Imperial Journal of Interdisciplinary Research (IJIR)
Vol.2, Issue-1 , 2016
ISSN : 2454-1362 , www.onlinejournal.in

An Efficient Approach for Near-duplicate page detection in web crawling

K. Subramanyam Sharma
Dept of CSE, CMR Technical Campus Hyderabad, India

Dr. K. Srujan Raju
Dept of CSE, CMR Technical Campus Hyderabad, India

P.Yadagiri
Dept of CSE, CMR Technical Campus Hyderabad, India

***Abstract*—** The drastic development of the World Wide Web in the recent times has made the concept of Web Crawling receive remarkable significance. The voluminous amounts of web documents swarming the web have posed huge challenges to the web search engines making their results less relevant to the users. The presence of duplicate and near duplicate web documents in abundance has created additional overheads for the search engines critically affecting their performance and quality. The detection of duplicate and near duplicate web pages has long been recognized in web crawling research community. It is an important requirement for search engines to provide users with the relevant results for their queries in the first page without duplicate and redundant results. In this paper, we have presented a novel and efficient approach for the detection of near duplicate web pages in web crawling. Detection of near duplicate web pages is carried out ahead of storing the crawled web pages in to repositories. At first, the keywords are extracted from the crawled pages and the similarity score between two pages is calculated based on the extracted keywords. The documents having similarity scores greater than a threshold value are considered as near duplicates. The detection has resulted in reduced memory for repositories and improved search engine quality.

Index Terms— ***Web Crawling, Generic Crawling, Focused Crawling, Web Mining, Search Indexing, Web Parsing, Keywords Extracted, Page Ranking, Similarity Score Calculation***

1. INTRODUCTION

The employment of automated tools to locate the information resources of interest, and for tracking and analyzing the same, has become inevitable these days owing to the drastic development in the information accessible on the World Wide Web. This has made the development of server-side and client-side intelligent systems mandatory for efficient knowledge mining [1]. A branch of data mining that deals with the analysis of World Wide Web is known as Web Mining. Web Mining owes its origin to concepts from diverse areas such as Data Mining, Internet technology and World Wide Web, and lately, Semantic Web [2]. Web mining includes the sub areas: web content mining [3], web structure mining [4], and web usage mining [5] and can be defined as the procedure of determining hidden yet potentially beneficial knowledge from the data accessible in the web. The process of mining knowledge from the web pages besides other web objects is known as Web content mining. Web structure mining is the process of mining knowledge about the link structure linking web pages and some other web objects. The mining of usage patterns created by

the users accessing the web pages is called Web usage mining [6].

The World Wide Web owes its development to the Search engine technology. The chief gateways for access of information in the web are Search engines. Businesses have turned beneficial and productive with the ability to locate contents of particular interest amidst a huge heap [31]. Web crawling, a process that populates an indexed repository of web pages is utilized by the search engines in order to respond to the queries [20]. The programs that navigate the web graph and retrieve pages to construct a confined repository of the segment of the web that they visit. Earlier, these programs were known by diverse names such as wanderers, robots, spiders, fish, and worms, words in accordance with the web imagery [7].

Generic and Focused crawling are the two main types of crawling. Generic crawlers [9] differ from focused crawlers [10] in a way that the former crawl documents and links of diverse topics whereas the latter limits the number of pages with the aid of some prior obtained specialized knowledge. Repositories of web pages are built by the web crawlers so as to present input for systems that index, mine, and otherwise analyze pages (for instance, the search engines) [8]. The subsistence of near duplicate data is an issue that accompanies the drastic development of the Internet and the growing need to incorporate heterogeneous data [21]. Even though the near duplicate data are not bit wise identical they bear a striking similarity [21]. Web search engines face huge problems due to the duplicate and near duplicate web pages. These pages either increase the index storage space or slow down or increase the serving costs thereby irritating the users. Thus the algorithms for detecting such pages are inevitable [22]. Web crawling issues such as freshness and efficient resource usage have been addressed previously [11], [12], [13]. Lately, the elimination of duplicate and near duplicate web documents has become a vital issue and has attracted significant research [15].

