Journal of Theoretical and Applied Information Technology
10th September 2014. Vol. 67 No.1
© 2005 - 2014 JATIT & LLS. All rights reserved·
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

DUPLICATE WEB PAGES DETECTION WITH THE SUPPORT OF 2D TABLE APPROACH

[1]A.C. SANTHA SHEELA, [2]C. JAYA KUMAR

[1]Research scholar, Department of Computer Applications, Faculty of Computing, Sathyabama University, Chennai, TamilNadu.

[2]Prof, Department of Computer Science &Engineering, R.M.K. Engineering College, Chennai, TamilNadu.

E-mail: [1]Shantha.jasmine@gmail.com, [2]cjayakumar2007@gmail.com

ABSTRACT

Duplicate and near duplicate web pages are stopping the process of search engine. As a consequence of duplicate and near duplicates, the common issue for the search engines is raising the indexed storage pages. This high storage memory will slow down the process which automatically increases the serving cost. Finally, the duplication will be raised while gathering the required data from the various sources based on the user's query. The duplication will definitely slow down the information retrieval process. Duplication is nothing but the similar content or documents located under various sites. Content duplication can be taken place at different forms and levels such as exact document copy, paragraph copy, sentence copy, single word changes and sentence structure changes. Duplication detection is the process of identifying the multiple representations of a same real world object. In this paper, the content duplication is identified using two dimensional (2D) text matrix approach. By using the proposed 2D matrix approach, the system was able to detect duplicate web pages with a high precision value 92% is highlighting that the duplicate web page detection with the 2D technique is performing well

Keyword: *Near Duplicate Detection, 2D Approach, Information Retrieval, Content Duplicate*.

1. INTRODUCTION

Web mining is an application of the data mining, i.e., when the data mining techniques are applied to the web content, it is referred to be the web mining. Web mining can be classified as web content mining, web usage mining and web structure mining. Mining the content of the web page is called the web content mining. Web usage mining is the process of extracting the information about the web page usage based on the user's requirements as such text, multimedia, etc. Mining the structure of the data is referred web structure mining [12]. Web page consists of the structured, unstructured and semi structured data. Web data defined in a tabular format are referred as the structured data. Data represented in the hierarchical format is called the unstructured data. Combination of structured and unstructured is called the semi structured data. It is nothing but the web page with a collection of the text, images, audio, video data. The chief gateways for access of information in the web are Search engines [2]. Search engines are applied for searching the web pages based on the portrayal from the web database. When the user enters their query on the search engine, the web pages put across with the contents relevant to the query will be returned. Web crawler crawls the web page depends upon the user query, i.e. fetches the web pages from single or multiple web database. Crawler populates an indexed repository of web pages. Indexing means sorted on the terms or keywords. The keywords are identified or extracted from the document or web page to allow quick searching for a particular query. The indexed information is utilized by the search engine in order to respond the queries. When the crawler receives more pages, it needs to be indexed more pages. Ranking fully based on the prominence and weight of the keywords in the document. Prominence of a document is nothing but the file name with keywords, title tag with keywords, Meta tag with keywords, first paragraph with keywords, last paragraph with keywords and body content with keywords. Weight can be calculated as the ratio of keywords frequency (number) to the total number of words on the page. Ranking is calculated to identify the importance of each word in the document or web page. Web search engine faces massive problems due to the duplicate and near duplicate web pages. The search engines are very much affected by content duplication due to the high indexed storage pages, which leads to slow processing and which directs the high serving cost. So duplication detection is very much essential. Web content duplication is done to speed up the access to a remote user community. The duplications can be occurred in many ways. Some of them are listed below.

- Multiple URL's for a same document
- Same web site hosted on multiple host names
- Web Spammers

Duplication detection can be done by either supervised learning or unsupervised learning. Supervised learning is the machine learning approach, here, the duplication detection system is trained using a set of data (training data), the system is tested with another set of data (test data) and the system is evaluated with a new set of data (validation data). The given data set is split into training, testing and validation data. Classification method or algorithm is used for supervised learning. Unsupervised learning means without any training data, data have been classified Clustering method or clustering algorithm is used for unsupervised learning.

