A HYBRID MODEL FOR DETECTION AND ELIMINATION OF NEAR- DUPLICATES BASED ON WEB PROVENANCE FOR EFFECTIVE WEB SEARCH

Tanvi Gupta[1] and Latha Banda[2]
[1]Department of Computer Science, Lingaya's University, Faridabad, India
[2]Associate Prof. in Department of Computer Science, Lingaya's University, Faridabad, India

ABSTRACT

Users of World Wide Web utilize search engines for information retrieval in web as search engines play a vital role in finding information on the web. But, the voluminous amount of web documents has weakened the performance and reliability of web search engines. As, the subsistence of near-duplicate data is an issue that accompanies the growing need to incorporate heterogeneous data. 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. Such near-duplicates can be detected and eliminated using the concept of Web Provenance and TDW matrix Algorithm. The proposed work is the model that combines content, context, semantic structure and trust based factors for classifying and eliminating the results as original or near-duplicates.

KEYWORDS: *Web search, Near-duplicates, Provenance, Semantics, Trustworthiness, Near-Duplicate Detection, Term-Document-Weight Matrix, Prefix filtering, Positional filtering, Singular Value Decomposition.*

I. 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. Hence it has become very important that the users get the best results for their queries. However, in any web search environment there exist challenges when it comes to providing the user with most relevant, useful and trustworthy results, as mentioned below:

- The lack of semantics in web
- The enormous amount of near-duplicate documents
- The lack of emphasis on the trustworthiness aspect of documents

There are also many other factors that affect the performance of a web search. One of the most important factor is the presence of duplicate and near-duplicate web documents which has created an additional overhead for the search engines. 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. These (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.

The following subsection briefly discuss the concepts of near-duplicate detection , TDW matrix Algorithm and Provenance.

A. Near-Duplicates Detection

The processes of identifying near duplicate documents can be done by scanning the document content for every document. That is, when two documents comprise identical document content, they are regarded as duplicates. And files that bear small dissimilarities and are not identified as being exact duplicates of each other but are identical to a remarkable extent are known as near-duplicates.
Following are some of the examples of near duplicate documents :-

- Documents with a few different words - widespread form of near-duplicates
- Documents with the same content but different formatting – for instance, the documents might contain the same text, but dissimilar fonts, bold type or italics
- Documents with the same content but with typographical errors
- Plagiarized documents and documents with different versions
- Documents with the same content but different file type – for instance, Microsoft Word and PDF.
- Documents providing same information written by the same author being published in more than one domain.

B. TDW Matrix Based Algorithm

Midhun.et.al[7] had described the TDW Matrix based Algorithm as a three-stage algorithm which receives an input record and a threshold value and returns an optimal set of near-duplicates.

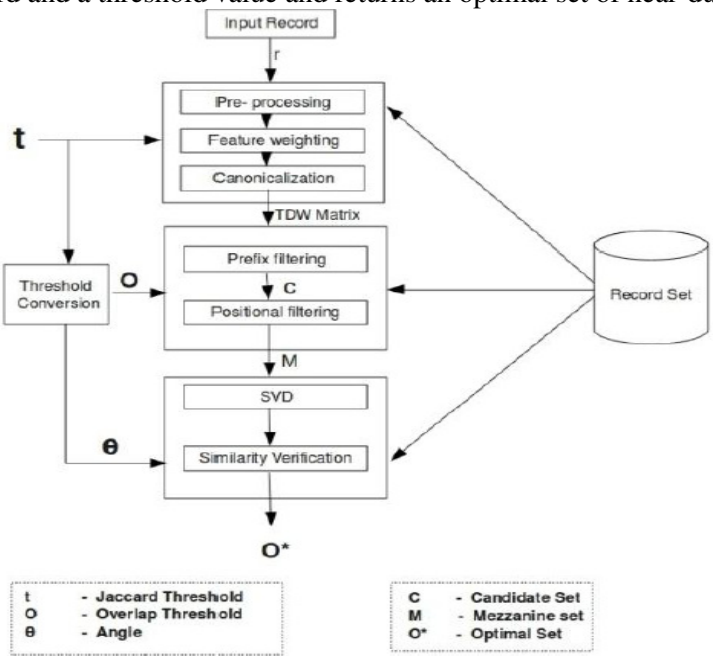

Figure1: General Architecture

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, 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.

