CHAPTER 2

LITERATURE REVIEW

2.1 INTRODUCTION

In this chapter, the related work for this research work and existing techniques for noise-free information retrieval of web pages is discussed. It is observed that the web page noise-free information retrieval is related to feature selection, feature weighting, block splitting, duplicate block elimination, and important block calculation in the web content mining field where text files or databases are preprocessed to improve subsequent mining tasks by filtering irrelevant or useless information.

2.2 FEATURE SELECTION TECHNIQUES

Isabelle Guyon and Andre Elisseeff (2003) have proposed a feature selection technique is dealing with text categorization the high dimensionality of feature space. Some feature selection methods remove non-informative terms according to some prior criteria like information gain, document frequency, term frequency mutual information and etc.

Bekkerman et al (2003) has proposed a further dimension by constructing higher level dimensions from combining lower level dimensions. Textual documents or web document are typically modeled as a term vector space where features are personal terms. However, local noise in web pages is regularly blocked by items like images, Texts, hyperlinks etc and Instead of single individual terms. Furthermore, the vector space model cannot capture
the occurring position of terms in web pages. That is, in the traditional feature selection work, a term that occurs in a noisy part of web pages are treated the same as if it occurred in the main part. Different from pure text files, web pages do have some structures which are reflected by their nested hypertext HTML tags. This literature survey assumes that such structural information is useful for noise determination. Therefore, traditional feature selection techniques cannot be directly used to do the web page cleaning. More suitable models are needed to represent web pages and new techniques are needed to do the web page cleaning.

2.3 WEB MINING

Etzioni (1996) has proposed the web techniques to automatically discover and extract information from Web documents and services. This area of research is so huge now available in part on the interests of different research communities, the tremendous growth of information sources on the Internet and the recent interest in e-commerce. Web mining field consists of three main categories, web usage mining, web structure mining and Web content mining. Web usage mining refers to the discovery of user access patterns from web usage logs. Web structure mining tries to discover useful knowledge about the structure of hyperlinks. Web content mining aims to extract / reduce useful information or knowledge of website content. Web mining is often associated with information retrieval (IR) and Information Extraction connected (IE). However, Web mining is not the same as IR or IE

2.3.1 Information Retrieval

Sandor Dominich (2000) have proposed an IR technique which is the automatic retrieval of all relevant documents, while at the same time as some of retrieving irrelevant as possible. Some have claimed that resource or document discovery on the Web an instance of web content mining and web
mining other employees with intelligent IR. Actually, the IR primary goals of indexing text and searching for useful documents in a collection and research today in IR include modeling, document classification and categorization, user interfaces, data visualization, Filtering etc.

Markov et al (2006) have proposed the task that can be considered to be an example of web mining, web document categorization or classification which could be used for indexing. Viewing in this respect, web mining is part of the (web) IR process.

2.3.2 Information Extraction

Cowie and Lehnert (1996) have proposed an Information Extraction (IE) is the goal of transforming a set of documents, usually with the help of an IR system, into information that is more readily digested and analyzed. IE aims to extract relevant facts from the documents while IR aims to select relevant documents. While IE is interested in the structure or representation of a document, IR views the text in a document just as a bag of unordered words. Thus, in general IE works at a finer granularity level than IR dose on the documents.

El-Beltagy et al (2007) have proposed an example of IE without web mining for building a model for automatically augmenting segment documents with metadata using dynamically acquired background domain knowledge in order to assist users in easily locating information within these documents through a structured front end.

Sunita Sarawagi (2008) has proposed a building IE system manually is not feasible and scalable for such a dynamic and diverse medium such as web contents. Due to this nature of the web, most IE systems focus on specific web sites to extract. Others use machine learning or data mining
techniques to learn the extraction patterns or rules for web documents semi automatically or automatically. Within this view, web mining is used to improve web IE (web mining is part of IE).

2.4 WEB CONTENT MINING

The web content mining is mostly based on research in information retrieval and text mining, such as information extraction, text classification and clustering, and information visualization. However, it also includes some new applications, such as Web resource discovery. Various important web content mining techniques and applications are reviewed in this subsection.

Bing Liu (2005) has proposed a web content mining describes the discovery of useful information from the web data/ content/documents. However, what consist of the web contents could encompass a very broad range of data. In this section begin by reviewing some of the important problems that web content mining aims to solve.

Liu et al (2003) have proposed a list some of the special approaches in this field classified; depend on the dissimilar types of web content data. In each approach list some of the most used methods. It is often said that the web offers an extraordinary opportunity and challenge for data mining.

2.4.1 Text Mining for Web Documents

As discussed earlier, text mining is often considered a sub-field of data mining and refers to the extraction of knowledge from text documents (Chen, 2001). Because the majority of documents on the Web are text documents, text mining for Web documents can be considered a sub-field of Web mining, or, more specifically, Web content mining. Information
extraction, text classification, and text clustering are examples of text-mining applications that have been applied to Web documents.

