DOI: http://dx.doi.org/10.26483/ijarcs.v9i1.5352

ISSN No. 0976-5697

Volume 9, No. 1, January-February 2018

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

A NOVEL TWO-PHASE PAGE FEATURE AND KTH KEYPHRASE FINGERPRINT BASED DUPLICATE DETECTION TECHNIQUE

Ashlesha Gupta
Assistant Professor
Computer Engineering Deptt., YMCAUST
Faridabad, Haryna, India

Ashutosh Dixit
Assosiate Professor
Computer Engineering Deptt., YMCAUST
Faridabad, Haryna, India

A.K. Sharma
Professor& Dean(Research)
Computer Engineering Deptt., BSAITM
Faridabad, Haryana, India

Abstract: The World Wide Web is a huge repository of network-accessible information including text, image, audio, video and metadata. With rapid increase in information resources available via WWW and users of the Internet, it is becoming difficult to manage and access the desired information on the web. Therefore, majority of users use information retrieval tools like search engines to find the desired information from the WWW. Web search engines work by storing information about many web pages, which they retrieve from the WWW itself in search engine repositories . These repositories may contain duplicate or near duplicate pages which results in higher storage area and processing costs. To overcome these increased space /cost requirements and to provide redundant free results to the users duplicate detection and elimination algorithms are used. The proposed duplicate detection approach uses two phase page level and kth keyphrase fingerprint based scheme to detect and eliminate near duplicate web pages so that quality of the result-sets may be improved.

Keywords: Duplicate page, Near-duplicate page, Filtering, Finger-print, Page-features

I. INTRODUCTION

World Wide Web is a huge, diverse and dynamic source of information which is expanding day by day. In order to trace relevant data, users depend on variety of search engines for finding suitable answers for their queries. A Search Engine is an information retrieval system which helps users finds information on WWW by making the web pages related to their query available. It gathers, analyzes and organizes the data from the internet and offers users an interface to retrieve the network resources[1]. With the search engine user types in the query keywords and in response to it, Search Engine returns a list of clickable URL's. This returned result set of a search engine, however, contains a mixture of both relevant and irrelevant information including duplicate and near-duplicate web pages. Duplicate web pages are mirrored copies of some web pages and near duplicates are not bit wise identical but differ in some parts of the web page like the advertisements, counters and timestamps [2]. Multiple versions of the same page, storing the document on multiple servers , creating documents using same templates or spamming are the probable reasons of existence of duplicate pages[3]. Early recognition of these duplicates help in reducing network bandwidth and storing costs .

To detect duplicate and near duplicate web pages a novel duplicate detection technique based on fingerprint similarity is being proposed. In the proposed technique firstly a web page feature based filtering is applied to eliminate the far similar documents and then kth keyphrase fingerprint technique of the web page is computed and a bit-by-bit difference is measured between new crawled web page and other web pages in the repository. If the difference is less than the threshold value, the cawled page is a near duplicate and is discarded else the page is stored in the repository. The objective of the technique is to quickly and accurately identify duplicate and near duplicate web pages thereby improving the quality of search results and saving storage space and hence network bandwidth.

The rest of the paper is organized as follows: The Existing approaches to Duplicate detection are covered in Section II. Section III discusses the proposed duplicate detection technique. Experimental set-up and results are summarized in Section V. Section VI includes conclusion and Future scope.

II. LITERTURE REVIEW

Duplicate Detection techniques help search engines in providing quality and redundant free results to the user. A number of methods have been proposed for recognizing and eliminating such duplicates. Broder et al. [4] have suggested a technique, in which all sequences of adjacent words are extracted. If two documents contain the same shingles set they are treated as equivalent and if the shingles set overlaps, they are considered as exact similar. But this method does not work well on small documents. Theobald et al.'s[5] proposed SpotSigs method[6] for duplicate detection. In this method stop words in anchor tag were first identified then k tokens after an anchor leaving the stop words were grouped as k-gram and were referred as spot signatures . The document was then represented as a collection of these spot signatures . The

