creates numerous information sources published as HTML pages on the Internet. Search engines crawl the World Wide Web to collect web pages. These pages are stored and indexed. The user who is performing a
search using search engine is interested in primary informative content of the web page data mining on the web thus becomes an important task for discovering useful information from the web. But a large part of these web pages is content that can not be classified as the primary informative content of the web page. Useful information on the web is often accompanied by advertisements, image-maps, plug-ins, logos, search boxes, category information, navigational links, related links, footers and headers, and copyright information. Although such information items are useful for human viewers and necessary for the web site owners, they often hamper automated information gathering and web data mining. These blocks are not relevant to the main content of the page. Such blocks are referred to as non-content blocks; these blocks are very common in web pages.

The content from a web page can be used, after subdividing the page into smaller semantically homogeneous sections based on their content these sections are referred as blocks or web page blocks. The different types of HTML tags are used to design any web page. Actual data is enclosed within a pair of open and a close tag, a web page block B is a portion of web page enclosed within an open-tag and its matching close-tag. These open and close tags belong to an ordered tag-set T that includes tags like <TABLE>, <TR>, <P>, <HR> and <UL>. The advantage of identifying blocks from web pages is that if user does not want non-content block these can be deleted. These non-content blocks are normally large part of the web pages so eliminating them will be a saving in storage cache and indexing.

Apart from the main (linguistically valuable) content, a typical web page (W) contain reference enters which are referred as primary noises. Removing these noises will help in improving the mining of the web. To assign importance to a region in a web page, first need to segment a web page
into a set of blocks. Hence, to clean a web page, a preprocessing step called Block Splitting Operation is performed.

Basically, the layout of many web pages follows a similar pattern in such a way that the main content is enclosed in one big <DIV> or <TD> element which is HTML tags. This research work concentrating only the content inside the “DIV” tag. The <DIV> tag defines a division or a section in an HTML document and it is often used to group block-elements. The Block Splitting Process aims at cleaning the local noises by considering only the main content of a web page enclosed in a div tag. The main contents obtained are divided into various blocks. The result of this process can be represented as follows:

\[ B = \{B_1, B_2, B_3, \ldots, B_n\}, \quad B \subseteq W \]

Where, 

\[ B \rightarrow \text{A set of blocks in the web page } W \]

\[ n \rightarrow \text{Number of blocks in a web page } W \text{ (cardinality)} \]

\[ n \neq \emptyset \]

Figure 3.7 (a) Part of a Web Page Taken as an Example
The Figure 3.7(a) is taken as an example of a sample web page which consists of local noises such as images, multiple links, etc. and also the main content useful for mining. The dotted line represented in the Figure 3.7(b) is denoted as local noises. The useful main contents for web content mining are highlighted with dark lines. The main content has some sub-contents which are divided into blocks B1, B2 and B3 using block splitting operation.

### 3.5.8 System Description

A table is used for designing maximum web page. In HTML the table format is used for creating almost all web pages. The tag `<TABLE>` is used for creating Table. In table all the fields are considered as a block. Where as tables are not available, block identification process block involves partitioning a web page into sections that are coherent and that have table specific functions. For example, a block with links for navigation is a navigation block. Another example is an advertising block that contains one or more advertisements. Usually, a navigation block is found on the left side of a web page. Typically, the primary informative content block is laid out to the right of a web page.

Similar blocks across different web pages obtained from different websites can be identified. For example, a search on Google News almost any
topic returns several syndicated articles, popular items like syndicated columns or news article written by global news agencies appear in tens of newspapers, Ideals, the user wants only one of these several copies of articles. Since the different copies of the article are from different newspapers web sites, they have similar content blocks while they differ in their non-content blocks only. Separating and indexing only the content blocks it can be easily identified if two web pages have identical content block. This will save on storage and indexing by saving only one copy of the block this will make search results better by returning more unique articles. Even search times improve because it has less data to search.

Content block can be identified based on the appearance of the same block in multiple web pages or based on the occurrence of certain features for identifying content blocks. First, the algorithms partitioning the web page into blocks based on heuristics. These heuristics are based on previous research work of HTML editing style. A web page is divided into blocks on the basis of HTML tables as well as some other tags. Second, the algorithms classify each block as either a content block or a non-content block. While the algorithm decides whether a block, B, is content or not, it also compares B with stored blocks to determine whether B is similar to a stored block. If the block is same as the block stored then it is not necessary to store that again.