The rest of the paper is organized as follows. In section 2, the related work done on the approaches available for duplicates and near duplicate detection is described. Section 3, 2D approach for duplicate web page detection is presented. In section 4, the experimental results are discussed. The conclusions are summed up in section 5.

2. RELATED WORK

Broder et al.[5] proposed a Digital Syntactic Clustering (DSC) algorithm for splitting the document into several shingles. A set of immediate terms in a document is called shingle. Shingles are chosen from each document to compare the similarity using jaccard overlap measure between these documents. Overlapping shingles are considered for duplication detection. Even the work done in billions of documents but its efficiency is low. The drawback of this method is more number of comparisons required. The performance of the algorithm works poorly on small documents and produces many potentially duplicated. Later, Border [6] proposed an improved algorithm named DSC-SS Digital syntactic clustering- Super Shingle. In this algorithm, the shingles are merged as super shingles and hash values have been generated for these super shingles. However, the algorithm is performing poorly on small documents.

Charikar's [7] proposed a random projection algorithm for identifying near-duplicates in web documents by mapping high dimensional vector of small sized fingerprint. It is a dimensionality reduction technique. For applying dimensionality reduction to web pages, features are extracted from the page and weights are assigned to the feature. These features are computed by tokenization, case folding, stop word removal, stemming and phrase detection. The author converted a high-dimensional vector into an f-bit fingerprint. Property of fingerprints for near-duplicates are differing in a small number of bit positions. This technique used to find more near duplicate pairs on different sites. Henzinger [8] improved the precision by comparing the shingles and simhash.

G S Manku et al. [9] proposed a simhash algorithm by adding the feature weight to random projection. Simhash is generated by identified feature vectors and corresponding feature weights. This technique is useful for both single fingerprints (online queries) and multiple fingerprints (batch queries). Hamming distance is applied for identifying the difference in the fingerprints. The fingerprints are considered as duplicate when the hamming distance of two simhash fingerprints is smaller than the threshold value.

Syed Mudhasir Y [3] described a method for detecting and eliminating the near duplicates and also re-ranking the documents by calculating their respective trustworthiness values. Their approach uses the web provenance concept using semantics by means of the provenance factors (Who, When, Where, What, Why and How). In this paper the author have tried with the first four factors for duplication detection and not considering the factors ‘Why’ and ‘How’. Thus, the architecture of a search engine or a web crawler based on the provenance and ranking are more effective in web search.

Theobald [10] proposed, SpotSigs: robust and efficient near duplicate detection in large web collections and Kołcz [11] proposed Lexicon randomization for near-duplicate detection with I-Match concept.

3.2 D Approach For Duplicate Web Page Detection

Detecting the duplicates and near duplicates are more essential with the aim of supporting the search engines to retrieve useful and distinct results on the first page. The size of the World Wide Web is estimated every day. The indexed web contains at least 5 billion of pages. The crawler crawl billion of web pages every day. This marking a page as a near duplicate should be done at quicker speeds. Near duplication detection is performed using the keywords extracted from the web documents. The crawled web documents are parsed to extract the distinct keywords. Parsing includes elimination of tags, stop words/common words to find the keywords. They pulled out keywords are stored in the table for processing the duplication detection. The similarity score of the two documents has been calculated using the keywords from the pages. One document is a new document another is already stored in the repository. The threshold value is evaluated at an average of similarity measures of more than hundred documents. The similarity score is lesser than the threshold value, then, the document is declared as near duplicate.

The duplication process is comprised with five steps:

A. Web Page Pre-processing

The first step in the identification of web content duplication is removing the tags and extracting the text contents alone for duplicate evaluation.

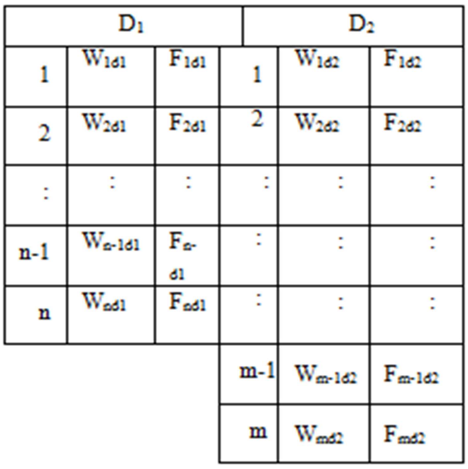
B. Stop Word Elimination

More than hundred stops words are collected and maintained as a table in the database. After removing the tags, the text contents from the web and the stop word table are passed as input for the stop word eliminator. It will remove the stop word from the text and produces the collected keywords from the newly considered web page as output.