Although information extraction techniques have been applied to plain text documents, extracting information from HTML Web pages can present a quite different problem. As has been mentioned, HTML documents contain many markup tags that can identify useful information.

However, Web pages are also comparatively unstructured. Instead of a document consisting of paragraphs, a Web page can be a document composed of a sidebar with navigation links, tables with textual and numerical data, capitalized sentences, and repetitive words. The range of formats and structures is very diverse across the Web. If a system could parse and understand such structures, it would effectively acquire additional information for each piece of text. For example, a set of links with a heading “Link to my friends’ homepages” may indicate a set of people’s names and corresponding personal home page links. The header row of a table can also provide additional information about the text in the table cells. On the other hand, if these tags are not processed correctly but simply stripped off, the document may become much noisier.

Chang and Lui (2001) used a PAT tree to construct automatically a set of rules for information extraction. The system, called IEPAD (Information Extraction Based on Pattern Discovery), reads an input Web Mining: Web page and looks for repetitive HTML markup patterns. After unwanted patterns have been filtered out, each pattern is used to form an extraction rule in regular expression. IEPAD has been tested in an experiment to extract search results from different search engines and achieved a high retrieval rate and accuracy. Wang and Hu (2002) used both decision tree and SVM to learn the patterns of table layouts in HTML documents. Layout features, content type, and word group features are combined and used as a
document’s features. Experimental results show that both decision tree and SVM can detect tables in HTML documents with high accuracy. Borodogna and Pasi (2001) proposed a fuzzy indexing model that allows users to retrieve sections of structured documents such as HTML and XML.

In some applications the HTML tags are simply stripped from the Web documents and traditional algorithms are then applied to perform text classification and clustering. However, some useful characteristics of Web page design would be ignored. For example, Web page hyperlinks would be lost, but “Home,” “Click here,” and “Contact us,” would be included as a document’s features.

2.4.2 Structured Data Extraction

This is perhaps the most widely studied research topic of web content mining. One of the reasons for its importance and popularity is these structured data on the web are often very important as they represent their host pages. Essential information is like lists of products and services. Extracting such data allows one to provide value added services, like comparative shopping, and meta-search. Structured data are also easier to extract compared to unstructured texts.

This problem has been studied by researchers in AI, database and data mining, and web communities. There are several approaches to structure data extraction, which is also called wrapper generation.

The first approach is to write manually an extraction program for each website based on observed format patterns on the site. This approach is very labor intensive and time-consuming. It thus does not scale too many sites. The second approach is wrapper induction or wrapper learning, which is
the main technique currently. Wrapper learning works as follows: The user first manually labels a set of trained pages. A learning system then generates rules from the training pages. The resulting rules are then applied to extract target items from web pages. Since structured data objects on the web are normally database records retrieved from underlying databases and displayed in web pages with some fixed templates. Automatic methods aim to find patterns/grammars from the web pages and then use them to extract data.

2.4.3 Unstructured Data Extraction

Philipp Cimiano et al (2005) have proposed use of common language patterns (common sentence structures used to express certain facts or relations) and redundancy of information on the web to find concepts, relations among concepts and named entities.

Abidin et al (2010) have proposed patterns that automatically learned or supplied by human users. Another direction of research in this area is, web question-answering. Although question-answering was first studied in information retrieval literature, it becomes very important on the web as the web offers the largest source of information and the objectives of many web search queries are to obtain answers to some simple questions. Extend question-answering to the web by query transformation, query expansion, and then selection.

Lu (2010) had proposed a most web pages can be seen as text documents. Extracting information from web documents has also been studied by many researchers. The research is closely related to text mining, information retrieval and natural language processing. Current techniques are mainly based on machine learning and natural language processing to learn extraction rules from manual labeled examples.
2.4.4  Web Information Integration

Due to the sheer scale of the web and diverse authorship, various web sites may use different syntaxes to express similar or related information. To make use of to extract information from multiple sites to provide value added services, like Metasearch, deep web search, etc., one needs semantically to integrate information from multiple sources. Recently, several researchers attempted this task.

Two popular problems related to the web are (1) Web query interface integration, to enable querying multiple web databases (which are hidden in the deep web) (Solanki and Kumar 2010), and (2) schema matching, like Integrating Yahoo and Google's directories to match concepts in the hierarchies (Erhard Rahm, and Bernstein 2001). The ability to query multiple deep web databases are attractive and interesting because the deep web contains a huge amount of information or data that is not indexed by general search engines.

2.4.5  Building Concept Hierarchies

Because of the huge size of the web, organization of information is obviously an important issue. Although it is hard to organize the whole Web, it is feasible to organize web search results of a given query. A linear list of ranked pages produced by search engines is insufficient for many applications. The standard method for information organization is concept hierarchy and categorization.

