International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 4 Issue 07, July-2015

An Extended TDW Scheme by Word Mapping Technique

Arun P. R.
PG Scholar,
Computer Science and Engineering
Adi Shankara Institute of Engineering and Technology
Kalady, India

Sumesh M. S.
Assistant Professor,
Computer Science and Engineering
Adi Shankara Institute of Engineering and Technology
Kalady, India

***Abstract*—In the recent years there is a large growth in web contents over the internet. The internet does not provide any standard mechanism for verification of web contents before hosting them in web servers, which cause to increase the near and exact duplicated contents over the internet from heterogeneous sources. These duplicate contents can exist either intentional or accidental. The problem of finding near-duplicate web pages has been a subject of research in the database and web-search communities for some years. Since most prevailing text mining methods adopted term-based approaches, they all suffer from the problems of word synonymy and large number of comparison. In this paper, we are going to deal with the detection of near and duplicate web pages detection by using term document weighting scheme, sentence level features and addressing the synonym detection. The existence of these near and duplicate web pages causes the problems that ranges from network band width utilization, storage cost, reduce the performance of search engines by duplicated content indexing, increase load on a remote host.**

Keywords*— *Near Duplicate Detection, Sentence TDW Matrix, Stemming, Word mapping

I. INTRODUCTION

Web Mining is the branch of data mining which deals with the study of World Wide Web [1]. It refers to the use of data mining techniques to automatically find out and mine information from World Wide Web documents and services. In every second millions of bytes are added all over the world. As the size of data is increasing there should be a mechanism in order to find the duplicate and nearly duplicate contents in internet. The existence of the same contents in multiple times and in various format will lead to a wide array of problems that ranges from basic storage ,to network bandwidth utilization, to search result quality, to load on a remote host etc.

Most people depend on the search engines for finding the required information. The existence of nearly duplicated contents frustrate the user by returning duplicated contents as search results, thereby lead to use more bandwidth for transferring these contents from a remote server which again lead to make unnecessary load on a remote machine. There will not be any benefit of keeping the nearly duplicated contents in multiple host, which demands more space for storage [2]. The existence of the nearly duplicated contents can either accidental or intentional. The reason of the existence these contents are due to the absence of a standard mechanism for developers in order to ensure the existence of their webpage contents, which cause the accidental content occurrence. Intentional duplicate contents arise for web spamming to get a higher ranking position by keyword stuffing or to make doorway pages. Search engines are suffers from indexing of nearly duplicated contents which reduces the quality of result.

For a method to detect the nearly duplicated web page in addition to address the problems such as synonym detection, rank based outcome, dimensionality of document representation, reduced number of comparison should consider the following, existence of local noise, html tag based content representation, URL in the page, heterogeneous sources.

The recognition of similar or near-duplicate documents in a huge collection is a momentous problem with extensive applications. This is certainly demanding in the web-scale due to the voluminous data and high dimensionalities of the document [3].Due to high rates of duplication in the web document the need for detection of duplicated and nearly duplicated document is high in diverse applications like crawling [4], ranking [5], clustering [6], archiving and caching [7].

II. STATE OF ART

There are research papers have suggested methodologies for near duplicate detection both in general documents and the web documents obtained by web crawling. The representation of document content is an important factor because which represents as features of that particular document for further comparison with other documents.

In Boolean model, a document term weight is represented by either zero or one. The occurrence of a term represented by one and its absence by zero. But this model fails to rank the similar documents outcomes due to its equal weighting scheme for all terms .This model has extended by Gerard Salton *et al.*[8]. The check summing approaches are used to find out the web pages that are exact duplicates of each other due to mirroring [9].

Broder et.al [1] has proposed a method to find syntactic similarity of files based on shingles, shingles is the word sequence of adjacent words shingles are generated for all documents after tokenizing it. Similarity of files calculated by the common shingle occurrence among them. In this method the authors noted that it does not work well on small documents.