Identification of the near duplicates can be advantageous to many applications. Focused crawling, enhanced quality and diversity of the query results and identification on spam's can be facilitated by determining the near duplicate web pages [19, 26, 22]. Numerous web mining applications depend on the accurate and proficient identification of near duplicates. Document clustering [17], detection of replicated web collections [18], detecting plagiarism [29], community mining in a social network site [30], collaborative filtering [16] and discovering large dense graphs [27] are a notable few among those applications. Reduction in storage costs and enhancement in quality of search indexes besides considerable bandwidth conservation can be achieved by eliminating the near duplicate pages [9]. Check summing techniques can determine the documents that are precise duplicates (because of mirroring or plagiarism) of each other [14]. The recognition of near duplicates is a tedious problem. Research on duplicate detection was initially done on databases, digital libraries, and electronic publishing. Lately duplicate detection has been extensively studied for the sake of numerous web search tasks such as web crawling, document ranking, and document archiving. A huge number of duplicate detection techniques ranging from manually coded rules to cutting edge machine learning techniques have been put forth [21, 22, 23, 34 - 37]. Recently few authors have projected near duplicate detection techniques [38, 39, 40, 25]. A variety of issues such as from providing high detection rates to minimizing the computational and storage resources have been addressed by them. These techniques vary in their accuracy as well. Some of these techniques are computationally pricey to be implemented completely on huge collections. Even though some of these algorithms prove to be efficient they are fragile and so are susceptible to minute changes of the text.

The primary intent of our research is to develop a novel and efficient approach for detection of near duplicates in web documents. Initially the crawled web pages are preprocessed using document parsing which removes the HTML tags and java scripts present in the web documents. This is followed by the removal of common words or stop words from the crawled pages. Then the stemming algorithm is applied to filter the affixes (prefixes

and the suffixes) of the crawled documents in order to get the keywords. Finally, the similarity score between two documents is calculated on basis of the extracted keywords. The documents with similarity scores greater than a predefined threshold value are considered as near duplicates. We have conducted an extensive experimental study using several real datasets, and have demonstrated that the proposed algorithms outperform previous ones.

The rest of the paper is organized as follows. Section 2 presents a brief review of some approaches available in the literature for duplicates and near duplicates detection. In Section 3, the novel approach for the detection of near duplicate documents is presented. The conclusions are summed up in Section 4.

2. RELATED WORK

Our work has been inspired by a number of previous works on duplicate and near duplicate document and web page detection.

In [34], a system was proposed for registering documents and then detecting copies, either complete copies or partial copies. Algorithms were described for detection, and metrics required for evaluating detection mechanisms covering accuracy, efficiency and security. A prototype implementation of the service, COPS, was also described and experimental results were presented that suggest the service can indeed detect violations of interest.

In [35], an efficient way to determine the syntactic similarity of files was developed and was applied it to every document on the World Wide Web. Using this mechanism, a clustering of all the documents that are syntactically similar was built. Possible applications include a "Lost and Found" service, filtering the results of Web searches, updating widely distributed web-pages, and identifying violations of intellectual property rights.

In [36], the extent and the types of duplication existing in large textual collections were determined. The research was divided into three parts. Initially it was started with a study of the distribution of duplicate types in two broad-ranging news collections consisting of approximately 50 million documents. Then the utility of document signatures in addressing identical or nearly identical duplicate documents and their sensitivity to collection updates was examined. Finally, a flexible method of characterizing and comparing documents in order to permit the identification of non-identical duplicates was investigated. The method has produced promising results following an extensive evaluation using a production-based test collection created by domain experts.