The popular technique for hierarchy construction is text clustering, which groups similar search results together in a hierarchical fashion. Several researchers have attempted the task using and clustering (Chuang et al 2004). In (Liu et al 2003), a different approach is proposed which does not use
clustering. Instead, it exploits existing organizational structures in the original Web documents, emphasizing tags and language patterns to perform data mining to find important concepts, sub-concepts and their hierarchical relationships.

2.5 WEB PAGE SEGMENTION AND NOISE DETECTION

A typical web page consists of many blocks or areas, like main content areas, navigation areas, advertisements, etc. It is useful to separate these areas automatically for several practical applications. For example, in web data mining, like classification and clustering, identifying main content areas or removing noisy blocks like advertisements, navigation panels, etc. Enable one to produce much better results. It was shown in (Yi et al 2003) that the information contained in noisy blocks can seriously harm Web data mining. Another application is, web browsing and using a small screen device, such as a PDA. Identifying different content blocks allow one to re-arrange the layout of the page so that the main contents can be seen easily without losing any other information from the page.

2.5.1 Mining Web Opinion Sources

Consumer opinions used to be very difficult to obtain before the web are available. Companies usually conduct consumer surveys or engage external consultants to find such opinions about their products and those of their competitors. Now much of the information is publicly available on the web. There are numerous web sites and pages containing consumer opinions, like customer reviews of products, forums, discussion groups, and blogs. This online word-of-mouth behavior represents new and measurable sources of information for marketing intelligence techniques are now being developed to exploit these sources to help companies and individuals to gain such information effectively and easily.
Hui Guo and Amanda Stent (2006) have proposed a feature based summarization method to automatically analyze consumer opinions in customer reviews of the online merchant sites and dedicated review sites. The result of such a summary is useful to both potential customers and product manufacturers.

### 2.5.2 Classification Based Cleaning Method

A simple method of web page cleaning is to detect specific noisy items like Advertising images, nepotistic hyperlinks, etc., in web pages by adopting some pattern classification techniques. This web page cleaning method is based on classification based cleaning. All existing classifications based cleaning methods simply adopt decision tree classifier to detect noisy items in web pages.

The decision trees classifier technique can be adopted to detect certain kinds of noisy items like Images and linkages in web pages. For example, (Paek and Smith 1998) train the decision tree classifier to recognize banner advertisements; (Kushmerick 2001) trains the decision tree classifier to deal with nepotistic links in web pages. For a certain type of items in web pages, some natural properties and composite properties can be determined, thus each item can be represented as nominal variable. The main steps of decision tree based web page cleaning are as below:

1. Define nominal features for the target type of item (e.g., Images, linkages, etc.)
2. Build decision tree based on (noisy and non-noisy) sample items and extract rules
3. Determine noisy items from non-noisy ones by creating a decision tree or rules
Image and linkages are not the only types of items on web pages. To build decision trees for each type of item is inefficient and inapplicable in practice. For example, it is hard to represent words on web pages with simple and small number of features. Thus the decision tree technique is not applicable for noisy words/sentence detection.

A division (2000) has proposed a decision tree based system, namely AdEater that detects and cleans advertising images in web pages. The AdEater system first defines features for images in web pages. These features include height, width, aspect ratio, ALT features like If the ALT text contains, words free, stuff or not, Ubase features like If the current base URL contains word index, index + html", etc., or not, Udest features like if the destination image URL contains the words “sales”, “contact”, etc., or not, etc. Based on these features, sample images in web pages are encoded as numeric vectors and input to the decision tree training algorithm. After the decision tree was built, the extracted rules or the decision tree was then used to classify real images into noisy and non-noisy. Some interesting rules can be extracted from the decision trees. For example:

- If aspect ratio > 4.5833, alt does not contain “to” but contains “click+here”, and Udest does not contain “http+World Wide Web”, then an instance is an Advertising image.

- If Ubase does not contain “messier”, and Udest contains the “redirect+cgi”, then an instance is an Advertising image.

However, the decision tree is not the only technique that can be adopted to classify noisy items. Some other classification techniques like the support vector machines and the Naïve Bayes can also be used if necessary. The classification based cleaning method is not completely automatic. It requires a large set of manually labeled training data and domain knowledge to define features and generate classification rules.
2.5.3 Segmentation Based Cleaning Method

Yan Fu et al (2007) has proposed a segmentation base cleaning method to detect informative content blocks in web pages based on the observation that a web site usually employs one or several templates to present its web pages. A set of pages created using the same templates is grouped together is called page cluster. If a web site is a page cluster, this work classifies the content blocks in web pages into informative ones and redundant ones. The informative content blocks are the distinguished parts of the page whereas redundant content blocks are common parts. Basically the segmentation based cleaning method discovers informative blocks in four steps: page segmentation, block evaluation, block classification and informative block detection.

1. Page segmentation step extracts out each `<TABLE>` in the DOM tree structure of an HTML page to form a content block. The rest contents which are not contained in any `<TABLE>` also form a special block. Note that the `<TABLE>` may be embedded nodes with `<TABLE>` children if necessary.