© 2015-19, IJARCS All Rights Reserved

length of these spot signature vectors were reduced using hash function.The documents with similar spot signatures were identified as duplicates. Fetterly et al. [7] use five-gram as a shingle and sample 84 shingles for each document. Then the 84 shingles are built into six *super shingles*. The documents having two *super shingles* in common are considered as nearly duplicate documents. A sentence level duplicate detection technique for news-articles was proposed by Hung-Chi Chang and Ten-Hour Wang [8]. Hannaneh Hajishirzi *et al.* [9] developed duplicate detection technique for identifying duplicates in same domains. Every document in the proposed technique was treated as a k-gram vector. These k-gram vectors were then mapped to hash-values as document signatures through locality sensitive hashing scheme. Bingfeng Pi *et al.* [10] proposed the use of SimHash algorithm for finding near duplicate. Narayana et al. [11] proposed duplicate detection technique wherein the keywords of the document are extracted and keyword similarity score between newly crawled and stored web pages is used for duplicate detection. Salha Alzahrani et al. [12] suggested a fuzzy based semantic method for detecting plagiarism.

III. PROPOSED TECHNIQUE

For fast and efficient detection of duplicate pages , a novel duplicate detection approach is being proposed that uses two phase filtering techniques to detect duplicates and near duplicates. In the proposed approach first a web page level feature based comparison among the new and stored web pages is done. This action eliminates the far similar documents and thus reduces the number of web documents to the second level of filtering. Since only numerical values are compared so this filtering requires less time and disk space. In the second stage fingerprint of all kth keywords in the web pages are computed and a bit by bit difference between the fingerprints of the crawled and stored web pages is made . If the difference is less than the specified threshold value, the crawled page is near duplicate and dropped otherwise the crawled page is stored in the repository. Fingerprints of only limited web pages need to be created due to initial page level feature filtering, the proposed technique incurs less time for duplicate detection and provides precise results. The working of the proposed technique is given in Figure 1

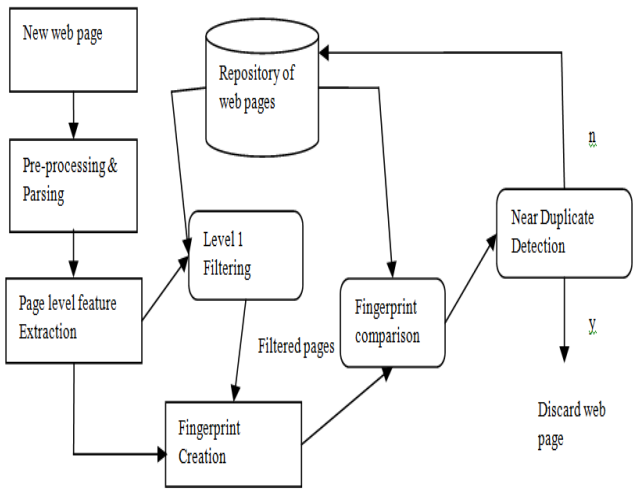

Figure 1: Working of Proposed Technique

A. *Preprocessing*

The newly crawled page is parsed and preprocessed for page level feature extraction . The Preprocessing includes stop-word removal and stemming. Following web page features are extracted from the parsed and preprocessed web page : No. of tables in the web page, No.of images/figures in the web page, No. of sentences, No. of Anchor tags and keywords of the web page. The information is then passed to Level 1 filtering.

B. *Level-1 Filtering*

After preprocessing, Level 1 filtering is used for finding the duplicate web pages. The filtering involves comparing the extracted page level features of the newly added web page with the similar extracted features of already stored web pages and assigning a score value to the compared page.The score value is assigned as per the details given in Table 1.