3.5.9 Block splitting Algorithm

Step 1 : Select the web page
Step 2 : Identifying Local and Primary web page
Step 3 : Find out Local Noise
Step 4 : If Tag EQL <DIV> or <TD> Then
Read Content

Else

Exit

Step 5 : If identified <DIV> or <TD> tag then

The store content in the each block values {“Prathik”,
“Priya”, “Siva”}

\[ B = \{B_1, B_2, B_3, \cdots, B_n\}, B \subseteq W \]

Where, \( B \rightarrow \) A set of blocks in the web page \( W \)

\( n \rightarrow \) Number of blocks in a web page \( W \)

Step 6 : Continue this process up to end of the web page

Step 7 : Exit

In Figure 3.2, an example of a sample web page is taken which consists of local noises such as images, multiple links, etc., and the main content useful for mining
The Figure 3.8 shows how a block is splitted in a webpage that has been given as an input. The webpage mostly consists of HTML tags, head, title and body that are the most important tags among the various tags available in the HTML web script. Blocks are splitted based on the content, composed in the HTML web script. Usually the content are composed within the DIV and TD sub tags of Body tag. These contents are said to be information. A set of string written within either DIV tag or TD tag is called as one block. If HTML Body tag contains two DIV tag and 4 TD tag then it is accounted that HTML Body has 6 blocks.
3.6 DUPLICATE AND NEAR DUPLICATE DOCUMENTS DETECTION

Duplicate documents are often found in large databases of digital documents like those found in digital libraries or in the government declassification effort. Competent duplicate document detection is significant not only to permit querying for identical documents, but as well to filter out redundant information in large document databases. Duplicate document detection is a scheme employed to avert search results from constituting multiple documents with the same or nearly the same content. There is a possibility for the search quality being degraded as a result of multiple copies of the same or nearly the same documents being listed in the search results.
Following are some of the examples of near duplicate document

- Files with a few different words - widespread form of near-duplicates
- Files with the same content but different formatting – for instance, the documents might contain the same text, but dissimilar fonts, bold type or italics
- Files with the same content but different file type – for instance, Microsoft Word and PDF versions of the same file.

The most challenging among all the above, from the technical perspective, is the first situation i.e. small differences in content. The application of a near de-duplication technology can provide the capacity to recognize these files.

The administration of large, unstructured document repositories is carried out with the aid of Near Duplicate Detection (NDD) Technology. NDD reduces costs, conserves time, and diminishes the risk of errors, building a compelling Return on investment (ROI) in all circumstances where it is necessary for people to make sense of large sets of documents. Near-duplicates are widespread in email, business templates, like proposals, customer letters, and contracts, and forms, including purchase or travel requests. The grouping of near duplicates together improves the document review procedure by:

- The user is offered with set of near duplicates. In place of random review of individual documents, near-duplicate sets make possible a systematic, coherent review process.
- The user need not read individual document anymore. As an alternative the user reads one document from each near
duplicate set. The user just compares the small differences in order to analyze the other documents.

- The regular treatment of near duplicate documents is also guaranteed by the near duplicate grouping.

3.7 NEAR-DUPLICATE DETECTION USING SIMHASH

3.7.1 Simhash and Hamming Distance

simhash is a fingerprinting technique that produces a compact sketch of the objects being study allows for various processing once applied to original data sets to be done on the compact sketches a much smaller and well formatted or fixed length space. With input documents simhash works as follows: a web document is converted into a set of features, each feature tagged with its weight. Then such a high dimensional vector is transformed into an \( f \) - bit fingerprint where \( f \) is quite small compared with the original dimensionality. To make the document self contained here the algorithm’s specification is given in algorithm and explain it with a little more detail. Assumption made here is the input document \( D \), is pre-processed and composed with a series of features (tokens). First initialize an \( f \) –dimensional vector \( V \) with each dimension as zero (step 1). Then for each feature it is hashed into an \( f \) – bit hash value. These \( f \) bits are incremented or decremented the \( f \) components of the vector by the weight of that features based on the value of each bit of the hash value calculated (step 4-8). Finally the signs of the components are determined the corresponding bits of the final fingerprint (step 9-11).
3.7.2 Algorithm specification of Simhash

1. Simhash (document D)
2. { Int vectorSim[0..(f-1)]=0;
3. For (each feature F in document D) Do
4. F is hashed into an f-bit hash value X;
5. For (i=0;i<f;i++) Do
6. If (X[i]==1) Then
7. Stm[i]=Stm[i]+weight(F);
8. Else
9. Stm[i]=Stm[i]-weight(F);
10. End
11. For (t=0;t<f;t++) Do
12. If ( Sim[t]>0)Then
13. sim[t]=1;
14. Else
15. sim[t]=0;
16. End
17. End
18. }