I-Match relies on collection statistics which proposed by Abdur Chowdhury et.al [10]. A hash value for each document is computed using SHA1 algorithm. For each document generates a pair <doc_id,hash value>. Duplicate files is detect by a collision while inserting the pair into a tree or a hash table. This method applicable to the detection of exact match in a large dataset

A signature based approach proposed by Martin Theobald et.al [11].The extracted pattern from a document is termed as a signature. A spotsigs at a location sj in a document is represented by aj(dj , cj) which defines an antecedent word(aj),spot distance (dj) and a contiguous spot chain of cj word.

MaoshengZhonget.al [12] introduced a practical approach for relevance measure of inter-sentence. The method focused in extracting the interior meaning of sentence. When considering a web page It is not possible to focus only in the main content, it might contains the advertisement, banner contents, links into other web pages. Even the main content is same the existence of noise will affect the performance of this method.

Midhun Mathew et.al [13] proposed a novel approach for near-duplicate detection of web pages using TDW Matrix. This method uses a new weighting scheme for terms which based on the tag where the content is belongs. Each html tag is pre assigned with a weight, then the term weight is calculated by multiplying term frequency and tag weight.

III. METHODOLOGY

The proposed approach uses the term document weight (TDW) scheme for the detection of near and duplicate web pages, because a web page content is completely different from an ordinary text file so the relevance of a term is varying not only based upon the frequency of term appeared but also where it present in the document. Consider an example where a shared term is present in two web documents. In the first document it present in the title tag where as in the second page it present in the content block, the relevance of the term in these two documents are different.

The proposed method consist of four phases preprocessing, word mapping, comparison reducer and cosine similarity. The preprocessing stage takes an input web page. First step is to extract the web page content based on the html tags, then extract sentence count from web page by parsing each tag contents then perform stop word removal. The stop words are consider to be the connectives and preposition in English language. The resulting string content is tokenized.

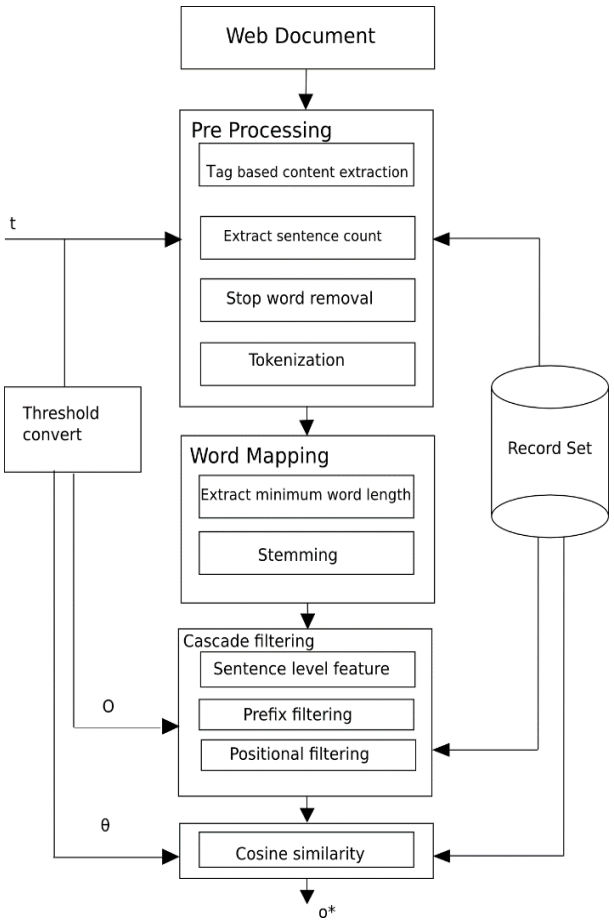

Fig 1: Proposed Architecture

The word mapping phase receives the tokenized content and for each word having a synonym extract the row of its synonyms and returns the minimum length word for stemming. For example a word *x3* is having a synonym then extract the first synonym field value for *x1* from the particular row, the first field will be minimum length synonym term for *x3*.