C. Word Frequency Estimator

The collected keywords are used to find the total occurrences of the keywords in the actual document. After calculating the frequencies of each word, the keywords along with their frequency count are stored in the database, as in Table 1.

TABLE 1: Word-Frequency Pair of Both Documents D_1 & D_2



D_1			D_2		
1	W_{1d1}	F_{1d1}	1	W_{1d2}	F_{1d2}
2	W_{2d1}	F_{2d1}	2	W_{2d2}	F_{2d2}
:	:	:	:	:	:
n-1	W_{n-1d1}	F_{n-d1}	:	:	:
n	W_{nd1}	F_{nd1}	:	:	:
			m-1	W_{m-1d2}	F_{m-1d2}
			m	W_{md2}	F_{md2}

Here D1 and D2 represent the two documents and W_{id1}, (i=1,2..n) represents the words in D1 and F_{id1}, (i=1,2..n) represents the frequency count of ith word in D1. Similarly, in D2, the words and their frequencies are represented as W_{jd1},(j=1,2..m) F_{jd1}, (j=1,2..m). Here, n and m are used to represent the total number of keywords in D1 and D2. The values of n and m may or may not be equal, based on the number of keywords in D1 and D2.

D. 2D Word Table Construction

The unique keywords are considered for eliminating the duplication and sorted by alphabetical order. The keywords are testing to put up on a 2D table with 26 rows with nil or n number of columns. The 26 rows represent the 26 English alphabets. The word starts with ‘a’ will be placed in the 1st row. Similarly, the word starts with ‘z’ will be placed at 26th row. The number of columns of the first row is equal to the number of words starting with the alphabet ‘a’. In each row, the words will be arranged in ascending order. The table is an unequal sized one and it is represented as 26×n, n≥0, represented as in Table 2.

TABLE 2: 2D Word Table Representation

web-text	web-text			→ Words starts with ‘a’
web-text	web-text	web-text		→ Words starts with ‘b’
web-text	web-text	web-text	web-text	→ Words starts with ‘c’
:	:	:		:
:	:	:	:	:
:	:	:	:	
web-text				→ Words starts with ‘y’
web-text	web-text	web-text	web-text	→ Words starts with ‘z’

E. Similarity Measure Evaluation

A newly considered web page should be evaluated for its content duplication. If the contents or texts are similar as the old or existing document D1, no need to consider the new web page. To decide whether the new web page is a duplication of an existing one or not, the similarity significance should be measured between the D1 and the new document D2. Instead of comparing the documents in the form of strings as words or text, the supporting information such as the frequencies of the keywords and word position in the 2D table are taken in finding the similarity measures.

While considering the documents, the keyword set W_s is used to evaluate the similarity measures. W_s word set is nothing but the collection of three possible word sets such as common word set $\{W_c\}$ (appearing in both documents), words in D1 document alone but not in D2 document, $\{W_{d1}\}$ and words in D2 documents which are not in the D1 document, $\{W_{d2}\}$, represented in Equ (1).

$$W_s = \{\{W_c\}+\{W_{d1}\}+\{W_{d2}\}\} \qquad (1)$$

where,

W_s - Keyword set used for similaṛity measures

W_c - Common keywords, keywords occurring in both documents

W_{d1} - keywords only in D1 but not in D2

W_{d2} - keywords only in D2 but not in D1

All three word sets are kept in a form of table for the remaining process. They are framed as table as in Table 3.

TABLE 3: Keyword set - W_s

Common Keywords occurring in both documents (D1 and D2) - W_c	Keywords available only in D1 document - W_{d1}	Keywords available only in D2 document - W_{d2}

The three sets of words are taken from the table for evaluating the similarity measure between the documents. The total size of word sets is required to repeat the process for all individual keywords in the sets. The word set size is represented as in Equ (2).