Properties: (1) The fingerprint of a document is a “hash” of its features, and (2) Similar documents have similar hash values. The latter property is quite different from traditional hash function, like message digest algorithm 5(MD5) or Secure Hash Algorithm (SHA-1), where the hash-values of two documents may be quite different even they are slightly different. This property makes Simhash an ideal technique for detecting near-duplicate ones, determining two documents are similar if their corresponding hash values are close to each other. The closer they are, the more similar are these two documents; when the two hash-values are completely same, actually find two exact duplicates, as what MD5 can achieve.
In this research work, a 64-bit fingerprint construction is chosen for each web document because it also works well as shown in (Gong et al 2008). Then the detection of near-duplicate documents becomes the search of hash values with $k$-bit difference, which is also known as searching for nearest neighbors in hamming space (Manku et al 2007). How to realize this goal efficiently? One solution is to directly compare each pair of simhash codes, and its complexity is $O(N/2)$, where $N$ is the size of document repository and each unit comparison needs to compare 64 bits here. A more efficient method as proposed in (Gong et al 2008) is implemented as well in this work. It is composed of two steps. Firstly, all $f$-bit simhash codes are divided into $(k + 1)$ block(s), and those codes with one same block, say 1, 2, …, $(k + 1)$, are grouped into different list. For example, with $k = 3$, all the simhash codes with the same 1st, 2nd, 3rd, or 4th block are clustered together. Secondly, given one simhash code, can get its 1st block code easily and use it to retrieve a list of which all codes sharing the same 1st block as the given one. Normally, the length $N$ of such list is much smaller than the whole size of repository. Besides, given the found list, it is necessary to check whether the remaining blocks of the codes differ with $k$ or fewer bits. The same checking need to be applied to the other 3 lists before we find all simhash codes, i.e. all near-duplicate documents. This search procedure is referred as hamming distance measure in the remaining text.

3.7.3 Formation of Fingerprint using Simhash

The simhash algorithm is used to find both duplicate blocks and important blocks; the duplicate blocks are identified and removed from the web page to retain only one copy of the block. For identifying the importance of each and every block, the three parameters namely Keyword Redundancy ($K_R$), Linkword Percentage ($L_R$) and Titleword Relevancy ($T_R$) are used. The Simhash is a fingerprinting method that enjoys the characteristic where the
fingerprints of near duplicate blocks vary in a small number of bit positions. In this method, at first keywords are extracted from each block and their corresponding frequency is identified.

The keyword frequency refers to the number of times a keyword or keyword phrase appears within a web page. The concept is that the most important keyword or keyword phrase has been the most frequently used keywords in a web page. Using this keyword and its frequency, fingerprint of each block is identified. The process of generating a fingerprint from a block is given as follows.

1. It maintains k-dimensional vector \( A \), each of whose dimensions is initialized to zero.
2. A keyword is hashed into \( f \)-bit hash value using, hashing schema.
3. These \( f \)-bits (unique to the keyword) are incremented or decremented in the \( f \) components of the vector \( A \) by the frequency of that keyword as follows:
   
   i. If the error! Bookmark not defined. i-th bit of the hash value is 1, and then the i-th component of \( A \) is incremented by the frequency of that keyword;
   
   ii. If i-th bit of the hash value is 0, i-th component of \( A \) is decremented by the frequency of that keyword.
4. When all keywords have been processed, some components of the result obtained is the fingerprint of a block. Similarly,
repeat the above steps until \( n \) blocks are processed for finding the fingerprint of each block. Then, from the collection of resultant fingerprints, remove the duplicate blocks whose fingerprints differ at most \( l \)-bit position with other fingerprints.

![Figure 3.9 Generation of Fingerprint using simhash Algorithm](image)

The result obtained in this process is a set of distinct blocks which represents as follows:

\[
B_d = \{ B_{d1}, B_{d2}, B_{d3}, \ldots, B_{dm} \}, B_d \in B, B_{di} \in B_d, i = 1, 2, 3, \ldots, m
\]

Where, \( B_d \rightarrow \) a set of distinct blocks,

\( m \rightarrow \) Number of distinct blocks

### 3.8 IMPORTANT BLOCK SELECTION USING SKETCHING ALGORITHM

The Sketching algorithm is used to identify and select the important blocks from the distinct blocks which are selected by simhash algorithm. The process of finding important blocks is given as follows:
3.8.1 Algorithm specification of Sketching

Step 1: \( U \rightarrow \) space of all possible documents

Step 2: \( S \rightarrow U \): collection of documents

Step 3: Sketching: \( U \times U \rightarrow [0, 1] \): a similarity measure among documents.
   
i) If \( p, q \) are very similar Sketching \( (p, q) \) is close to 1
   
ii) If \( p, q \) are very dissimilar, Sketching \( (p, q) \) is close to 0
   
iii) Usually: Sketching \( (p, q) = 1 - d(p, q) \), where \( d(p, q) \) is a normalized distance between \( p \) and \( q \).