Table 1: Page Feature Table

Feature	Threshold Value	Computed Value	Score
No. of tables	A1	Tn-To where Tn and To refer to number of tables in new and old web page	If Tn-To is less than A1 a score of 1 is assigned else score value is 0
No. of Images/Figures	A2	In-Io where In and Io refer to number of images in new and old web page	If In-Io is less than A2 a score of 1 is assigned else score value is 0
No. of Anchor tags	A3	An-Ao where An and Ao refer to number of anchor tags in new and old web page	If An-Ao is less than A3 a score of 1 is assigned else score value is 0
No. of Sentences	A4	Sn-So where Sn and So refer to number of sentences in new and old web page	If Sn-So is less than A4 a score of 1 is assigned else score value is 0
Keyword Similarity	A5	No. of Common	If KS is less than

© 2015-19, IJARCS All Rights Reserved

		words in new and old web page ———— Total number of words in new and old page	A5 a score of 0 is assigned else score value is 1

The scores thus assigned for each feature are combined and a final score value is calculated. If the total score value is greater or equal to the specified threshold score , then the compared page is filtered in. The advantage of this step is that it is fast and consumes less disk space since only numbers are compared and reduces the number of web pages for next level of filtering. Hence, the number of inputs to the fingerprint comparison will get reduce, so that fingerprint of only a limited number of web page is to be found out, rather than all the web pages in the database.

C. Fingerprint Comparison

Fingerprints of the filtered web pages is then computed. Instead of finding the fingerprint of the entire document , kth keyphrases are selected and fingerprint of these kth keyphrases are computed. To compute the fingerprint first character of each kth key phrase is taken and is converted to its ascii value. The sum of all ascii values is computed and converted to binary form. This process of finger-print comparison is repeated with each document filtered in first phase.

D. Near Duplicate Detetion

For near duplicate detection first a threshold is set. The kth keyphrase fingerprint of the newly crawled page is compared with fingerprint of all other stored pages. The comparison is done bit-by-bit. If the difference is less than the threshold value , the document is discarded as it is a near-duplicate else the newly crawled page is added to the search engine repository.

IV. RESULTS & DISCUSSION

The proposed duplicate detection approach has been implemented in Java with Net beans as frontend and MS Access as the backend. The query terms “SEO”, “Web-Crawler” and “I-Phones” were given to Google search engine. The results of first 5 pages are being used for experimentation. The results of the study are being given in following sub-sections. The analysis of the experimental results confirmed that the proposed approach is able to achieve its objective of time and space reduction.

A. Experimental Results

For providing sample results, we take 3 web page documents from web crawling procedure. The web pages were first processed to remove stop words and then process of stemming was applied. The web page documents along with the extracted keywords are given in the Table 2.

Table 2: Web documents along with extracted keywords

Web page documents	Keywords
WP1	Web, Search, Engine, Software, information, mine, data, editor, crawler, database, WWW, System, design, user, index, huge, site, directory
WP2	Search, Engine, document, keywords, list, Google, Bing, list, specific, WWW
WP3	Internet, Search, Engine, program, information, crawler, index, Bing, WWW, directory, index

First, the page level feature in each of the web page document is computed and then the finger print of each document is obtained using the proposed algorithm. The features extracted and the fingerprints from the taken documents are given in the table 2 and table 3. The fingerprints are calculated by taking every 6^{th} keyphrase in the web document

Table 3: Page Level Features of web Documents

Web Page	No. of Tables	No. of Figures	No. of Anchor tags	No. of Sentences
Wp1	0	2	6	6
WP2	0	1	9	4
WP3	0	2	5	5

Table 4: Fingerprint of web documets

Web Document	Fingerprint
WP1	0010010011100111
WP2	1001011100011110
WP3	0001111000111110

For near duplicate web page detection, Cw is taken as newly crawled web page. The page-level features of Cw were extracted and compared with similar features of all three pages The threshold values for A1,A2,A3,A4 and A5 were set equal to 2,2,2,2 and 50% respectively.

Table 5 : Page level features and Fingerprint of new web page Wpnew

Web Page	Keywords	No. of Tables	No. of Images	No. of Anchor tags	No. of Sentences	Fingerprint
new	Search, Engine, Mine, data, information, user,	0	2	8	6	0010110011101100011

© 2015-19, IJARCS All Rights Reserved

	index, site, repository , crawler, google, Bing, keywords					

Since WP1 and WP2 match Cw in page level1 filtering, they are selected for kth keyphrase filtering. The next process is comparing bit-by-bit the kth keyphrase finger prints of Cw with fingerprint of WP1 and WP3. The threshold value is set to 3. By comparing the fingerprint of Cw with WP1 and WP3 it was found that the difference between Cw and WP1 is 1 which is less than the specified threshold value of 3. Hence, Cw is considered as near duplicate web page and it is not added to the database.