TABLE I. SYNONYM TERMS

Id	*Synonym1*	*Synonym2*	*Synonym3*
1	*x1*	*x2*	*x3*
2	*y1*	*y2*	*y3*

In many cases, morphological alternatives of words have similar semantic interpretations and can be considered as equivalent for the purpose of many applications. Porter's stemming algorithm is used to map the term into its root form. Compute the TDW matrix for the web page based on the term appearing frequency and tag weighting. Letrec1, rec2, rec3 be three records. rec1= {t2, t1, t3} rec2= {t4, t1,

t3} rec3={t2, t4, t1, t3}.The notation fti be the number of times the term appear in a specific tag. The TDW matrix with sentence feature is given in Table 2.

TABLE II. SAMPLE TDW MATRIX WITH SENTENCE FEATURE

Terms / Records	Sentence Feature	t1	t2	t3	t4
rec1	c1	$\sum ft1*w_t$	$\sum ft2*w_t$	$\sum ft3*w_t$	0
rec2	c2	$\sum ft1*w_t$	0	$\sum ft3*w_t$	$\sum ft4*w_t$
rec3	c3	$\sum ft1*w_t$	$\sum ft2*w_t$	$\sum ft3*w_t$	$\sum ft4*w_t$

$w_t$represent the tag weighting scheme. Here we use the weighting scheme [15].

TABLE III. WEIGHTING SCHEME

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

Comparison reducer phase is used to reduce the number of record comparison. In this phase three filtering mechanisms are used. The first step in filtering is done based on the number of sentences. The number of input web page sentenceS_k is compared with the number of sentences in the record setS_{di}. The web pages which satisfy the sentence feature difference within a threshold bound will consider

$$|S_{di} - S_k| < T_s \quad (1)$$

The resultant web pages are filter by prefix and positional filtering [14] mechanism.

$$\text{Prefix length} = |r| - \lceil t.|r| \rceil + 1 \quad (2)$$

By assuming the threshold value *t* as Jaccard similarity threshold. Each term in prefix set of record*r* is equated with prefix set of all records in the repository and if any record *ri*is sharing a term with *r* in its prefix set, it is added to pre-final set P_{fs}. The records should be in global ordering while taking the prefix set. The basic idea behind prefix filtering principle is that if two web pages share infrequent terms, there is a chance that it might be similar. If there is no terms are common in prefix set, that record can be evaded from further processing. Once prefix filtering is over, positional filtering principle is applied in order to prune unwanted records from pre-final set P_{fs}.

Positional information can be exploited in several ways to further reduce the pre final set size. Position of each term 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

$$J(r,ri) \geq t \Leftrightarrow O(r,ri) \geq (|r|+|ri|) \quad (3)$$

Upper bounds of *O* can be calculated as

$$u\_bound = 1 + min(|r|-p, |ri|-q) \quad (4)$$

Where record *r* shares a term at p^{th} position with another record *ri*at position *q*. If *ubound*satisfies overlap threshold *O*, record *ri* can be added into the *final set F* from where an *optimalset* is extracted. Based on records from the final set F having a matrix M with rows be the term weights and attribute name is the stemmed word.

Final phase is to find the cosine similarity of input record set with the final record set F.TheJaccard threshold $0 \leq t \leq 1$,can be mapped into an angle$180 \geq \theta \geq 0$ accordingly, using the formula

$$\theta_t = 180*(1 - t) \quad (5)$$

Let *tw1, tw2,…,twn* be the input record term weight set and the remaining rows be the outcome of filtering

$$M = \begin{bmatrix} tw1 & tw2 & . & . & twn \\ x_1 1 & x_1 2 & . & . & x_1 n \\ . & . & . & . & . \\ . & . & . & . & . \\ x_m 1 & x_m 2 & . & . & x_m n \end{bmatrix} \quad (6)$$

The first row be the term weighs of the input record and the remaining rows be the term weights of final set F. Each column weight represent the weight for the same term, if any term is absent then the weight will be zero.

$$\text{Cos } \theta = \frac{tw1.x_i1 + tw.x_i2 + \cdots + twn.x_in}{\sqrt{tw1^2 + tw2^2 + \cdots + twn^2}.\sqrt{x_i1^2 + x_i2^2 + \cdots + x_in^2}} \quad (7)$$

Obtain the ɵ by $\cos^{-1}$ measure and compare with the $\theta_t$to obtain the final near and duplicate set of web pages.