C. Provenance

According to Y. Syed Mudhasir.et.al[6] ,one of the causes of increasing near-duplicates in web is that the ease with which one can access the data in web and the lack of semantics in near-duplicates detection techniques. It has also become extremely difficult to decide on the trustworthiness of such web documents when different versions/formats of the same content exist. Hence, the needs to bring in semantics say meaningful comparison in near-duplicates detection with the help of the 6W factors –

Who *(has authored a document)*, What *(is the content of the document)*, When *(it has been made available)*, Where *(it is been available)*, Why *(the purpose of the document)*, How *(In what format it has been published/how it has been maintained)*. This information can also be useful in calculating the trustworthiness of each document. A quantitative measure of how reliable that any arbitrary data is could be determined from the provenance information. This information can be useful in representative elimination during near-duplicate detection process and to calculate the trustworthiness of each document.

ORGANIZATION

SECTION 2: Related Work.
SECTION 3: Problem Formulation along with details of Proposed Work.
SECTION 4: Experimental set up to implement the steps.
SECTION 5: Analysis of result in terms of precision and recall.
SECTION 6: A conclusion detailing and Future advancement.

II. RELATED WORK

Works on near-duplicates detection and elimination are many in the history. In general these works may be broadly classified as:

1) Syntactical Approach
   (a) Shingling (b)Signature (c)Pair wise Similarity (d)Sentence Wise Similarity
2) URL Based Approach
   (a) DUST BUSTER Algorithm
3) Semantics Approach
   (a) Fuzziness Based (b) Semantic Graphs

A. Syntactical Approach

One of the earliest was by Broder et al[1] , proposed a technique for estimating the degree of similarity among pairs of documents, known as shingling, does not rely on any linguistic knowledge other than the ability to tokenize documents into a list of words, i.e., it is merely syntactic. In this, all word sequences (shingles) of adjacent words are extracted. If two documents contain the same set of shingles they are considered equivalent and can be termed as near-duplicates. The problem of finding text-based documents similarity was investigated and a new similarity measure was proposed to compute the pair-wise similarity of the documents using a given series of terms of the words in the documents.

The Signature method[2], suggested a method of descriptive words for definition of near-duplicates of documents which was on the basis of the choice of N words from the index to determine a signature of a document. Any search engine based on the inverted index can apply this method. Any two documents with similar signatures are termed as near-duplicates.

Problems in Syntactic Approach:

- The stated syntactic approaches carry out only a text based comparison.
- These approaches did not involve the URLs in identification of near-duplicates.

B. URL Based Approach

A novel algorithm, Dust Buster[3], for uncovering DUST (Different URLs with Similar Text) was intended to discover rules that transform a given URL to others that are likely to have similar content. Two DUST rules are:-

1) Substring substitution rule
2)Parameter substitution rule

C. Semantics Approach

A method on plagiarism detection using fuzzy semantic-based string similarity approach was proposed by Salha.et.al[4]. The algorithm was developed through four main stage:-

1) Pre-processing which includes tokenization, stemming and stop words removing.
2) Retrieving a list of candidate documents for each suspicious document using shingling and

Jaccard coefficient.

3) Suspicious documents are then compared sentence-wise with the associated candidate documents. This stage entails the computation of fuzzy degree of similarity that ranges between two edges: 0 for completely different sentences and 1 for exactly identical sentences. Two sentences are marked as similar (i.e. plagiarized) if they gain a fuzzy similarity score above a certain threshold.
4) The last step is post-processing hereby consecutive sentences are joined to form single paragraphs/sections.