![Figure 2.1 Extracting Content Blocks with Text Strings](image_url)
Figure 2.1 shows the content blocks extracting from a sample page, where each rectangle denotes a table with child tables and content strings. Content blocks CB2, CB3, CB4 and CB5 contain content strings S1, S3, S4 and S6 correspondingly. The special blocks CB1 contains strings S2 and S5 which are not contained in any existing blocks.

![Content Blocks Diagram]

**Figure 2.1**

Figure 2.2 Measuring the Entropy Value of a Feature

2. Block Evaluation step selects feasible features (i.e., Terms) from blocks and calculates their corresponding entropy values. The entropy values $H$ of a feature $F_i$ is estimated according to the weight distribution of features appearing in a page cluster.

$$0 \leq H (F_i) = - \sum_{j=1}^{N} W_{ij} \log_d w_{ij} \leq 1$$  \hspace{1cm} (2.1)

Where $w_{ij}$ is the normalized weight of $F_i$ in document $D_j$ and $n$ is the number of documents.
The averaged entropy values $H_{CB_i}$ of a content block $CB_i$ is the normalized summation of features' entropies.

$$H_{CB_i} = \frac{1}{k} \sum_{j=1}^{k} H(F_j)$$ (2.2)

For the example of Figure 2.2, there are $N$ pages with five content blocks (i.e. <TABLE> blocks) on each page. Features F1 to F10 appear in one or more pages as shown in the Figure 2.2. The layout is widely used in dot-com web sites with the logo of a company on the top, followed by advertisement banners or texts, navigation panels on the left, informative content on the right, and its copyright policy at the bottom.

Without losing generality, assume there are only two pages in this Figure 2.2 and the feature entropy is calculated as follows.

$$H(F_1) = \cdots = H(F_6) = -\sum_{j=1}^{6} \frac{1}{2} \log_2 \frac{1}{2} = 1$$ (2.3)

$$H(F_7) = \cdots = H(F_{10}) = -1 \log_2 1 - 0 \log_2 0 = 0$$ (2.4)

3. Block classification step decides the best block entropy threshold to discriminate the informative content blocks from redundant content blocks. By increasing the threshold from 0 to 1.0 with a fixed interval (e.g., 0.1) the approximate best threshold is dynamically decided by a greedy approach.

4. Informative block detection step simply classifies content blocks into informative ones and redundant ones according to the decided best threshold.
The segmentation the following limited based method two assumptions:

1. The system knows a priority how a web page can be partitioned into coherent content blocks; and
2. The system knows a priority which blocks are the same blocks in different web pages.

As it is noticed, partitioning a web page and identifying and corresponding blocks in different pages are actually two critical problems in web page cleaning. Here on this proposed approach are able to perform these tasks automatically. Besides, their work views a web page as a flat collection of blocks which corresponds to <TABLE> elements in web pages, and each block is viewed as a collection of words. These assumptions are often true in news web pages, which is the domain of their applications. In general, these assumptions are too strong.

2.5.4 Template Based Cleaning Method

Neha Gupta and Saba Hilal (2011) have proposed a template based cleaning method to detect templates whereas the templates found were viewed as local noisy data in a web page. With minor modifications, their algorithm can be used for our web page cleaning purpose.

Basically, the template based cleaning method first partitions web pages into piglets and then detects frequent templates among the piglets.

1. Page partition step segments all web pages into logically coherent pagelets. In the template based cleaning method, web pages are assumed to consist of small pagelets. Figure 2.3
shows pagelet examples on the cover page of Yahoo! The pagelet is syntactically defined as follows:

Definition (pagelet): An HTML element in the parse tree of a page p is a pagelet if (1) none of its children contains at least k hyperlinks; and (2) none of its ancestor elements are a pagelet.

Figure 2.3 The Yahoo pagelets

2. Template Detection step finds those frequently occurred pagelets in different Web pages as templates. The syntactic definition of templates is as below.

Definition: A template is a collection of pagelets p, p that satisfies the following two requirements: 1k

1. $C(p_I) = C(p_j)$ for all $1 \leq I \neq j \leq k$  \hspace{1cm} (2.5)

2. $O(p_I) \cdots O(p_k)$ form an undirected connected component \hspace{1cm} (2.6)
Where \( (p_i) \) denotes the page owning pagelet \( p_i \), and \( (p_i) \) denotes the content (HTML content) of the pagelet \( p_i \).

Broder et al (1999) have proposed a set of pagelets which can be viewed as templates, their HTML contents are identical and they are linked by hyperlinks as an undirected connected component. However, the complete matching of the pagelet contents is not applicable because of the natural distortions in World Wide Web such as the version difference and illegal duplications. In practice, the first requirement of completely identical in contents becomes identical with fingerprint method.