In [37], mechanisms for measuring the intermediate kinds of similarity were explored, focusing on the task of identifying where a particular piece of information originated. A range of approaches was proposed to reuse detection at the sentence level, and a range of approaches for combining sentence-level evidence into document-level evidence. They considered both sentence-to-sentence and document-to-document comparison, and have incorporated the algorithms into RECAP, a prototype information flow analysis tool.

In [38], the use of simple text clustering and retrieval algorithms for identifying near-duplicate public comments was explored. It was focused on automating the process of near-duplicate detection, especially form letter detection. A clear near-duplicate definition was given and explored simple and efficient methods of using feature-based document retrieval and similarity-based clustering to discover near-duplicates. The methods were evaluated in experiments with a subset of a large public comment database collected for EPA rule.

In [22], the comparison of the two algorithms namely shingling algorithm [35] and random projection based approach [14] on a very large scale set of 1.6B distinct web pages was done. The results showed that neither of the algorithms works well for finding near-duplicate pairs on the same site, while both achieve high precision for near-duplicate pairs on different sites. She has

presented a combined algorithm which achieves precision 0.79 with 79% of the recall of the other algorithms.

In [39], DURIAN (DUplicate Removal In lArge collectioN) was presented, a refinement of a prior near-duplicate detection algorithm. DURIAN uses a traditional bag-of-words document representation, document attributes ("metadata"), and document content structure to identify form letters and their edited copies in public comment collections. The results have demonstrated that statistical similarity measures and instance-level constrained clustering can be quite effective for efficiently identifying near-duplicates.

In the course of developing a near-duplicate detection system for a multi-billion page repository, in [25] two research contributions was made. First, it was demonstrated that fingerprinting technique in [14] is appropriate for this goal. Second, an algorithmic technique was presented for identifying existing f-bit fingerprints that differ from a given fingerprint in at most k bit-positions, for small k. This technique is useful for both online queries (single fingerprints) and batch queries (multiple fingerprints).

In [40], the problem of DUST: Different URLs with Similar Text was considered. A novel algorithm, DustBuster, for uncovering dust; that is, for discovering rules that transform a given URL to others that are likely to have similar content was proposed. DustBuster mines dust effectively from previous crawl logs or web server logs, without examining page contents.

In [21], exact similarity join algorithms with application to near duplicate detection and a positional filtering principle was proposed, which exploits the ordering of tokens in a record and leads to upper bound estimates of similarity scores. They demonstrated the superior performance of their algorithms to the existing prefix filtering-based algorithms on several real datasets under a wide range of parameter settings.

3. NOVEL APPROACH FOR NEAR DUPLICATEWEBPAGE DETECTION

A novel approach for the detection of near duplicate web pages is presented in this section. In web crawling, the crawled web pages are stored in a repository for further process such as search engine formation, page validation, structural analysis and visualization, update notification, mirroring and personal web assistants or agents and more. Duplicate and near duplicate web page detection is an important step in web crawling. In order to facilitate search engines to provide search results free of redundancy to users and to provide distinct and useful results on the first page, duplicate and near duplicate detection is essential. Numerous challenges are encountered by the systems that aid in the detection of near duplicate pages. First is the concern of scale since the search engines index hundreds of millions of web-pages thereby amounting to a multi-terabyte database? Next is the issue of making the crawl engine crawl billions of web pages every day. Thus marking a page as a near duplicate should be done at a quicker pace. Furthermore, the system should utilize minimal number of machines [25].

The near duplicate detection is performed on the keywords extracted from the web documents. First, the crawled web documents are parsed to extract the distinct keywords. Parsing includes removal of HTML tags, java scripts, stop words/common words and stemming of remaining words. The extracted keywords and their counts are stored in a table to ease the process of near duplicates detection. The keywords are stored in the table in a way that the search space is reduced for the detection. The similarity score of the current web document against a document in the repository is calculated from the keywords of the pages. The documents with similarity score greater than a predefined threshold are considered as near duplicates.

