International Journal of Computer Applications (0975 – 8887)
Volume 19– No.7, April 2011

A Novel Approach for Near-Duplicate Detection of Web Pages using TDW Matrix

Midhun Mathew
School of Computing
SASTRA University
Thanjavur, India

Shine N Das
Dept. of Computer Applications
Cochin University of Science & Technology
Cochin, India

T R Lakshmi Narayanan
School of Computing
SASTRA University
Thanjavur, India

Pramod K Vijayaraghavan
Dept. of Computer Applications
Cochin University of Science & Technology
Cochin, India

ABSTRACT
The voluminous amount of web documents has weakened the performance and reliability of web search engines. The subsistence of near-duplicate data is an issue that accompanies the growing need to incorporate heterogeneous data. Web content mining face huge problems due to the existence of duplicate and near-duplicate web pages. These pages either increase the index storage space or increase the serving costs thereby irritating the users. Near-duplicate detection has been recognized as an important one in the field of plagiarism detection, spam detection and in focused web crawling scenarios. Here we propose a novel idea for finding near-duplicates of an input web-page, from a huge repository. We proposes a TDW matrix based algorithm with three phases, rendering, filtering and verification, which receives an input web-page and a threshold in its first phase , prefix filtering and positional filtering to reduce the size of records in the second phase and returns an optimal set of near-duplicate web pages in the verification phase after calculating its similarity. The experimental results show that our algorithm outperforms in terms of two benchmark measures, precision and recall, and a reduction in the size of competing record set.

General Terms
Web Content Mining, Web Technologies, Information Retrieval, Search Engine Optimization

Keywords
Near-Duplicate Detection, Term-Document-Weight Matrix, Prefix filtering, Positional filtering, Singular Value Decomposition.

1. INTRODUCTION

Recent years have witnessed the drastic development of World Wide Web (WWW). Information is being accessible at the finger tip anytime anywhere through the massive web repository. The performance and reliability of web engines thus face huge problems due to the presence of enormous amount of web data. The voluminous amount of web documents has resulted in problems for search engines leading to the fact that the search results are of less relevance to the user. In addition to this, the presence of duplicate and near-duplicate web documents has created an additional overhead for the search engines critically affecting their performance [1]. The demand for integrating data from heterogeneous sources leads to the problem of near-duplicate web pages. Near-duplicate data bear high similarity to each other, yet they are not bitwise identical [2][3]. The near-duplicate web pages either increase the index storage space or increase the serving costs which annoy the users, thus causing huge problems for web search engines. The existences of near-duplicate web page are due to exact replica of the original site, mirrored site, versioned site, and multiple representations of the same physical object and plagiarized documents. In many situations, two documents that are not exactly indistinguishable may still contain the same content and should be treated as near-duplicates. For example, web pages from different mirrored sites may only differ in the header or footnote zones that denote the site URL and update time. Two such documents are identical in terms of content but differ in a small portion of the document such as advertisements, counters and timestamps. These differences are irrelevant for web search [4].

Near-duplicate detection has been recognized as an important research problem in recent years. Several applications are benefited by the identification of the near-duplicate detection in the field of plagiarism detection, spam detection and in focused web crawling scenarios. In plagiarism detection, a portion of one document, such as a sentence or a paragraph, is contained in another document, and then these two documents could be seen as near-duplicates. Spam messages that belong to the same campaign may look very different because spammers often need to randomize the messages by modifying it with obscure terms or adding unrelated paragraphs to pass the filters [5][3]. However, these spam could be easily discovered by near-duplicate detection methods. The determination of near-duplicate web pages aids in focused crawling, and ensures enhanced quality and diversity of query result.

In this paper, we propose a novel idea for finding near-duplicates of an input web-page, from a huge repository. This approach explores the semantic structure, content and context, of a web page rather than the content only approach. The weighting scheme suggested in [6] is considered for creating a term-document-weight matrix (TDW) which plays an important role in the proposed algorithm. We present a three-stage algorithm which receives an input record and a threshold value and returns an optimal set of near-duplicates. In first phase, rendering phase, all pre-processing are done and a weighting scheme is applied. Then a global ordering is performed to form a term-document-weight matrix. In second phase, filtering phase, two well-known filtering mechanisms, prefix filtering and positional filtering [2], are applied to reduce the size of competing record set and hence to reduce the number of comparisons. In third phase, verification phase, singular value decomposition is applied and a similarity checking is done based on the threshold value and finally we get an optimal number of near-duplicate records.