The size of W_s = The size of W_c +The size of W_{d1} + The size of W_{d2}

$$N_s = N_c + N_{d1} + N_{d2} \quad (2)$$

Where,

N_s - Total number of keywords to be processed

N_c - Total number of common keywords in D1 and D2

N_{d1} - Total number of keywords only in D1, not in D2

N_{d2} - Total number of keywords only in D2, not in D1

The similarity measure is estimated as in Equ (3).

$$\mathrm{SM}_{d1d2} = \frac{S_1 + S_2 + S_3}{N} \quad (3)$$

Where,

SM_{d1d2} - Similarity measure between D1 and D2 documents

S1- similarity score for common word set, W_c

S2 -Similarity score for keyword set W_{d1}

S2 - Similarity score for keyword set W_{d2}.

The N value is nothing but the average of the number of keywords in D1 and D2, calculated as in Equ (4).

$$N = \frac{n \times m}{2} \quad (4)$$

Where,

N - Average word count

n - Total keyword count in D1

m - Total keyword count in D2

Equ (5) is used to calculate S1, by substituting the values from Equ (6) to (8).

$$S1 = \sum_{i=1}^{N_c} a_i \times |b_i - c_i| \quad ($$

$$a_i = \log(F_{id1} \times F_{id2}), i = 1,2, \dots, N_c \quad ($$

$$b_i = row_{T1_{w_{c_i}}} \times col_{T1_{w_{c_i}}}, i = 1,2, \dots, N_c \quad ($$

$$c_i = row_{T2_{w_{c_i}}} \times col_{T2_{w_{c_i}}}, i = 1,2, \dots, N_c \quad ($$

Where,

F_{id1}	-	Frequency count word of {W_c} in D
F_{id2}	-	Frequency count word of {W_c}in D2
$row_{T1_{w_{c_i}}}$	-	Row position o word of {W_c} fro table (T_1) of D1
$col_{T1_{w_{c_i}}}$	-	Column position word of {W_c} fro table (T_1) of D1
$row_{T2_{w_{c_i}}}$	-	Row position o word of {W_c} fro table (T_2) of D2
$col_{T2_{w_{c_i}}}$	-	Column position word of {W_c} fro table (T_2) of D2

S2 is calculated using Equ (9), by substituting the values from Equ (10) and Equ (11). And S3 is also calculated in the same way, by considering the words only in D2 alone, using Equ (12) to (14).

$$S2 = \sum_{j=1}^{N_{d1}} e_j \times g_j \quad (9)$$

$$e_j = \log(F_{jd1}), j = 1,2, \ldots, N_{d1} \quad (10)$$

$$g_j = row_{T1_{w_{d1_j}}} \times col_{T1_{w_{d1_j}}}, j = 1,2, \ldots, N_{d1} \quad (11)$$

$$S3 = \sum_{k=1}^{N_{d2}} h_k \times p_k \quad (12)$$

$$h_k = \log(F_{kd2}), k = 1,2, \ldots, N_{d2} \quad (13)$$

$$p_k = row_{T2_{w_{d2_k}}} \times col_{T2_{w_{d2_k}}}, k = 1,2, \ldots, N_{d2} \quad (14)$$

Where,

F_{jd1} - Frequency count of j^{th} word of $\{W_{d1}\}$ word set

$row_{T1_{w_{d1_j}}}$ - Row position of j^{th} word of $\{W_{d1}\}$ from 2D table (T_1) of D1

$col_{T1_{w_{d1_j}}}$ - The column position of j^{th} word of $\{W_{d1}\}$ from 2D table (T_1) of D1

F_{kd2} - Frequency count of k^{th} word of $\{W_{d2}\}$ word set

$row_{T2_{w_{d2_k}}}$ - Row position of k^{th} word of $\{W_{d2}\}$ from 2D table (T_2) of D2

$col_{T2_{w_{d2_k}}}$ - Column position of k^{th} word of $\{W_{d2}\}$ from 2D table (T_2) of D2

The document duplication decision is made using the following conditions.

if SM = 0 [Cond 1]

The documents are mirrored duplicates / Exact duplicate

else if $0 < SM \leq 15$ [Cond 2]

The documents are close similar

else if $16 < SM \leq 50$ [Cond 3]

The documents are slightly similar

else if SM > 50 [Cond 4]

The documents are least similar

end.