III. PROPOSED WORK

Problem Formulation: The paper proposed the novel task for detecting and eliminating near-duplicate web pages to increase the efficiency of web crawling. So, the technique proposed aims at helping document classification in web content mining by eliminating the near-duplicate documents and then re-ranking the documents using trustworthiness values. For this , a hybrid model of Web Provenance Technique and TDW-based Algorithm. To evaluate , the accuracy and efficiency of the model two benchmark measures are used: Precision and recall.

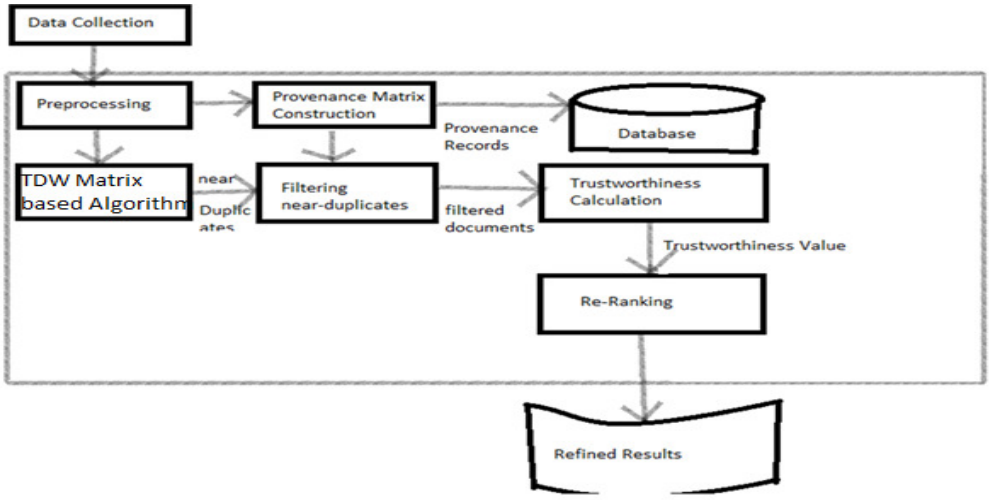

Figure2: A Hybrid model of Web Provenance and TDW based Matrix Algorithm

A. Architectural Steps

Figure2 shows the architectural steps which includes:(i)Data Collection (ii) Pre-processing (iii) Construction of Provenance Matrix (iv) Construction of Who Matrix, Where Matrix, When Matrix (v) Store in database (vi) Rendering Phase in TDW-Matrix Based Algorithm (vii) Filtering Phase in TDW-Matrix Based Algorithm (viii) Verification Phase in TDW-Matrix Based Algorithm (ix) Filtering Near Duplicates (x) Trustworthiness Calculation (xi) Re-Ranking using trustworthiness values(xii) Refined Results

1. Data Collection

Data is in the form of html pages in a specified format. For this project , 100 html pages is being used to check the accuracy and efficiency.