The rest of this paper is organized as follows. Section 2 describes related works. Section 3 gives the details of proposed work. Section 4 analyses the experimental results in terms of precision and recall. In the last section, we give the conclusion and future works.

2. RELATED WORKS

The original idea of this work has evolved while developing an innovative approach for effective optimal feature subset selection for web page categorization. The subsistence of near-duplicate data is an issue that accompanies the growing need to incorporate heterogeneous data. Web content mining face huge problems due to the presence of duplicate and near-duplicate web pages.

Very few papers have suggested methodologies for near-duplicate detection both in general documents and the web documents obtained by web crawling. Technique for the estimation of the degree of similarity among pairs of documents is known as *shingling*. Broder et al. [7] have suggested a technique on this, 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. In this method the authors noted that it does not work well on small documents. Fetterly et.al. [8] 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. Andrei Z. Broder et al. [7] have developed an efficient way to determine the syntactic similarity of files and have applied it to every document on World Wide Web. Using their mechanism, they have built a clustering of all the documents that are syntactically similar.

Another method to detect and delete near-duplicated web pages; priority-based on text information, is proposed [9]. By this method, an algorithm to extract text information of web pages by DOM tree and a priority-based algorithm for detecting near-duplicated text information are implemented, so as to reduce the noise of web pages and hence to improve the efficiency of detecting near-duplicated text information.

Narayana et al. [10] presented an approach for the detection of near-duplicate web pages in web crawling. Near-duplicate web pages are detected followed by the storage of crawled web pages in to repositories. The keywords are extracted from crawled pages and on the basis of these keywords; the similarity score between two pages is calculated. The documents are considered as near-duplicates if its similarity scores are lesser than a threshold value. But they haven't exploited the structural information of web-page which is more important for getting the semantic information about the page.

3. PROPOSED WORK

Here we propose an innovative idea for finding near-duplicate web pages of an input record *r*. Similarity verification is done on a huge record set having *n* number of records $\{r_1,r_2,\ldots,r_n\}$ and an optimal set of near-duplicate records are returned. The similarity verification is based on a Jaccard threshold value *t*, $0 \le t \le 1$, such that our algorithm returns a set of records having a minimum similarity *t*. $t=0$ returns full record set which shares at least a single word with input record *r*, while $t=1$ return only exactly similar records of *r*. Our objective is to find how to select similar records from the entire record set with a reduced number of comparisons. A three-stage selection algorithm is proposed by combining a different term weighting approach [6], term filtering approach and standard singular value decomposition techniques for this purpose.