4. RESULT AND DISCUSSION

The result of our experiment is presented in this section. The proposed approach is implemented in MATLAB with MS Excel. The Keywords are extracted from the web documents are stored in MS Excel. We experimented with several documents and one example is given below:

Step 1: Two documents are considered for the experiment that extracted from the URL mentioned in Table 4.

D1 Cricket Rules
D2 About Cricket

By removal of the tags we get the content for D1 are as follows:

Welcome to the greatest game of all - Cricket. This site will help explain to an absolute beginner some of the basic rules of cricket. Although there are many more rules in cricket than in many other sports, it is well worth your time learning them as it is a most rewarding sport. Whether you are looking to play in the backyard with a mate or join a club Cricket-Rules will help you learn the basics and begin to enjoy one of the most popular sports in the world. Cricket is a game played with a bat and ball on a large field, known as a ground, between two teams of 11 players each. The object of the game is to score runs when at bat and to put out, or dismiss, the opposing batsmen when in the field. The cricket rules displayed on this page here are for the traditional form of cricket which is called "Test Cricket".

Step 2: More than hundred stop words are identified and saved in the database like “a”, “above”, “across” and so on. Using that database stop words are removed and keywords are extracted for document D1 is shown. Similarly keywords for D2 also taken.

'welcome' 'greatest ' 'game ' 'cricket' 'site' 'help' 'explain' 'absolute' 'beginner' 'basic' 'rules ' 'cricket' 'rules' 'cricket' 'sports' 'worth' 'time' 'learning' 'rewarding' 'sport' 'looking' 'play' 'backyard' 'mate' 'join' 'club' 'cricket-rules' 'help' 'learn' 'basics' 'begin' 'enjoy' 'popular' 'sports' 'world' 'cricket' 'game' 'played' 'bat' 'ball' 'large' 'field' 'known' 'ground' 'teams' 'players' 'object' 'game' 'score' 'runs' 'bat' 'dismiss' 'opposing' 'batsmen' 'field' 'cricket' 'rules' 'displayed' 'page' 'traditional' 'form' 'cricket' 'called' 'test' 'cricket' 'formats' 'game' 'matches' 'twenty20' 'cricket' 'rules' 'differ' 'slightly'

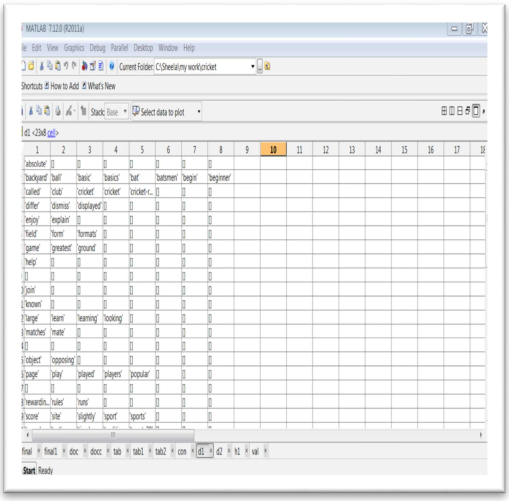
Step 3: Extracted keywords are stored and arranged in alphabetical with word frequency are as follows in Table 4.

Table 4: Word Frequency Estimator for document “Cricket Rules”.

Keyword	Frequency
'absolute'	1
'backyard'	1
'ball'	1
'basic'	1
'basics'	1
'bat'	2

In the above Table word “absolute” appears only once in the document and word “bat” appears twice in the document and so on.

Step 4: 2D Word Table Construction for “Cricket Rules” as shown in Fig 1. It consists word started with “a” in row1 and “b” in row2 and so on. Total 26 rows with n columns needed for 2D word table construction.

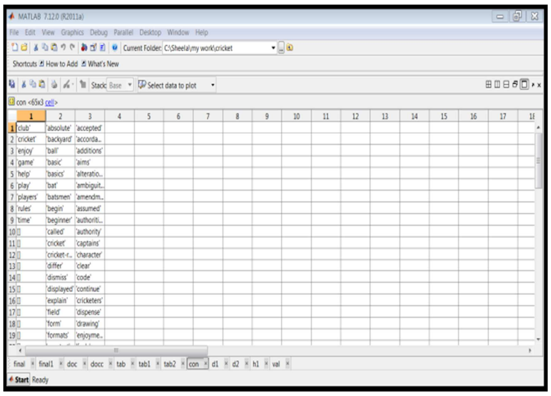

MATLAB 7.12.0 (R2011a)
le Edit View Graphics Debug Parallel Desktop Window Help
Current Folder: C:\Sheela\my work\cricket
Shortcuts How to Add What's New
Stack: Base
Select data to plot
final final1 doc docc tab tab1 tab2 con d1 d2 h1 val
Start Ready