Table 6: Comparison Table

Dataset	Count of web pages considered	Precision of NDupDet Algo			Proposed Technique		
		Precision	Recall	Time to detect duplicates (ms)	Precision	Recall	Time to detect duplicates (ms)
Search Engine	50	0.7	0.4	180	0.8	0.8	120
Crawler	45	0.58	0.5	160	0.8	0.75	80
SEO	35	0.5	0.5	120	0.7	0.7	60

B. Performance Evaluation

The proposed duplicate detection approach is compared with Near Duplicate Web Page Detection using NDupDet Algorithm in terms of Precision , Recall and Computation time where Precision and Recall are defined as :

Precision=No. of true duplicates detected/Total number of duplicate detected.
Recall= No. of true duplicate detected/Total number of duplicates in the dataset
Computation time= Time required to identify near duplicates.

Table 6 lists out the precion, recall and computation time values for NDupDet algorithm[13] and proposed technique for duplicate detection. The precion, Recall and Computation plots are shown in Figure 2, 3 and 4 respecctively.

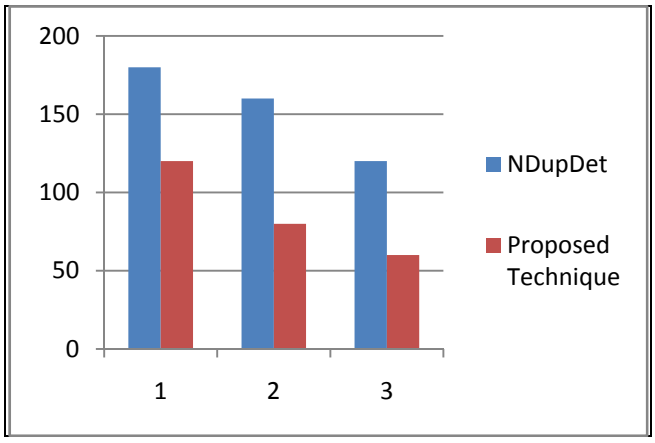

Fig 2: Precision plots between two techniques

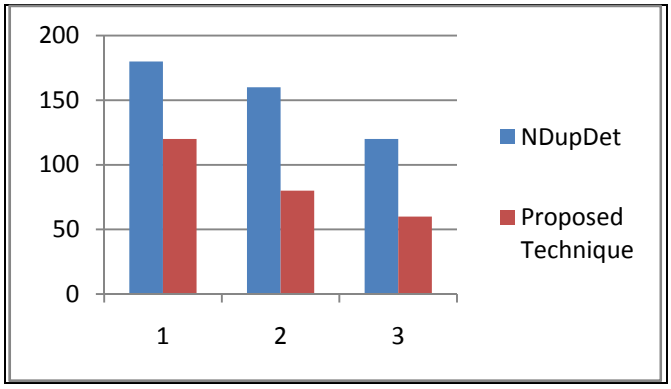

Fig 3: Recall plots between two techniques

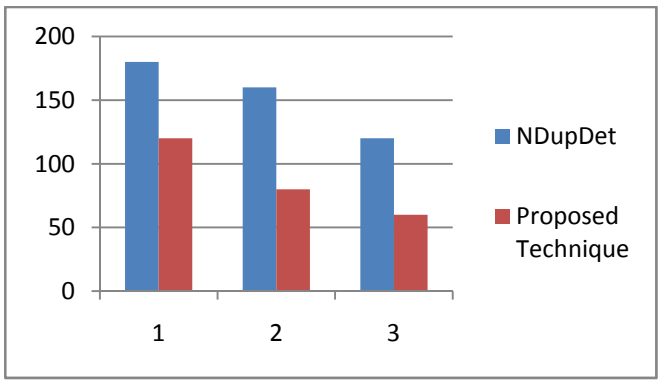

Fig 4: Computation time plots

It can be viewed from the Precision and Recall plots that the proposed technique is better and accurate in identifying near duplicates. Also time incurred for identifying near duplicates has been significantly reduced by using the proposed technique (as shown in Fig 4) since the new web page have to be compared with less number of web documents rather than all web documents in the database. Hence the proposed technique outperforms in accuracy and incurs less time and disk space.