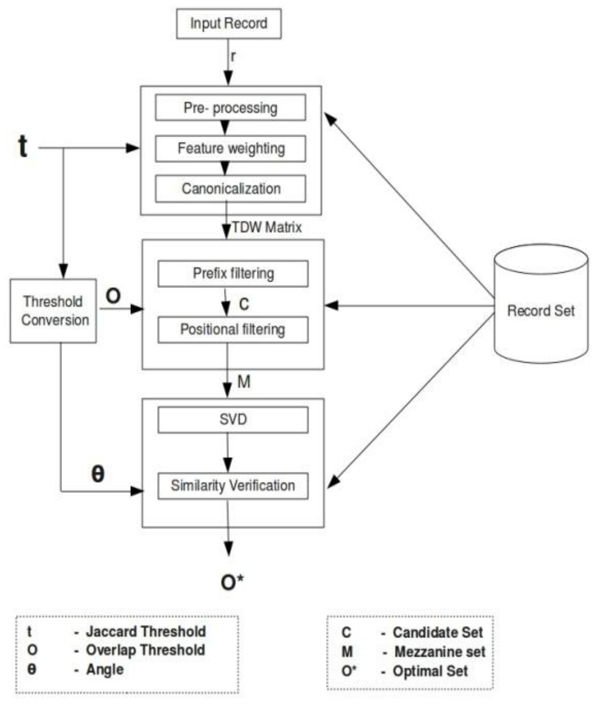

Fig 1: General Architecture

In the first phase, rendering phase, standard web page pre-processing methods like *html* tag removal, tokenization, removal of stop words and stemming are applied on the input record *r* and a feature set *F* of *m* tokens $\{x_1,x_2,\ldots,x_m\}$ are retrieved. Then a standard weighting scheme *W* is proposed, based on the term field where the term *x* is present in a record. Even though a common term is present in two web pages, it not only depends on the number of occurrences in each page, but also depends on the field in which it is present, since a web page is entirely different from a normal text file. For example, if a word is present in the title of one web page and in the content block of another web page, they differ by significance. In the first document, it may be a main feature while in the second one it has got a less importance. Number of occurrences of each token x is multiplied with the weight of respective term fields w_i wherever the term is present and added together for total weight of a term W_x. Further this weight is normalized based on the total weight of the record, W_r using some standard approaches. This is done in perspective of the global ordering to be used in the upcoming stages. Obtained feature set along with its weights is compared with the features of other records present in the repository for measuring the similarity.

Table 1. Proposed Weighting Scheme

Term Field	Weight
URL	2
Heading	2
Title	2
Anchor Text – To the same web site	1
Anchor Text – To a different web site	0.5
Keywords	3
Description	3
Main content block	1

If a term x is present in a record r, its total weight is represented as W_{xr}. Here we apply some filtering mechanisms which further reduce the size of candidate set and improve the efficiency. Prior to this, we have to perform a global ordering based on an inverted index used in vector space model (VSM). The document frequency of a token is the number of records that contain the token. We can canonicalize a record by sorting its tokens based on a global ordering. A document frequency ordering arranges the entire tokens according to the increasing order of its document frequency. The next step is to create a term-document-weight matrix (TDW) that maps a token x to a list of records that contain x and the total weight of the token in each record. If a particular token is not present in a record, its weight is considered as zero. Each record is represented as per the global ordering.

Let r_1, r_2, r_3 be three canonicalized records. $r_1=\{x_2, x_1, x_3\}$ $r_2=\{x_4, x1, x_3\}$ $r3=\{x_2, x_4, x_1, x_3\}$ The TDW matrix is given in Table 2.

In filtering phase, two filtering mechanisms, prefix filtering and positional filtering, are performed to reduce the number of candidate records. Prefix length of r is calculated as**:**

$$\textit{Prefix length} = |r| - \lceil t.|r| \rceil + 1 \qquad (1)$$

Table 2. Sample TDW Matrix

records / terms	r_1	r_2	r_3
x_2	W_{x2r1}	0	W_{x2r3}
x_4	0	W_{x4r2}	W_{x4r3}
x_1	W_{x1r1}	W_{x1r2}	W_{x1r3}
x_3	W_{x3r1}	W_{x3r2}	W_{x3r3}

by assuming the threshold value t as Jaccard similarity threshold. Each token in prefix set of r is compared with prefix set of all records in the repository and if any record r_i is sharing a token with r in its prefix set, it is added to candidate set C. The basic idea behind prefix filtering principle is that if two web pages share rare tokens, there is a chance that it might be similar. Since we are doing a global ordering based on document frequencies, prefix set of a record contain rare tokens. If no tokens are shared in prefix set, that record can be avoided from further processing. Once prefix filtering is over, positional filtering principle is applied in order to prune unwanted records from candidate set C.

Chuan Xiao et. al.[2] observed that positional information can be exploited in several ways to further reduce the candidate size. Position of each token in a record can be counted, starting from one, which gives information about the upper bounds of overlapping in which Jaccard threshold t can be stated in terms of overlap threshold O as

$$J(r,r_i) \geq t \Leftrightarrow O(r,r_i) \geq \frac{t}{1+t}(|r|+|r_i|) \qquad (2)$$

and upper bounds of O can be calculated as

$$\textit{ubound}=1+\textit{min}(|r|-p, |r_i|-q) \qquad (3)$$

where record r shares a term at p^{th} position with another record r_i at position q. If *ubound* satisfies overlap threshold O, record r_i can be added into the *mezzanine set M* from where an *optimal set* is extracted.