d1 <23x8 cell>

1	2	3	4	5	6	7	8
'absolute'	[]	[]	[]	[]	[]	[]	[]
'backyard'	'ball'	'basic'	'basics'	'bat'	'batsmen'	'begin'	'beginner'
'called'	'club'	'cricket'	'cricket'	'cricket-r...	[]	[]	[]
'differ'	'dismiss'	'displayed'	[]	[]	[]	[]	[]
'enjoy'	'explain'	[]	[]	[]	[]	[]	[]
'field'	'form'	'formats'	[]	[]	[]	[]	[]
'game'	'greatest'	'ground'	[]	[]	[]	[]	[]
'help'	[]	[]	[]	[]	[]	[]	[]
[]	[]	[]	[]	[]	[]	[]	[]
'join'	[]	[]	[]	[]	[]	[]	[]
'known'	[]	[]	[]	[]	[]	[]	[]
'large'	'learn'	'learning'	'looking'	[]	[]	[]	[]
'matches'	'mate'	[]	[]	[]	[]	[]	[]
[]	[]	[]	[]	[]	[]	[]	[]
'object'	'opposing'	[]	[]	[]	[]	[]	[]
'page'	'play'	'played'	'players'	'popular'	[]	[]	[]
[]	[]	[]	[]	[]	[]	[]	[]
'rewardin...	'rules'	'runs'	[]	[]	[]	[]	[]
'score'	'site'	'slightly'	'sport'	'sports'	[]	[]	[]

Fig 1. 2D Word Table Construction for “Cricket Rules”

Step 5: Similarity Measure Evaluation shown in Fig 2 done by listing the common words from document D1 and D2 in column1 and words only in D1 not in D2 listed in column2 and words only in D2 not in D1 are listed in column3. In this sample, we get nine common words, 48 distinct words in D1 and 65 distinct words in D2. For calculating similarity between these two documents Equ (3) has been used. The similarity value is 11.4989. It satisfies the condition cond 2, concluded that document D1 & D2 “Cricket Rules” and “About Cricket” are close similar.


MATLAB 7.12.0 (R2011a)
File Edit View Graphics Debug Parallel Desktop Window Help
Current Folder: C:\Sheela\my work\cricket
Shortcuts How to Add What's New
Stack: Base
Select data to plot
final final1 doc docc tab tab1 tab2 con d1 d2 h1 val
Start Ready