Step 4: \( t = (d - 1/N) \) where \( d \rightarrow \) Number of distinct blocks,
   
   \( N \rightarrow \) Number of blocks

Step 5: \( R \) (Results) \( \rightarrow R \): \( p, q \) is selected if Sketching \( (p, q) \rightarrow t \) (threshold)

Here the normalization technique used is Third Normal Form (3NF)

Web page designers have a tendency to categorize their contents in a reasonable way by giving significance to important contents with proper features such as position, size, color, word, image, link, etc., without highlighting the irrelevant parts. Consequently, a block importance can be defined as a function that map features of importance for each block and is formalized as:

\[
< \text{Block features}> \rightarrow \text{block importance} \quad (3.1)
\]

Thus, there is a need to know which distinct blocks have high importance so that the useful contents are extracted from those high importance blocks. The retrieval of the important blocks of the distinct blocks
is done by computing the importance of each distinct block. The following equation computes the block importance \((b_i)\) of each distinct block.

\[
b_i = 1 - \left\lfloor \frac{1}{2} K_R(i) + \frac{1}{3} L_P(i) + \frac{1}{6} T_R(i) \right\rfloor \quad \text{where} \ 0 \leq b_i \leq 1
\]  

(3.2)

The preprocessing of this computation is to extract essential keywords for the calculation of Block Importance. In this research work, three parameters have been concentrated namely Keyword Redundancy \((K_R)\), Linkword Percentage \((L_P)\) and Titleword Relevancy \((T_R)\) which are very significant for identifying the important blocks. As per the calculations each parameter has assigned its own significant weightage for the best solution, which is \(1/2\) for \(K_R\), \(1/3\) for \(L_P\) and \(1/6\) for \(T_R\). The point to be noted here is the change in weightage of three parameters gave negative impacts in the finding of importance block calculation. The representation of each parameter is as follows

1. **Keyword Redundancy**, \(K_R = \frac{N/d - 1}{N - 1}\)  

(3.3)

Where, \(N\rightarrow\text{Number of Keywords in a block}\)  
\(d\rightarrow\text{Number of Distinct Keywords in a block}\)

2. **Linkword Percentage**, \(L_P = \frac{n_l}{N}\)  

(3.4)

Where, \(n_l\rightarrow\text{Number of Link Keywords in a block}\)

3. **Titleword Relevancy**,  
\[
T_R = - \frac{n_t}{\left\lfloor n_t + \sum_{i=1}^{[p_i]} F(n_t^{(i)}) \right\rfloor}
\]  

(3.5)

Where, \(n_t\rightarrow\text{Number of Title Keywords}, \)
Thus, the values can be obtained by using the above mentioned equations. By substituting the values in Equation (3.2), the block importance of each distinct block is determined.

### 3.9 EXTRACTION OF KEYWORDS

A block importance is obtained for every distinct block $B_d$, from the sketching algorithm. Then, the important blocks are selected only if the block importance $b_i$ is above the threshold level. ($b_i \geq T(i)$). Otherwise, the other blocks are eliminated from the web page as noise blocks. Thus, finally a set of important blocks are obtained and extract the keywords from those important blocks for effective web content mining.

\[
B_I = \{B_{i1}, B_{i2}, B_{i3}, \ldots, B_{ip}\}; \quad B_{i1} \subseteq B_d; \quad B_{ij} \subseteq B_{d_j}; \quad j = 1, 2, 3, \ldots, p
\]  

Where,  
\[
B_I \rightarrow \text{A set of important blocks}  
\]

\[
p \rightarrow \text{Number of important blocks}
\]

### 3.10 CONCLUSION

To improve the performance of web content mining, an approach is proposed which removes noises from web pages. The irrelevant data considered as primary noises have been removed using block splitting operation. From the resultant blocks, the duplicate blocks are removed by computing the fingerprint of each block using simhash. For each block three parameters are computed such as Keyword Redundancy, Linkword Percentage and Titleword Relevancy for knowing the importance of each block. Then the noise blocks are removed by using the threshold value. After removing the noise blocks, the remaining blocks considered as important blocks are extracted. Also the keywords from those blocks are extracted.