Based on the records from mezzanine set M, we create a weight matrix A such that columns represent documents and rows represent terms. An element a_{ij} represents the weight of the global feature x_i in record r_{j-1} since the first column represents input record r. In verification phase, singular value decomposition is applied on weight matrix A and each record can be represented as a vector in 2D space. Then Jaccard threshold $0 \leq t \leq 1$, can be mapped into an angle $180 \geq \theta \geq 0$ accordingly, using the formula

$$\theta = 180 * (1 - t). \qquad (4)$$

We can say that two records are purely dissimilar when the angle between them is 180 and they are exactly similar if it is 0. Ultimately we get an optimum set of records by analyzing the angle of a document with respect to input record r. If it satisfies the threshold θ, it can be marked as a near- duplicate of r and ranked on the basis of angle.

3.1 Proposed Algorithm

3.1.1 Algorithm: Near_Duplicate_Detection

Input: A Web page (*Web_Document*), records in repository (*Record_Set*) and Jaccard threshold (t)
Output: Optimal near-duplicate record set, O^*
Remarks: $M \rightarrow$ Mezzanine record set

```
        $TDW\_Matrix \rightarrow$ term-document-weight matrix

Near_Duplicate_Detection ($Web\_Document, Record\_Set, t$)
    $TDW\_Matrix \leftarrow$ Rendering ($Web\_Document, Record\_Set$);
    $M \leftarrow$ Filtering($TDW\_Matrix, Record\_Set, t$);
    $O^* \leftarrow$ Verification($M, Record\_Set, t$);
    return $O^*$;
```

3.1.2 Algorithm: Rendering

```
Input:    $Web\_Document, Record\_Set$
Output:   $TDW\_Matrix$
Remarks:  $W_x \rightarrow$ total weight of the term $x$

Rendering ($Web\_Document, Record\_Set$)
        $Input\_Record \leftarrow$ Pre_Processing($Web\_Document$);
        $F \leftarrow$ Full feature set($Input\_Record$);
        for all $x_i \in F$
                $W_x \leftarrow$ Weight_Scheme($x_i$);
        $W_r \leftarrow \sum W_x$;
        for all $i$, $1 \leq i \leq |F|$
                $W_x \leftarrow$ Normalize($W_x, W_r$);
        $T \leftarrow$ Thresholding($W_r$);
        $r \leftarrow \varphi$;
        for all $x_i \in F$
                if ($W_x \geq T$)
                        $r \leftarrow r \cup x_i$;
        $TDW\_Matrix \leftarrow$ Canonicalize($r, Record\_Set$);
        return $TDW\_Matrix$;
```

3.1.3 Algorithm: Filtering

```
Input:    $TDW\_Matrix, Record\_Set, t$
Output:   $M$
Remarks:  Assume that $Input\_Record$ is represented as the first
          entry in $TDW\_Matrix$

Filtering ($TDW\_Matrix, Record\_Set, t$)
    $r \leftarrow TDW\_Matrix[1]$;
    //prefix filtering
    $C \leftarrow \varphi$;
    $Prefix\_Length \leftarrow |r| - \lceil t.|r| \rceil + 1$;
    for all $r_i \in Record\_Set$
        $Prefix_i \leftarrow |r_i| - \lceil t.|r_i| \rceil + 1$;
        for all $j,k$; $1 \leq j \leq Prefix\_Length$, $1 \leq k \leq Prefix_i$
                if ($r[j] == r_i[k]$)
                        $C \leftarrow C \cup r_i$;
    //positional filtering
    $M \leftarrow \varphi$;
    for all $r_i \in C$
        $O \leftarrow \frac{t}{1+t}(|r|+|r_i|)$;

        for all $p,q$; $1 \leq p \leq Prefix\_Length$, $1 \leq q \leq Prefix_i$
                if ($r[p] == r_i[q]$)
                        $ubound \leftarrow 1 + \min(|r|-p, |r_i|-q)$;
                        if ($ubound \geq O$)
                                $M \leftarrow M \cup r_i$;
    return $M$;
```