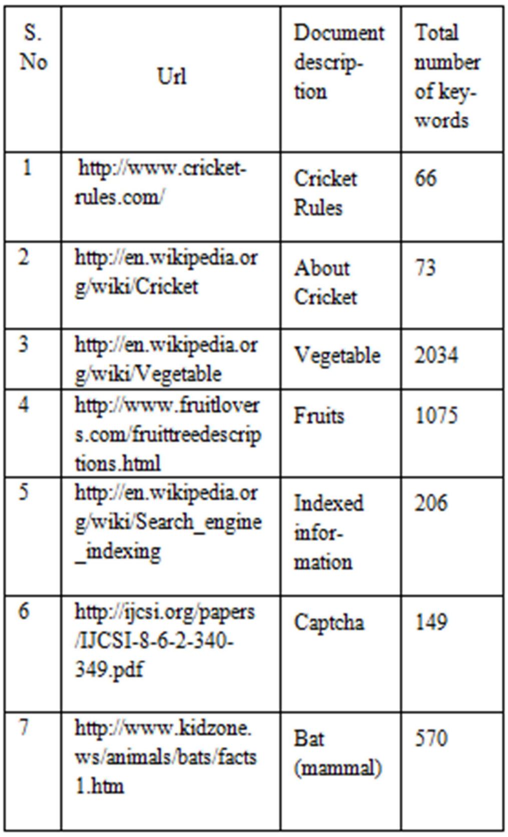
con <65x3 cell>

	1	2	3
1	'club'	'absolute'	'accepted'
2	'cricket'	'backyard'	'accorda...
3	'enjoy'	'ball'	'additions'
4	'game'	'basic'	'aims'
5	'help'	'basics'	'alteratio...
6	'play'	'bat'	'ambiguit...
7	'players'	'batsmen'	'amendm...
8	'rules'	'begin'	'assumed'
9	'time'	'beginner'	'authoriti...
10	[]	'called'	'authority'
11	[]	'cricket'	'captains'
12	[]	'cricket-r...	'character'
13	[]	'differ'	'clear'
14	[]	'dismiss'	'code'
15	[]	'displayed'	'continue'
16	[]	'explain'	'cricketers'
17	[]	'field'	'dispense'
18	[]	'form'	'drawing'
19	[]	'formats'	'enjoyme...

Fig 2. Similarity Measure Evaluation for document “Cricket Rules”

The system is tested with more than hundred web pages. To show the performance of the new approach, a few documents are listed in Table 5.

Table 5: Document Details



S. No	Url	Document description	Total number of keywords
1	http://www.cricket-rules.com/	Cricket Rules	66
2	http://en.wikipedia.org/wiki/Cricket	About Cricket	73
3	http://en.wikipedia.org/wiki/Vegetable	Vegetable	2034
4	http://www.fruitlovers.com/fruittreedescriptions.html	Fruits	1075
5	http://en.wikipedia.org/wiki/Search_engine_indexing	Indexed information	206
6	http://ijcsi.org/papers/IJCSI-8-6-2-340-349.pdf	Captcha	149
7	http://www.kidzone.ws/animals/bats/facts1.htm	Bat (mammal)	570

All the documents are compared with each other, with n^2 (100 × 100 = 10000) combinations. It is observed that, for the similar documents such as considering a document as D1 and D2, the similarity measure for the duplicate documents is generated as 0. This is the indication of mirror duplication. Sample similarity values are listed in Table 6.

From the Table 6 initially, two different documents about cricket are tested for the similarity. The similarity score for the two documents is 11.4989. This score helps to decide the status of the two documents are close similar.

Instead of taking two different documents, the same document Cricket is considered as Doc1 and Doc2. From this, the score is generated at 0.0. From the decision condition, this value helps to fix the two documents are exactly similar.

Next, the similarity measure is found between the documents Fruits and Vegetables, and it produced the 222.7227, which helps to decide the two documents are not similar.

When finding similarity between documents which consist less than the threshold value, then the documents are dissimilar documents. Otherwise the documents are similar documents.

5. CONCLUSION

Duplicate web pages detection techniques are important for improving the quality of search engine for extracting information. Numerous pages in web either duplicated or near duplicated. This kind of duplication leads to take more time to retrieve the results. When the data are integrated from the multiple web database duplicates or near duplicates arise. A 2D approach to duplicate web page identification is tried by 92% of precision. This is simple and fast to estimate the different web contents such as exact duplicates, close duplicates, slightly duplicates, or least duplicates.

REFERENCE

[1]. Narayana, V. A., Premchand, P., & Govardhan, A. “Effective Detection of Near Duplicate Web Documents in Web Crawling”, International Journal of Computational Intelligence Research 5.1, 83-96, 2009.

[2]. Syed Mudhasir, Y., Deepika, J.,Sendhilkumar, S., & Mahlakshmi, G.S. “ Near-Duplicates Detection and Elimination Based on Web Provenance for Effective Web search”, IJIDCS vol 1: No:1,2011.

[3]. Tanvi Gupta., & Latha Banda. “ A Hybrid Model For Detection And Elimination Of Near- Duplicates Based On Web Provenance For Effective Web Search.” International Journal Of Advances In Engineering & Technology (IJAET), 2012.

[4]. Broder, S., Glassman, M., Manasse, & Zweig, G. “Syntactic Clustering of the Web”. In 6th International World Wide Web Conference, 393-404, 1997.

[5]. Broder, A. “Identifying and filtering near-duplicate documents”. In Giancarlo, R.,

Sankoff, D. (eds.) CPM 2000. LNCS, vol. 1848, 1–10, 2000.

[6]. Charikar. M. S., "Similarity Estimation Techniques from Rounding Algorithms", 34th Annual ACM Symposium on Theory of Computing,2002.