3.1 Near Duplicate Web Documents

Even though the near duplicate documents are not bitwise identical they bear striking similarities. The near duplicates are not considered as “exact duplicates” but are files with minute differences. Typographical errors, versioned, mirrored, or plagiarized documents, multiple representations of

the same physical object, spam emails generated from the same template, and many such phenomenon's may result in near duplicate data. A considerable percentage of web pages have been identified as be near-duplicates according to various studies [17, 26 and 22]. These studies propose that near duplicates constitute almost 1.7% to 7% of the web pages traversed by crawlers. The steps involved in our approach are presented in the following subsections.

3.2. Web Crawling

The analysis of the structure and informatics of the web is facilitated by a data collection technique known as Web Crawling. The collection of as many beneficiary web pages as possible along their interconnection links in a speedy yet proficient manner is the prime intent of crawling. Automatic traversal of web sites, downloading documents and tracing links to other pages are some of the features of a web crawler program. Numerous search engines utilize web crawlers for gathering web pages of interest besides indexing them. Web crawling becomes a tedious process due to the subsequent features of the web, the large volume and the huge rate of change due to voluminous number of pages being added or removed each day.

Seed URLs are a set of URLs that a crawler begins working with. These URLs are queued. A URL is obtained in some order from the queue by a crawler. Then the crawler downloads the page. This is followed by the extracting the URLs from the downloaded page and enqueuing them. The process continues unless the crawler settles to stop [28]. A crawling loop consists of obtaining a URL from the queue, downloading the corresponding file with the aid of HTTP, traversing the page for new URLs and including the unvisited URLs to the queue [7].

3.3 Web Document Parsing

Information extracted from the crawled documents aid in determining the future path of a crawler. Parsing may either be as simple as hyperlink/URL extraction or complex ones such as analysis of HTML tags by cleaning the HTML content [7]. It is inevitable for a parser that has been designed to traverse the entire web to encounter numerous errors. The parser tends to obtain information from a web page by not considering a few common words like a, an, the and more, HTML tags, Java Scripting and a range of other bad characters [24].

3.3.1 Stop Words Removal

It is necessary and beneficial to remove the commonly utilized stop words such as "it", "can" ,"an", "and", "by", "for", "from", "of", "the", "to", "with" and more either while parsing a document to obtain information about the content or while scoring fresh URLs that the page recommends. This procedure is termed as stop listing [7]. Stop listing aids in the reduction of size of the indexing file besides enhancing efficiency and value.

3.4 Stemming Algorithm

Variant word forms in Information Retrieval are restricted to a common root by Stemming. The postulation lying behind is that, two words posses the same root represent identical concepts. Thus terms possessing to identical meaning yet appear morphologically dissimilar are identified in an IR system by matching query and document terms with the aid of Stemming [33]. Stemming facilitates the reduction of all words possessing an identical root to a single one. This is achieved by removing each word of its derivational and inflectional suffixes [32]. For instance, "connect", "connected" and "connection" is all condensed to "connect".

3.5 Keywords Representation

We possess the distinct keywords and their counts in each of the each crawled web page as a result of stemming. These keywords are then represented in a form to ease the process of near duplicates detection. This representation will reduce the search space for the near duplicate detection. Initially the keywords and their number of occurrences in a web page have been sorted in descending order based on their counts. Afterwards, n numbers of keywords with highest

counts are stored in a table and the remaining keywords are indexed and stored in another table. In our approach the value of n is set to be 4. The similarity score between two documents can be calculated if and only the prime keywords of the two documents are similar. Thus the search space is reduced for near duplicates detection.

3.6 Similarity Score Calculation

If the prime keywords of the new web page do not match with the prime keywords of the pages in the table, then the new web page is added in to the repository. If all the keywords of both pages are same then the new page is considered as duplicate and thus is not included in the repository. If the prime keywords of new page are same with a page in the repository, then the similarity score between the two documents is calculated. The similarity score of two web documents is calculated as follows:

Let T1 and T2 be the tables containing the extracted keywords and their corresponding counts.