There are two algorithms for template detection. The first one is the local template detection algorithm which is suitable for the document sets that consist of a small fraction of documents from the larger universe. The local template detection algorithm in fact only satisfies the first requirement of the template definition.

The second algorithm is the global template detection algorithm which is suitable for template detection in large subsets of the universe. It requires the detected templates be undirected connected by hyperlinks. The template based cleaning method is not concerned with the context of a web site, which can give useful clues for page cleaning. Moreover, in template based cleaning, the partitioning of a web page is pre-fixed by considering the number of hyperlinks that an HTML element has. This partitioning method is simple and useful for a set of web pages from different websites, while it is not suitable for web pages that are all from the same web site because a web site typically has its own common layouts or presentation styles, which can be exploited to partition web pages and to detect noises.
2.6 DETECTION TECHNIQUES FOR IDENTIFICATION OF DUPLICATE AND NEAR DUPLICATE DOCUMENTS

Broder et al (2000) has proposed a technique for the estimation of the degree of similarity among pairs of documents, which was known as shingles, does not rely on any linguistic knowledge, other than the ability to tokenize documents into a list of words, i.e., It is merely syntactic. In shingling, all word sequences of adjacent words are extracted. If two documents contain the same set of shingles they are considered equivalent and if their sets of shingles appreciable overlap, they are exceedingly similar. In order to reduce the set of shingles to a small, however representative, subset they authors employed an unbiased deterministic sampling technique that reduces the storage requirements for retaining information about each document, and the computational effort of comparing documents. A set of 30 million web pages obtained from an AltaVista crawl were employed to apply the technique.

Steinbach et al (2000) has proposed that web pages are grouped into clusters of incredibly similar documents. They identified that in their dataset almost one third of the pages are near duplicates of other pages. The Grainy Hash Vector has (GHV) representation, which can be deployed in cooperative DIR systems for efficient and accurate merge-time duplicates detection. GHVs have the ability to detect near-duplicates and duplicates. They have mathematical properties that are well-defined. They conducted experiments on TREC AP collection and demonstrated that GHVs identify duplicate and near-duplicate document pairs at merge time efficiently and effectively. The management of duplication in cooperative DIR can be excellently performed by GHVs. An approach, based on similarity metrics for the detection of duplicated pages in Web sites and applications, implemented in the HTML language and ASP technology.
Alpuente and Romero (2010) have proposed an approach to identify similar documents based on a conceptual tree-similarity measure. They used the concept associations obtained from a classifier to represent each document as a concept tree. Subsequently, they computed the similarities between concept trees by using a tree-similarity measure based on a tree edit distance. They conducted experiments on documents from the Site-Seer collection and illustrated that when compared to the document similarity based on the traditional vector space model, the performance of their algorithm was significantly better.

Tripathy and Singh (2004) have proposed a technique which was employed to eliminate noise. A tree structures, called the Pattern Tree was proposed to capture the general presentation styles and the definite essential of the pages in a specified Web site. A Pattern Tree called the Site Pattern Tree (SPT) was put up for the site, by sampling the pages of the site.

Cesario et al (2005) have proposed a hybrid queries-dependant duplicate detection scheme that combines the advantages of both online and offline methods. The solution provided in duplicate detection of the hybrid method was effective and in addition scalable. Precisely, the method initially conducts offline processing for popular queries. Then to improve additionally the performance for unpopular queries, it does additional work at run time. The scalability problem of traditional offline methods could be effectively dealt with such a strategy, if the performance problem of traditional online methods is avoided.

Deniel (1999) have proposed in the duplicate document detection, various works have been performed. Their techniques have been utilized by many applications. However, the investigation of the performance and scalability of Duplicate Document Detection (DDD) is modes performed a systematic study of parameter correlations in DDD and evaluated numerous
most important parameters of DDD. The results illustrate that particularly for small documents consisting of a major fraction of the whole Web; the precision of DDD is badly affected by the small sampling ratio. In order to make DDD feasible to deal with large scale document collections, they proposed an adaptive sampling strategy because of their observation, which minimizes the sampling ratio of documents with the constraint of giving precision thresholds. The observations in their work were intended to aid in guiding the future DDD work.

Hui Yang et al (2006) have proposed the problem of duplicate and near-duplicate text has become increasingly important owing to the growth of the text collection in size and various sources from which it is gathered. An instance level constrained clustering was proposed as a solution to near-duplicate detection for notice and comment rulemaking. The ability of Instance-level constrained clustering to express the varied information based on document attributes, information extracted from the document text, and structural relationships among pairs of documents as constraints on cluster contents is its advantage. Thus accuracy and efficiency are improved as the search space is narrowed. They conducted experiments with EPA and DOT datasets. They demonstrated that at less computational cost than competing methods, their approach in detection of near-duplicate was almost efficient as high quality manual assessment.