V. CONCLUSION

A new duplicate detection technique is being proposed based on two phase filtering techniques that are applied serially one after the other. The first phase of page level feature

© 2015-19, IJARCS All Rights Reserved

comparison is fast enough as it involves comparing numerical values only and fingerprint technique is used for providing high precision results and fast computation with limited storage. The proposed technique is efficient enough and is able to provide effective, precise and less time consuming results.

VI. REFERENCES

[1] Gupta Ashlesha, Dixit,Ashutosh Sharma A.K. : Relevant Document Crawling with Usage Pattern and Domain profile based Page Ranking. ,ISCON, 2013, Proceedings of IEEE International Conference held at GLA University, Mathura , pp 119-124,2012.

[2] Y. Syed Mudhasi et al. Near-Duplicates Detection and Elimination Based on Web Provenance for Effective Web Search. International Journal on Internet and Distributed Computing Systems (IJIDCS) . Vol: 1 No: 1, pp 22-32, 2011

[3] V.A. Narayana, P. Premchand and A. Govardhan, “Effective Detection of Near-Duplicate Web Documents in Web Crawling”, International Journal of Computational Intelligence Research, Volume 5, Number 1, pp. 83–96, 2009.

[4] Andrei Z. Broder, Steven C. Glassman, Mark S.Manasse, and Geoffrey Zweig, “Syntactic Clustering of the Web”, In Proceedings of the Sixth International Conference on World Wide Web, pp : 1157-1166, 1997.

[5] Theobald, M., Siddharth, J., and Paepcke, A. 2008. Spotsigs: robust and efficient near duplicate detection in large web collections. In SIGIR. pp. 563–570

[6] Bassma S. Alsulami, Maysoon F. Abulkhair, Fathy E. Eassa, “Near Duplicate Document Detection Survey",International Journal of Computer Science & Communication Networks,Vol 2(2), pp. 147-151,2010

[7] Fetterly, D., Manasse, M. and Najork, M. On the evolution of clusters of near duplicate web pages, In Proceedings of the first Latin American Web Congress (LAWeb), pp. 37–45, 2003.

[8] Hung-Chi Chang, and Jenq-Haur Wang, "Organizing News Archives by Near-Duplicate Copy Detection in Digital Libraries", *Asian Digital Libraries*, Vol. 4822, pp: 410-419, 2007.

[9] Hannaneh Hajishirzi, Wen-tau Yih, and Aleksander Kolcz, "Adaptive Near-Duplicate Detection Via similarity Learning" , In Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval ,SIGIR '10, pp.416-426, 2010.

[10] Bingfeng Pi, Shunkai Fu, Weilei Wang , and Song Han, "SimHash-Based Effective and Efficient Detecting of Near-Duplicate Short Message " Proceedings of the Second Symposium International Computer Science and Computational Technology(ISCSCT ’09), pp. 020- 025,Dec. 2009

[11] V.A. Narayana, P. Premchand and A. Govardhan, “Effective Detection of Near-Duplicate Web Documents in Web Crawling”, International Journal of Computational Intelligence Research, Volume 5, Number 1, pp. 83–96, 2009.

[12] Alzahrani, S., & Salim, N. (2010). Fuzzy semantic-based string similarity for extrinsic plagiarism detection. *Braschler and Harman*, *1176*, pp 1-8.

[13] N.Joshi, J.Gagde, Near Duplicate Web Detection Using NDupDet Algorithm, International Journal of Computer Applications , Volume 61, No.4, pp 56-59,2013

© 2015-19, IJARCS All Rights Reserved