T_1	K_1	K_2	K_4	K_5	…..	K_n
	C_1	C_2	C_4	C_5	…..	C_n

T_2	K_1	K_3	K_2	K_4	…..	K_n
	C_{11}	C_3	C_{21}	C_{41}	…..	C_{n1}

Individually the keywords of both the tables are taken into account for the calculation of score. The following is the formula used for calculating the score with common keywords in both the tables:

$$a = \Delta[K_i]_{T_1} \quad (1)$$

$$b = \Delta[K_i]_{T_2} \quad (2)$$

$$S_{D_C} = \log(count(a)/count(b)) * Abs(1 + (a - b))$$

The index of the keywords is represented by 'a' and 'b'.

If the keywords of T_1 / $T_2 \neq \varphi$, the following formula is used to calculate the similarity score. The occurrences of the keywords present in T_1 but not in T_2 is taken as N_{T_1}

$$S_{D_{T_2}} = \log(count(a)) * 1 + |T_2| \quad (4)$$

If the keywords of T_2 / $T_1 \neq \varphi$, the following formula is used to calculate the similarity score. The occurrences of the keywords present in T_2 but not in T_1 is taken as N_{T_2}

$$S_{D_{T_2}} = \log(count(b)) * (1 + |T_1|) \quad (5)$$

The Similarity Score Measure (SSM) of a page against another page is calculated by using the following equation, Where $N = (|T_1| + |T_2|)/2$.

$$SSM = \frac{\sum_{i=1}^{|N_C|} S_{D_C} + \sum_{i=1}^{|N_{T_1}|} S_{D_{T_1}} + \sum_{i=1}^{|N_{T_2}|} S_{D_{T_2}}}{N} \quad (6)$$

4. CONCLUSION

Although the web is a huge information store, Information Retrieval has been posed with serious difficulties owing to its various features, for instance, the presence of huge volume of unstructured or semi-structured data; their dynamic nature; existence of duplicate and near duplicate documents and the similar ones. Huge challenges have been posed by the voluminous amounts of web documents swarming the web to the web search engines making their less appropriate to the users. The web crawling research community has extensively recognized the detection of duplicate and near duplicate web pages. We have presented a novel and efficient approach for detection of near duplicate web documents in web crawling in this paper. On the basis of the keywords extracted from the web pages, the proposed approach has detected the duplicate and near duplicate web pages efficiently. The proposed duplicates detection approach also accomplishes reduced memory spaces for web repositories and improved search engine quality. The aspects which may be considered for the future scope for detecting the near duplicate documents could be Images, Color Text, Hyperlinks, Blinking words, and high priority text/words. All the above parameters were not

taken into consideration while the preprocessing is done. These all were removed in the preprocessing and then the detection process was performed.

5. REFERENCES

[1]V.A.Narayana,P.Premchandand A.Govardhan,(2009)" A Novel and Efficient Approach For Near Duplicate Page Detection in Web Crawling". IEEE International Advance Computing Conference.

[2]Arasu, A., Cho, J., Garcia-Molina, H., Paepcke, A., and Raghavan, S., (2001) "Searching the web", ACM Transactions on Internet Technology, vol. 1, no. 1: pp. 2-43.

[3] Brin, S., Davis, J., and Garcia-Molina, H., (1995) "Copy detection mechanisms for digital documents", In Proceedings of the Special Interest Group on Management of Data (SIGMOD 1995), ACM Press, pp.398–409, May.

[4]Broder, A. Z., Glassman, S. C., Manasse, M. S. and Zweig, G., (1997) "Syntactic clustering of the web", Computer Networks, vol. 29, no. 8-13, pp.1157–1166.

[5] Bacchin, M., Ferro, N., Melucci, M., (2002) "Experiments to evaluate a statistical stemming algorithm", Proceedings of the international Cross-Language Evaluation Forum Workshop, University of Padua at CLEF 2002.