3.1.4 Algorithm: Verification

```
Input:    $M, Record\_Set, t$
Output:   $O^*$

Verification($M, Record\_Set, t$)
        $O^* \leftarrow \varphi$;
        $\theta \leftarrow 180(1-t)$;
        $Coordinates \leftarrow$ SVD($M, Record\_Set$);
        Plot($Coordinates$);
        for each plot of $r_i \in M$
                $\theta_i \leftarrow$ Measure_Angle($r, r_i$);
                if ($\theta_i \geq \theta$)
                        $O^* \leftarrow O^* \cup r_i$;
        return $O^*$;
```

4. EXPERIMENTAL RESULTS

4.1 Data Collection

We have generated a huge repository of web page records from Google search engine. It was done for some specific query words given to Google and collected all similar pages with respect to the first ranked result of Google. Some of these pages were removed due to the lack of required contents retrieved and non-suitable file formats like PDF. Each page thus obtained is pre-processed, featured and weighted according to the weighting scheme and properly indexed to create a TDW matrix. This procedure was repeated for 10 different queries and experiments were conducted on 10 different repositories thus created. Each experiment is resulted an optimal set of near-duplicate web pages with respect to the first ranked web page in the query result.

4.2 Experimental Set up

To conduct required experiments, we created an online tool which retrieves the contents of an input web page; perform all pre-processing steps and extract weighted feature set. With respect to the huge repository created earlier, a TDW matrix is formed in a global ordering and filtering principles are applied. All these steps were implemented using PHP. The resultant matrix thus obtained is supplied to MATLAB for further processing. This matrix is then decomposed into 2D coordinates using singular value decomposition techniques [11][12]. Coordinates of each record is then plotted in a 2D vector space and the angle between each record with input record is measured. If it satisfies the threshold value θ, that record is marked as a near-duplicate one.

4.3 Result Analysis

To evaluate the degree of accuracy, efficiency and scalability of our algorithm, we have used two standard benchmark measures, precision and recall.

$$Precision = \frac{\text{Number of web pages detected correctly}}{\text{Total number of near\_duplicate web pages detected}}$$

$$Recall = \frac{\text{Number of web pages detected correclty}}{\text{Total number of near\_duplicate web pages}}$$

The measures obtained for 10 different repositories are given by:

Table 3: Performance Measures

Query word	No of near-duplicates	Precision %	Recall %
Q_1	167	94.7	93.4
Q_2	98	95.1	94.7
Q_3	156	95.7	93.2
Q_4	123	93.1	92.8
Q_5	79	96.4	95.1
Q_6	129	94.9	93.4
Q_7	63	93.5	92.0
Q_8	112	95.0	92.5
Q_9	109	94.6	93.0
Q_{10}	59	95.9	92.9
Average		**94.9**	**93.3**

While creating a repository of 50 pages, a TDW matrix of almost size 3000 x 50 is created in rendering phase. After filtering phase, its size was reduced to 2500 x 6 when a threshold *t*=0.5 is applied. And finally, a set of three near-duplicates $\{d_1, d_2, d_3\}$ are returned.