IV. PROPOSED ALGORITHM

A.Algorithm : Near_Duplicate_Detection

Input: Input Web page W,Record set R_i,jaccard threshold t

Output: Final near duplicate web pages from repository (o*).

1. Sentence_TDW_Feature =Preprocessing (W);

2.P_{fs}=Comparison_reducer(Sentence_TDW_Feature(W), Sentence_TDW_Feature(Ri),t);

3. θ_t=180*(1 – t);

4. Fs=Cosine_Similarity(P_{fs}, θ_t);

5. Return sort(Fs);

B.Algorithm: Pre-processing

Input: Input Web page W

Output: Sentence_TDW_Matrix

1. E_content=tag_based_content(W);

2. T_sentences=T_sentences+s_count(E_content);

3. Content=stop_word_removal(E_content);

4. T[]=Tokenize(Content);

5. For j=0 to |T|

5.1 Wm_c=WordMapping(T[j]);

5.2 Sentence_TDW_Matrix(W,Wm_c)=Sentence_TDW_Matrix(W,Wm_c)+(count(Wm_c)*tag_weight);

6. End

7 Sentence_TDW_Matrix(W,Wm_c).append(T_sentence);

C:Algorithm WordMapping

Input: Tokenized term Ti

Output: Stemmed term

1. If(exist(Ti).synonym)

2. Extract_synonym[]=Synonym(Ti);

2.1 Ti =Minimum_length(Extract_synonym[]);

3. End

4. Return porter_stem(Ti);

D:Algorithm Comparison_reducer

Input: Sentence_TDW_Feature(W),Sentence_TDW_Feature(Ri),Jaccard threshold t

Output: Pre final of Sentence_TDW_Feature(Ri)

1. P_len(W)= (Sentence_TDW_Feature(W).length-1)-ceil(t* Sentence_TDW_Feature(W).length-1)+1;

2. S_count= Sentence_TDW_Feature(W).T_sentences;

3. W_slice=Slice(Sentence_TDW_Feature(W),p_len(w));

4. For i=0 to |Ri|

4.1 S_count(i)= Sentence_TDW_Feature(i).T_sentences;

4.2If (|S_count(i)- S_count| <Ts)

4.2.1 P_len(i)= (Sentence_TDW_Feature(i).length- 1)-ceil(t* Sentence_TDW_Feature(i).length-1)+1;

4.3 W_slice(i)=Slice(Sentence_TDW_Feature(i),p_len(i));

4.4 If(exist((Sentence_TDW_Feature(w).attribute_name),(Sentence_TDW_Feature(i).attribute_name)))

4.4.1 P=array_pos(Sentence_TDW_Feature(w));

4.4.2 Q= array_pos(Sentence_TDW_Feature(i));

4.4.3 ot=(t/(1+t))*(|Sentence_TDW_Feature(W)|+|Sentence_TDW_Feature (i)|);

4.4.4u_bound=1+min(|Sentence_TDW_Feature(W)|-p, (|Sentence_TDW_Feature (i)|-q);

4.5 If(u_bount>= ot)

4.6 Pfs[]=Sentence_TDW_Feature(i);

5 End

6 End

7 End

8 Return Pfs;

E.Algorithm	*Cosine_similarity*

Input: Pre-final set (Pfs), Sentence_TDW_Feature(W)

Output: Final set of near duplicate web pages (F)

1. For i=0 to |Pfs|

   1.1 Cos_angle[]=$\cos^{-1}$(Sentence_TDW_Feature(i), Sentence_TDW_Feature(W));

2. End

3. Return sort(Cos_angle);

V. EXPERIEMENT AND RESULT

A. Data Collection

The required dataset was collected from Google search engine. Created a repository of web pages obtained by querying some specific keywords and collected all similar web pages with respect to the higher rank position. Some of the result were omitted due to the lack of required contents retrieved and based on required file formats. Each page thus obtained is pre-processed, featured and weighted according to the weighting scheme and properly indexed to create a sentence TDW matrix. This procedure was repeated for ten different queries and experiments were conducted on ten 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.