Manku et al (2007) have proposed made two research contributions in developing a near-duplicate detection system intended for a multi-billion page repository. Initially, they demonstrated the appropriateness of Charikar's fingerprinting technique for the objective. Subsequently, they presented an algorithmic technique to identify the existing f-bit fingerprints that varies from a given fingerprint in at most k bit positions, provided that the value of k is small. Both online queries (single fingerprints) and batch queries (multiple
fingerprints) are aided by this technique. The expediency of the experimental evaluation confirmed their design over real data.

Prasanna Kumar and Govindarajulu (2009) have proposed a problem of finding all documents-pairs swiftly whose similarities are equal to or greater than a given threshold is known as duplicate document detection. A multi-level prefix-filter, which is reduce the number of similarity calculation more efficiently and maintains the advantage of the current prefix filter by applying multiple different prefix-filters. They conducted an experiment with a customer database composed of 200,000 documents and edit distance for similarity calculation. The results illustrate that when compared with the current prefix-filter, the presented method reduces the number of similarity calculations to \( \frac{1}{4} \).

Arasu et al (2001) have proposed a study on the diverse technique to eliminate the duplicates and near duplicate objects in the MyLifeBits personal storage system results of near-duplicate detection for personal contents like emails, documents and web pages visited, are efficient. The number of documents and the number of web pages that a user must consider was reduced by 21% and 43% respectively, by the duplicate and near duplicate detection.

Ranjna Gupta et al (2010) have proposed a new approach that performs copy detection on web documents are copying detection approach determines the similar web documents, similar sentences and graphically captures the similar sentences in any two web documents. Besides handling a wide range of documents, their copy detection approach is applicable to web documents in different subject areas as it does not require static word lists.

Poonkuzhali et al (2009) have proposed a method of distinguishing the redundant links from the web documents that utilized set theory (classical
mathematics) such as subset, union, intersection etc., and proposed an algorithm for mining the web content. Then for obtaining the required information, the redundant links were taken out from the original web content.

Syed Mudhasir et al (2011) have proposed a problem of improving the stability of I-Match signatures with respect to small modifications to document content instead of using just one IMatch signature, they employed numerous that I-Match signatures all which were derived from randomized versions of the original lexicon, in their proposed solution. The proposed scheme does not involve direct computation of signature overlap regardless of employing multiple fingerprints. Hence, in comparison with the case of single-valued fingerprints, the signature comparison is just slightly slower. Furthermore, it can be observed that the addition of one extra signature component can improve signature stability, i.e. further addition of signature components can provide more gains. The successful derivation of lexicons for I-Match from a collection different from the target one, which is most preferred when the target collection is noisy, was demonstrated.

Jack G Conrad et al (2003) have proposed to investigate the phenomenon and determine one or more approaches that minimize its effect on search results. The determination of the extent and the types of duplication existing in large textual collections was their chief objective. In addition, one or more approaches that minimize its deleterious impact on search results in an operational environment were devised. The issues of computational efficiency and duplicate document detection (and, by extension, “deducing”) are effectiveness in relying on “collection statistics” to recognize consistently document replicas in full-text collections.

Ilyinsky et al (2002) have proposed the process of managing these during incremental growth, acquisitions, mergers, and integration efforts
inevitably results in some duplication are defined, this natural entropy by using a process that mines the repository for partially duplicated material, helping to maintain the highest quality control of the content. Although the overall process is satisfactorily efficient with computer resources, practically, human attention to consider the many results is the bottleneck. In conclusion, a special handling of duplicates and a way to reduce a frequent source of false alarms-template similarity is provided. Internet Search Engines are posed with challenges owing to the growth of the Internet that flood more copies of web documents over search results making them less relevant to users suggested a method of "descriptive words" for definition of near-duplicates of documents, which is because of the choice of N words from the index to determine a "signature" of a document. Any search engine based on the inverted index can apply this method.

The method based on "shingles" and the authors compared the suggested method. At almost equal accuracy of algorithms, their method in the presence of inverted index was more efficient.

The need for various forms of duplicate document detection has increased due to the accelerated growth of massive electronic data environments, both web-based and proprietary. This detection can take any of several forms based on the nature of the domain and its customary search paradigms; however either identical or non-identical deducing can be used to characterize basically them.

Huda Yasin and Mohsin Mohaammad Yasin (2011) have proposed a Jaccard Coefficients for calculating similarity of text attributes. Analytical Hierarchy Processed (AHP) is used to obtain the weights of entity. Using the sum of weights, the entity similarity is calculated and it needs to integrate duplicate entity for achieving the entity identification.
2.7 DETECTION ALGORITHMS

Broder et al (2000) have proposed an identification of near exact duplicate web page shingling algorithm random projection based aloft he were co-utilized state-of-the-art" algorithms. These two algorithms are compared, on a very large scale, specifically for a set of 1.6 Byte distinct web pages. In accord number of the results, if of identifying the near duplicate pairs on the same site, neither of the algorithms works well, whereas if of dissimilar sites, they both obtain high precision. In general, the Charikar's algorithm achieves a better precision, namely the 0.50 viewed archive's the (Broder et al 2000) algorithm as the former identifies more near-duplicate pairs on different sites. The combined algorithm presented by the author attains the precision of 0.79 with 79% of the recall of the other algorithms.