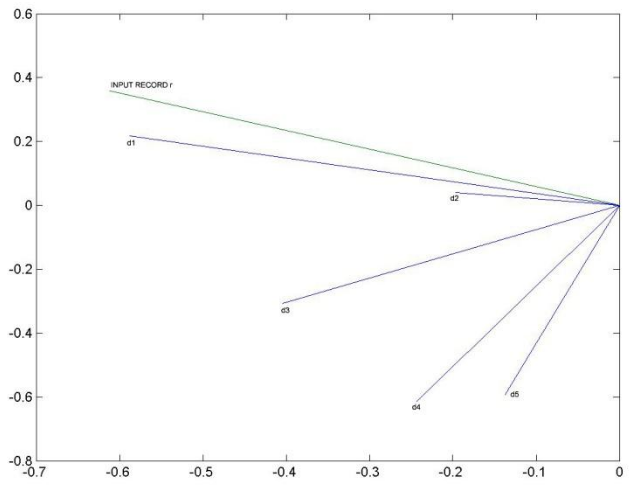

Fig 2: Vector Space Representation of Documents

All the results obtained during the experiments were having a good level of similarity both in terms of syntax and semantics. Weighting scheme plays an important role in semantic analysis while majority of existing works depend only on simple term frequency based approaches or boolean approaches. Shingle based approaches check only the number of overlapped shingles in the pair which is purely a syntax based one. Here, by measuring the angle in 2D vector space, it can predict the optimal set of near-duplicates instantly.

5. CONCLUSION AND FUTURE WORK

This paper proposed a novel task for finding near-duplicate web pages. The proposed technique aims at helping document classification in web content mining based on TDW matrix. We have also used a system based on feature weighting to extract the features using a different term weighting approach which explore the semantic information of a web page with a good level of accuracy. Instead of simply using the traditional cosine similarity for finding similar pages, we proposed a three stage algorithm which reduces the size further. In this paper, we focus a TDW matrix based method that can be applied for generating an optimal near-duplicate set. We evaluate the accuracy and scalability of our algorithm using two standard benchmark measures, precision and recall. The experiments proved that our work has a better performance than other methods.

We believe that this work represents an important step towards this direction and it is a promising method for near-duplicate detection which contributes more for web content mining. Further research works can extend this to a more efficient method for finding similarity joins which can be incorporated in a focused web crawler.

6. ACKNOWLEDGMENT

The authors thank the authorities of their working organization and management for their support and the encouragement to pursue research in the chosen field of study.

7. REFERENCES

[1] Fetterly D, Manasse M, Najork M, On the evolution of clusters of near-duplicate Web pages, In Proceedings of the First Latin American Web Congress, pp.37- 45 Nov. 2003.

[2] Chuan Xiao, Wei Wang, Xuemin Lin, Efficient Similarity Joins for Near-Duplicate Detection, Proceeding of the 17th international conference on World Wide Web, pp 131 – 140. April 2008.

[3] Gurmeet Singh Manku, Arvind Jain and Anish Das Sarma, Detecting near-duplicates for web crawling, In Proceedings of the 16th international conference on World Wide Web, pp. 141 - 150, Banff, Alberta, Canada, 2007.

[4] Dennis Fetterly, Mark Manasse and Marc Najork, Detecting phrase-level duplication on the world wide web, In Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, pp.170 - 177, Salvador, Brazil, 2005.

[5] D. Lowd and C. Meek, Good word attacks on statistical spam filters, Second Conference on Email and Anti-Spam, July 2005.

[6] Shine N Das, Midhun Mathew, Pramod K.Vijayaraghavan, An Approach for Optimal Feature Subset Selection using a New Term Weighting Scheme and Mutual Information, Proceeding of the International Conference on Advanced Science, Engineering and Information Technology, Malaysia, 2011, pp 273-278, January 2011.

[7] Broder, A., Glassman, S., Manasse, M., and Zweig G. Syntactic Clustering of the Web, In 6th International World Wide Web Conference, pp: 393-404, 1997.

[8] Fetterly, D., Manasse, M. and Najork, M. On the evolution of clusters of near-duplicate web pages, In Proceedings of

the first Latin AmericanWeb Congress (LAWeb), 37–45, 2003.

[9] Yun Ling, Xiaobo Tao Hexin Lv, A Priority-Based Method Of Near-duplicated Text Information Of Web Pages Deletion, IEEE International Conference on Software Engineering and Service Sciences (ICSESS), August 2010.

[10] 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.

[11] Jody S. Hourigan and Lynn V. McIndoo, A scientific Report on Singular Value Decomposition, 1998

[12] Shine N Das, K. V. Pramod, Relevancy based Re-ranking of Search Engine Result, Proceedings of International Conference on Mathematical Computing and Management, Kerala, India, June 2010.