[7]. Henzinger, M. "Finding near-duplicate web pages: a large-scale evaluation of algorithms", In Proceedings of the 29th annual international ACM SIGIR Conference on Research and Development in Information Retrieval, 284–291, 2006.

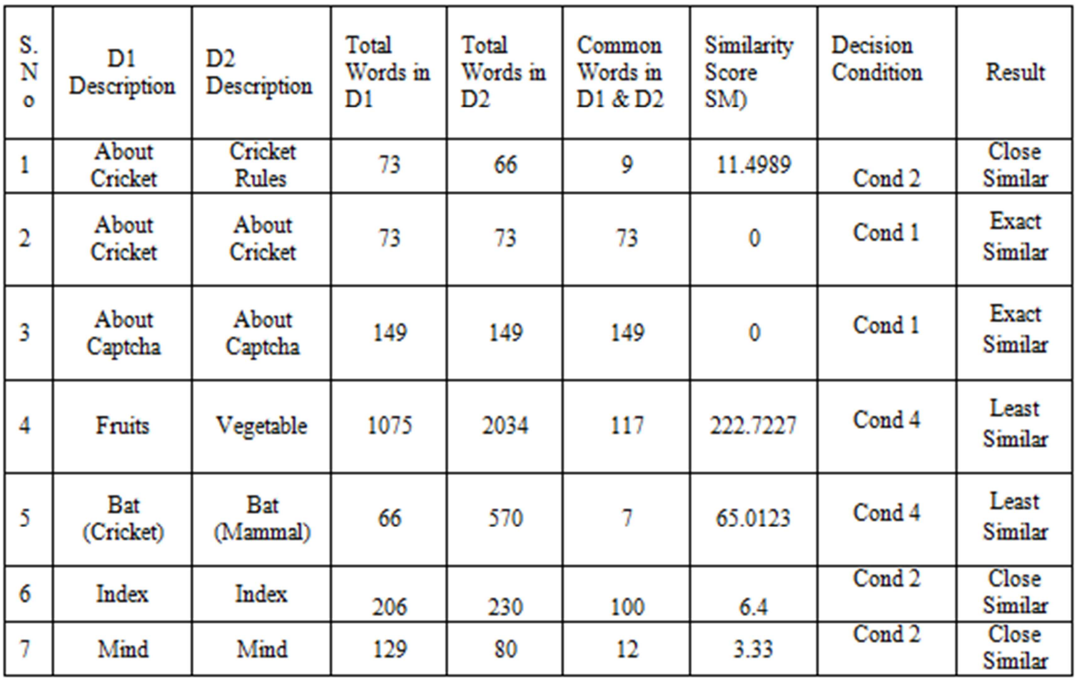
[8]. Manku, G.S., Jain, A., & Sarma, A.D., "Detecting near-duplicates for web crawling", In: Proceedings of the 16th International Conference on World Wide Web, 141–150, 2007.

[9]. Theobald, M., Siddharth, J., & Paepcke, "SpotSigs: robust and efficient near duplicate detection in large web collections". In: Proceedings of ACM SIGIR, 2008.

[10]. Kołcz, A., & Chowdhury, A. "Lexicon randomization for near-duplicate detection with I-Match. Supercomputer", 255-276, 2008.

[11]. Raymond Kosala & Hendrik Blockeel, "Web Mining Research: A Survey", ACM SIGKDD, 2000.

TABLE 6: Similarity Measures



S. N o	D1 Description	D2 Description	Total Words in D1	Total Words in D2	Common Words in D1 & D2	Similarity Score SM)	Decision Condition	Result
1	About Cricket	Cricket Rules	73	66	9	11.4989	Cond 2	Close Similar
2	About Cricket	About Cricket	73	73	73	0	Cond 1	Exact Similar
3	About Captcha	About Captcha	149	149	149	0	Cond 1	Exact Similar
4	Fruits	Vegetable	1075	2034	117	222.7227	Cond 4	Least Similar
5	Bat (Cricket)	Bat (Mammal)	66	570	7	65.0123	Cond 4	Least Similar
6	Index	Index	206	230	100	6.4	Cond 2	Close Similar
7	Mind	Mind	129	80	12	3.33	Cond 2	Close Similar