Chowdhury et al (2001) have proposed a novel similar document detection algorithm called I-Match. They used multiple data collections to evaluate their performance. The employed document collections were different in terms of size, degree of expected document duplication, and document lengths. NIST and Excite@Home were the source of the data employed. It was illustrated the key point IMatch, operates because of the number of documents and it deals with documents of all sizes efficiently. In comparison with the state of the art, their method proved to have improved the accuracy of detection. In addition, the exec web-robots were about one-fifth of the time taken by the state of art method.

Ahmad (2004) have proposed a novel data reduction algorithm employing the concept analysis which can be used as a filter in retrieval systems like search engines to eliminate redundant references to the similar documents. A study was performed on the application of the algorithm in automatic reasoning which effected in minimizing the number of stored facts without loosing of knowledge, by the authors. Their results illustrate that
besides reducing the user time and increase his satisfaction; there was a good increase in the precision of the retrieval system.

Fetterly et al (2004) have proposed a study on the evolution of web pages over time during which many machines-generated “spam” web pages emanating from a handful of web servers in Germany, was discovered. The grammatically well-formed German sentences drawn from a large collection of sentences were stitched together to assemble dynamically these spam web pages. The development of techniques to find other instances of such “slice and dice” generation of web pages, where pages are automatically generated by stitching together phrases drawn from a limited corpus, is aggravated by the discovery.

Muhammad Sheikh Sadi et al (2004) have proposed an efficient algorithm to measure relevance among web pages using hyperlink analysis (RWPHA). RWPHA searched the web using the URL of a page rather than the set of query terms, given as input to the search process. A set of related web pages is the output. A web pages that address the same topic as the original page is known as the related page. RWPHA does not employ the content of pages or usage information rather only the connectivity information on the web (i.e., the links between pages) is utilized. The extended Co citation analysis is the basis of the algorithm. The superior performance of the algorithm over some dominant algorithms in finding relevant web pages from the experimental results illustrated linkage information.

Daniel P Lopresti (1999) has proposed a significant amount of network bandwidth for clarifying and formalizing the duplicate document detection problem. They used uncorrected OCR output to study several issues related to the detection of duplicates in document image databases. They presented four distinct models for formalizing the problem and in each case they present algorithms that determine the best solution. The algorithm most
suited to a particular a solid highlighted problem dot (●). Whereas the algorithm that will find not only such duplicates but other types as well, is showed by a hollow dot (○). They conducted experiments by using data reflecting real-world degradation effects to illustrate the robustness of their techniques.

Kanhaiya Lal and Mahanti (2010) have proposed a novel algorithm, Dust Buster, for uncovering DUcorrespond trials with similar Text, Dust Buster, for uncovering DUcorrespond trials with similar Text. They intended to discover rules that transform a given URL to others that are likely to have similar content. Dust Buster employs previous crawl logs or web server logs instead of probing the page contents to mine the dust efficiently. It is necessary to fetch few actual web pages to verify the rules via sampling. Search engines can increase the effectiveness of crawling, reduce indexing overhead, and improve the call-in the form of popularity statistics such as Page Rank, which are the benefits provided by the information about the DUST.

Yang and Callan (2006) have proposed a work for exact-near-duplicate detection, for which the process of identifying near duplicates of form letters is the focus. They defined the most near and exact-duplicates that are appropriate to eRulemaking and explored the employment of simple text clustering and retrieval algorithms for the task. The effectiveness of the experiments illustrated the method in the public comment domain.

Yang and Callan (2006) have proposed a refinement of a prior near-duplicate detection algorithm. DURIAN (DUPLICATE REMOVAL IN large collection N), identifies form letters and their edited copies in public comment collections by employing a traditional bag-of-words document representation, document attributes ("metadata"), and document content structure. In accordance with the experimental results, DURIAN was almost as effective as
human assessors. They discussed the challenges in moving the near-duplicate detection into operational rulemaking environments, in conclusion.

Deng and Rafiei (2006) have proposed establishing a simple algorithm known as Stable Bloom Filter (SBF), which is based on the following idea: Given that there was no way to store the whole history of the stream, the stale information is removed by SBF to provide space for those more recent elements. They systematically identified some properties of SBF and consequently illustrated a guaranteed tight upper bound of false positive rates. The authors conducted experiments to compare the SBF with the alternative methods. If a fixed small space and an acceptable false positive rate were given, the outcome illustrated that their method was superior in terms of both accuracy and time efficiency.