B. Experimental Setup and Result

To conduct the required experiment, we created an online tool which is capable of extracting features form web page either by giving a URL or by upload the web page from the local system. The system build a sentence TDW matrix for the input web pages by applying stop word removal, extractingsentence feature and stemming. Applied the filtering principles and cosine similarity to tag a record or a web page as a near duplicate one. All these phases are implemented in PHP. The resultant information has used to plot the graph showing the retrieval status.

Table 4 shows a sample content processing in the preprocessing phase. The output of this phase will used to build Sentence TDW matrix. The matrix formed by arranging the term as attribute names. The weights are inserted into the column with respect to the record entry shown in fig2.

TABLE IV. PREPROCESSING

Steps	Obtained output
Tag based content extraction	In the recent years there is a massive development in the web pages, there are billions of web pages existing in the search engine which decreases the efficiency and effectiveness of the search results of the search engine.
Sentence feature	1
Stop word removal	recent years massive development web pages billions web pages existing search engine decreases efficiency effectiveness search results search engine
Tokenization	recent, years, massive, development, web, pages, billions, web, pages, existing, search, engine, decreases, efficiency, effectiveness, search, results, search, engine
Word mapping	recent,ages,bulky,development,web,side,existing,seek,motor,drop, efficiency, effectiveness, seek,results,seek,motor
Stemming	recent, ag. bulki, develop, web, side, exist, seek, motor, drop, effici, effect, seek, result, seek, motor

rec_no	rec_url	sentence_feature	bake	bread	chocol
1	uploads/cc_22.html	18.00	3.33	1.54	0.79
2	uploads/cc_A2Z Healthy Vegetarian Cuisine.html	19.00	0.00	1.51	NULL
3	uploads/cc_basic eggless chocolate cake recipe, whole w...	20.00	1.40	0.00	1.17
4	uploads/cc_Basic Vanilla Cake Recipe Food Network Kit...	16.00	0.00	NULL	0.00
5	uploads/cc_Basic Vanilla Cake Recipe - Simple Recipe for...	22.00	NULL	NULL	NULL
6	uploads/cc_Buy awesome cookware on www.ekitchen.in....	17.00	NULL	NULL	NULL
7	uploads/cc_Cakes _ Tasty Food Tips.html	22.00	3.00	NULL	0.00
8	uploads/cc_Cakes and Cookies - Part 3 _ the brown eyed ...	19.00	NULL	NULL	NULL
9	uploads/cc_Date Cake (Eggless) â€¦ Aayis Recipes.html	18.00	NULL	NULL	NULL
10	uploads/cc_Eggless fruit cake â€¦ Aayis Recipes.html	22.00	1.26	0.00	0.00

Fig 2: Sentence TDW Matrix

C. Result Analysis

For evaluating the degree of accuracy, efficiency and scalability of our proposed approach, we have used repository that contains the webpages documents obtained through querying in Google search engine. The performance of the proposed approach is evaluated with the help of evaluation metrics such as, Precision, Recall.

$$\text{Precision (P)} = \frac{Number\ of\ web\ pages\ detected\ correctly}{Total\ number\ of\ near\ duplicate\ web\ page\ detected} \quad (8)$$

$$\text{Recall(R)} = \frac{Number\ of\ web\ pages\ detected\ correctly}{Total\ number\ of\ near\ duplicate\ web\ pages} \quad (9)$$