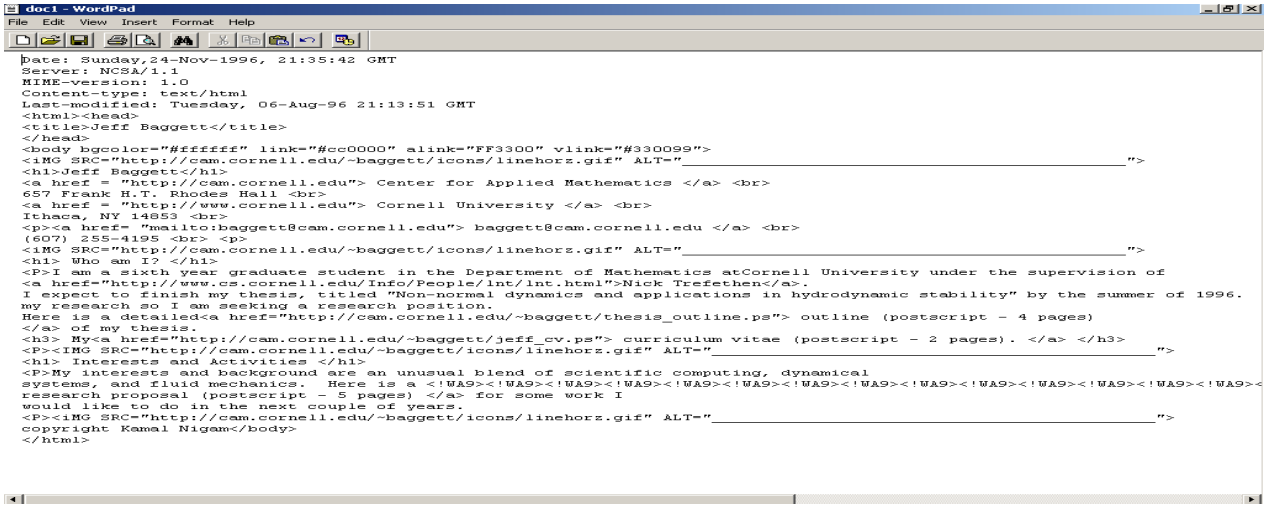

Figure3: Format of html page

2. Pre-Processing

The data collected in the form of html pages are then pre-processed using following techniques: (i) Tokenization (ii) Lemmatization (iii) Stop Word Removal

3. Construction of Provenance Matrix

Provenance matrix consists of 6 W factors:-Who(copyrighted by which company or person),When(it has been available (server name)),What(is the content of the html page in body),Why(the purpose of the document),How(In what format it has been published/how it has been maintained. Table 1 shows the provenance matrix described in [8].

Table1: Provenance Matrix

Factors	Doc1	Doc2	Doc3
Who	Company or Person Name of doc1 who has copyright of it.	Company or Person Name of doc2 who has copyright of it.	Company or Person Name of doc3 who has copyright of it.
When	Date or year of launch	Date or year of launch	Date or year of launch
Where	Server name	Server name	Server name
What	Content of Doc1	Content of Doc2	Content of Doc3
Why	Title of the page or first heading in the body part of doc1.	Title of the page or first heading in the body part of doc2.	Title of the page or first heading in the body part of doc3.
How	Format of doc1.	Format of doc2.	Format of doc 3.

4. Construction of Who Matrix, Where Matrix, When Matrix

Who matrix, Where Matrix, and When Matrix are binary matrices which represents the value '1' or '0' if the token is present or absent respectively.

5. Store in database

These Provenance Matrix, Who Matrix, Where Matrix and When Matrix of each document will be store in database.

6. Rendering Phase in TDW-Matrix

Rendering Phase Algorithm described by Midhun.et.al[7] is as follows:

Input: *Web_Document, Record_Set*
Output: *TDW_Matrix*

Remarks: $Wx \rightarrow$ total weight of the term x
Rendering (*Web_Document, Record_Set*)
Input_Record←Pre_Processed(*Web_Document*);
F←Full feature set(*Input_Record*);
for all $xi \in F$
Wx←Weight_Scheme(*xi*);
Wr←Σ*Wx*;
for all i, $1 \leq i \leq |F|$
Wx←Normalize(*Wx, Wr*);
T←Thresholding(*Wr*);
r← φ;
for all $xi \in F$
if $(Wx \geq T)$
$r \leftarrow r \cup xi$;
TDW_Matrix← Canonicalize(*r, Record_Set*);
return *TDW_Matrix*;
Rendering Phase in TDW-Matrix consists of following phases:

(i) Feature Weighting
Feature Weighting is done according to the following weighting scheme given in table 2 described in [7].

Table2: 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

Weight of each token =No. of occurrences of the token * weight of respective term field in weighting scheme-(1)

(ii) Normalization
W_x=Weight of every term/ average -(2)
Where, average= (sum of weights of terms in a document)/ no. of documents - (3)
Where, W_x is total weight of the token or term

(iii) Thresholding
Threshold value= Sum of terms weight in a document/ Sum of total weights of all documents.-(4)
Select those normalized weight values whose value is greater than threshold value, rest are rejected.