Huffman et al (2007) have proposed a problem of duplicate detection as a part of such evaluation was added. Their results illustrate that the combination of multiple text-based signals and its computation over both fetched and rendered bodies significantly improve the accuracy of duplicate detection algorithms. They deemed that by

1. Detecting and removing boilerplate from document bodies
2. More fine-grained feature selection
3. Using more sophisticated URL-matching signals, and
4. Training over large data sets, the quality of the model can be improved additionally. A computational cost is involved in the increase in performance. Only for the modest-sized problems, the resulting algorithm was feasible. The exploration of the idea of combining multiple signals in other duplicate detection domains with stricter computational limits, such as web crawling and indexing would be fascinating.
Gong et al (2008) have proposed the SimFinder, an effective and efficient algorithm to identify all near duplicates in large-scale short text databases. The three techniques, namely, the ad hoc term weighting technique, the discriminative-term selection technique, the optimization technique are included in this SimFinder algorithm. It was illustrated that the SimFinder was an effective solution for short text duplicate detection with almost linear time and storage complexity by the experiments conducted.

Bassma (2011) have proposed a method that can eliminate near-duplicate documents from a collection of hundreds of millions of documents by computing independently for each document a vector of features less than 50 bytes long and comparing only the vectors rather than entire documents are provided that m is the size of the collection, the entire processing takes time \( O(m \log m) \). The algorithm illustrated, has been successfully implemented and is employed in the context of the AltaVista search engine, currently.

Deng and Rafiei (2006) have proposed an efficient and elegant probabilistic algorithm to approximate the number of near-duplicate pairs are created. The algorithm scans the input data set once and uses only small constant space, independent of the number of objects in the data set, to provide a provably accurate estimate with high probability. They performed a theoretical analysis and the experimental evaluation on real and synthetic data. They illustrated that in reasonably small dimensionality, the algorithm significantly outperforms the alternative random-sampling method.

Xiao et al (2008) have proposed an efficient similarity joins algorithms that exploit the ordering of tokens in the records. Various applications such as duplicate web page detection on the web have been provided with efficient solutions. They illustrate that the existing prefix filtering technique and the positional filtering, suffix filtering are complementary to each other. The problem of quadratic growth of candidate
pairs when the data grows in size was effectively solved. They evaluated their algorithms on several real datasets under a wide range of parameter settings and proved to be superior when compared to the existing prefix filtering-based algorithms. In order to improve the result quality or accelerate the execution speed, their method can additionally be modified or incorporated with existing near duplicate web page detection methods.

### 2.8 WEB BASED TOOLS

Cooper et al (2002) have proposed a system for rapidly determining document similarity among a set of documents obtained from an Information Retrieval (IR). They utilized a rapid phrase recognizer system to obtain a ranked list of the most important terms in each document, which was stored in a database and a simple database query was used to compute document similarity. Two documents are determined to be similar if the number of terms found not to be contained in both documents is less than some predetermined threshold compared to the total number of terms in the document.

Raphael et al (2002) have proposed described a web-accessible text registry based on signature extraction. From every registered text, a small but diagnostic signature was extracted intended for permanent storage and comparison against other stored signatures. Even though the total time required was linear in the total size of the documents, the number of overlap between pairs of documents can be estimated by the comparison performed. Their results illustrate the efficiency of Signature Extraction (SE) for detecting document overlap. Hence, (1) hashed-breakpoint chunking, (2) culling by the variance method, (3) retaining only 10 hex digits of the MD5 digest, (4) storing in a Perl database, (5) computing symmetric similarity, were the basis of their algorithm.
Udi Manber (2007) have proposed a tool, known as SIF that intends to identify all similar files in a large file system. Files having a significant number of common pieces, though they are very different otherwise, are considered to be similar. The execution times for identifying all groups of similar files is in the order of 500MB to 1GB an hour, even for similarity of files as little as 25%. The rapid pre-processing stage can be employed by the user to determine the number of similarities and several other customized parameters. Additionally, SIF utilizes a preprocessed index to identify swiftly all similar files to a query file. The applications such as file management, information collecting (to remove duplicates), program reuse, file synchronization, data compression, and maybe even plagiarism detection employ SIF.

Jalbert and Weimer (2008) have proposed a system that automatically classified duplicate bug reports as they arrived to save development time. Their system predicted duplicate status by utilizing surface features, textual semantics, and graph clustering. They employed a dataset of 29,000 bug reports from the Mozilla project which is larger than the datasets commonly used before to evaluate experimentally their approach. They illustrated that their task was not helped by the inverse document frequency. They employed their model to simulate as a filter in a real-time bug reporting environment. Their system filtered out 8% of duplicate bug reports, thus ably reducing the development cost. Hence, practically, carried out in a production environment with little added effort and a feasible major payoff, based on their approach.
2.9 CONCLUSION

In this chapter the related work of this research work and existing techniques for noise free information retrieval of web pages are discussed. It is observed that Web page noise free information retrieval is related to feature selection, feature weighting, block splitting, duplicate block elimination, and important block calculation in the of web content mining field where text files and databases are preprocessed to improve subsequent mining tasks by filtering irrelevant or useless information.