TABLE V. PERFORMANCE MEASURE

Query Word	No.of near-duplicates	Precision%	Recall %
Q1	128	96.9	94.81
Q2	67	95.7	93.05
Q3	72	96.0	94.73
Q4	112	96.55	94.91
Q5	109	96.46	94.78
Q6	81	95.29	94.18
Q7	143	95.97	95.33
Q8	87	94.56	91.57
Q9	98	96.07	93.33
Accuracy		**95.94**	**94.07**

While creating a repository of 100 pages, a TDW matrix of almost size 100 x 1700 is created in preprocessing phase. When the word mapping scheme is applied the term count reduced to almost 1400. The sentence feature comparison greatly reduce the record size into 14 , as by continuing to prefix and positional filtering and final cosine similarity, the resultant near duplicates is three records.

VI. CONCLUSION

Near-duplicate web pages stance a serious problem to the web crawling and have become the main concern for the web search engines. Near duplicates rise the cost of giving results, suffer large amount of space to store the indexes and ultimately slows down the result, hence affecting both the accuracy, time for execution and frustrate the user. There has proposed a number of algorithms designed for the detection of near duplicate detection based on similarity scores and signatures. In this paper, we have proposed a four phase efficient method for detecting the near duplicates using the sentence level features, words mapping technique and the term document weighting scheme. The work includes a cascade filtering techniques. The experimental results have proved that the proposed approach is efficient and having improved precision and recall. The accuracy and scalability of our algorithm using two standard benchmark measures, precision and recall.

REFERENCES

[1] Andrei Z. Broder, Steven C. Glassman, Mark S. Manasse, and Geoffrey Zweig, "Syntactic Clustering of the Web", In Proceedings of the Sixth International.

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

[3] Ranjna Gupta , Neelam Duhan , A. K. Sharma , Neha Aggarwal,"Query Based Duplicate Data Detection on WWW" (IJCSE) International Journal on Computer Science and Engineering Vol. 02, No. 04, 2010, 1395-1400.

[4] G. S. Manku, A. Jain, and A. D. Sarma. "Detecting nearduplicates for web crawling". In In ACM WWW'07, pages 141–150, NY, USA, 2007.

[5] Lan Yi, Bing Liu, Xiaoli Li. "Eliminating noisy information in web pages for data mining", In: Proceedings of the 9th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD), pp. 296 – 305.

[6] Fetterly, D., Manasse, M., Najork, M., 2004. "Spam, damn spam, and statistics: Using statistical analysis to locate spam web pages", in: Proceedings of the 7th International Workshop on the Web and Databases (WebDB), pp. 1-6.

[7] Hung-Chi Chang and Jenq-Haur Wang. "Organizing news archives by near-duplicate copy detection in digital libraries". In Asian Digital Libraries. Looking Back 10 Years and Forging New Frontiers, volume 4822 of Lecture Notes in Computer Science, pages 410 – 419. Springer Berlin, Heidelberg, 2007.

[8] Gerard Salton,EdwardA.Fox,Harry WU, "Extended Boolean Information Retrieval", Communications of the ACM 1983 Volume 26 Number 12

[9] Moses S. Charikar, "Similarity Estimation Techniques from Rounding Algorithms", In Proceedings of theThiry-Fourth Annual ACM Symposium on Theory ofComputing, pp: 19-21, 2002.

[10] Abdur Chowdhury, OphirFrieder, David Grossman and Mary Catherine Mccabe, "Collection Statistics for Fast Duplicate Document Detection", ACM Transactions on Information Systems, Vol. 20, No. 2, April 2002.

[11] Martin Theobald, Jonathan Siddharth, Andreas Paepcke "SpotSigs: Robust and Efficient Near Duplicate Detection in Large Web Collections", ACM 2008

[12] MaoshengZhong, Yi Hu & Lei Liu & Ruzhan Lu "A Practical Approach for Relevance Measure of Inter-Sentence" , Conference on Fuzzy Systems and Knowledge Discovery –IEEE Computer society 2008

[13] MidhunMathew,Shine N Das,TR Lakshmi "A Novel Approach for Near-Duplicate Detection of Web Pages using TDW Matrix" IJCA 2011

[14] 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.

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