(iv) Canonicalization
1) Documents are canonicalized according to the document frequency ordering.
2) The terms for each documents are then arranged in increasing order according to the document frequency.

(v) TDW Matrix
TDW Matrix will consists of Weights of the token in each document. Following will show an example:

Let *r1, r2, r3* be three canonicalized records. *r1={x2, x1, x3} , r2={x4, x1, x3}, r3={x2, x4, x1, x3}*

terms \ records	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}

Figure4: TDW Matrix

7. Filtering Phase

Filtering Phase Algorithm described by Midhun.et.al[7] is as follows:

Input: *TDW_Matrix,Record_Set,t*
Output: *M(Mezzanine set)*
Remarks: Assume that *Input_Record* is represented as the first entry in *TDW_Matrix*
Filtering (*TDW_Matrix, Record_Set, t*)
r←*TDW_Matrix*[1];
//prefix filtering
C← φ;
Prefix_Length← |*r*|- ⌈*t*.|*r*| ⌉+1;
for all $r_i \in Record\_Set$
$Prefix_i$←|r_i|- ⌈*t*.|r_i| ⌉+1;
for all *j,k*; 1≤ *j* ≤ *Prefix_Length*, 1≤ *k* ≤ $Prefix_i$
if (*r*[*j*] == *ri*[*k*])
C← *C* ∪ *ri*;
//positional filtering
M← φ;
for all $r_i \in C$
O←t/t+1(|*r* |+|r_i|);
for all *p,q*; 1≤ *p* ≤ *Prefix_Length*, 1≤ *q* ≤ $Prefix_i$
if (*r*[*p*]==r_i[*q*])
ubound←1+ min(|*r*|-*p*, |r_i|-*q*);
if (*ubound* ≥ *O*)
M← *M* ∪ r_i;
return *M*;

Filtering Phase consists of :

1) Prefix Filtering
2) Positional Filtering

Prefix filtering and positional filtering, are performed to reduce the number of candidate records. In Prefix Filtering , the value of t (Jaccard similarity threshold)=0.5 is considered and in positional filtering , O is called Overlap Constraint.

(i)Prefix Filtering

Principle: Given an Ordering O of the token of the Universe U and a set of records, each with tokens sorted in the order O. Let the p-prefix of a record x be the first p tokens of x. If O(x,y)>=a, then the (|x|-a+1)-prefix of x and the (|y|-a+1)-prefix of y must share at least one token.

(ii) Positional Filtering

Principle:- Given an ordering O of the token universe U and a set of records, each with tokens sorted in the order of O. Let token w = x[i], w partitions the record into the left partition $x_l(w) = x[1 . . (i - 1)]$ and the right partition $x_r(w) = x[i . . |x|]$. If O(x, y) >=a, then for every token $w \in x \cap y$, $O(x_l(w), y_l(w)) + \min(|x_r(w)|, |y_r(w)|) >= a$.

Both principles are described by Chuan Xia et.al[5].

(iii) Mezzanine set

1) Final result after filtering we get is mezzanine set from where the optimal set is extracted.

2) Mezzanine set *M,* is a form of a weight matrix *A* such that columns represent documents and rows represent terms.

3)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.*

8.Verification Phase in TDW-Matrix Based Algorithm

(i) Singular Value Decomposition(SVD)

The singular value decomposition of an $m \times n$ real or complex matrix M is a factorization of the form

$$M=U\sum V^* \quad -(5)$$

where U is an $m \times m$ real or complex unitary matrix, Σ is an $m \times n$ rectangular diagonal matrix with nonnegative real numbers on the diagonal, and V^* is an $n \times n$ real or complex unitary matrix. The diagonal entries $\Sigma_{i,i}$ of Σ are known as the singular values of M. The m columns of U and the n columns of V are called the left singular vectors and right singular vectors of M, respectively.

(ii) Similarity Verification

Similarity verification is done on a huge record set having n number of records(documents) {r1,r2,….,rn} *and* an optimal set of near-duplicate records are returned. For similarity verification , Jaccard Coefficient similarity is used:

$$J(X,Y)=|X \cap Y|/|X \cup Y|, \quad -(6)$$

where X and Y are the document containing different tokens. The value of J(X,Y) is between 0 and 1 and the value lies above 0.5 is considered to be dissimilar whereas less than 0.5 is considered to be similar.

Formally, Two documents are purely dissimilar when the value of J(X,Y) is 1 and exactly similar when value is 0.

9. Filtering Near Duplicates

Algorithm for filtering Near-Duplicates referenced from [6]

1) Who-> Compare author_info(Di,Di+1) if equal return 1 , else return 0;
   If rule 1 return 0, then
2) When->Document with Earliest (Date of Publish(D_i), Date of Publish(D_i+1))
   If rule 1 returns 1, then
3) Where->Compare published_place(D_i,D_i+1) returns D_i/D_i+1 with standardized publication
4) Why->Check purpose(D_i,D_i+1), Returns D_i/D_i+1 with a better purpose
5) How->Check format(D_i,D_i+1),Returns D_i/D_i+1 with a better format

10. Trustworthiness Calculation

The trustworthiness value for each document can be calculated with the help of factors[6]:-

1) Accountability:- deals with the author information.
2) Maintainability :-deals with the availability of up-to-date content
3) Coverage:- deals with the number of working links with respect to the total number of links.
4) Authority:- deals with the place where the document has been published.

11. Re-Ranking using trustworthiness values

Re-Ranking of the documents are done using the concept of maintainability that deals with the update-date-content.

12. Refined Results

Refined Results are in the form of near-duplicates and non-near-duplicates.

IV. EXPERIMENTAL SET UP

To conduct the required experiments, we use the dataset described in the proposed work. To implement the above mentioned steps described in Section III C # .Net is used .The database used is SQL Server 2000. Also, to implement the last stage of TDW matrix, Matlab can be used to process the matrix which is decomposed into 2D coordinates using SVD techniques.

V. RESULT AND DISCUSSION

For evaluating the degree of accuracy, efficiency and scalability of the proposed work , two standard benchmark are used:

1) Precision 2) Recall

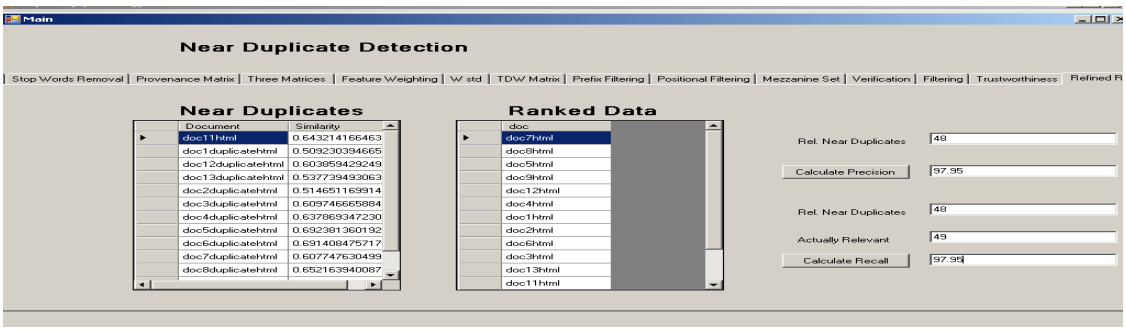

Figure:5 Outcome for 100 documents

$$precision = \frac{|\{relevant\ documents\} \cap \{retrieved\ documents\}|}{|\{retrieved\ documents\}|} \quad -(7)$$

$$recall = \frac{|\{relevant\ documents\} \cap \{retrieved\ documents\}|}{|\{relevant\ documents\}|} \quad -(8)$$

A. Outcome and Performance Measures

Figure5 shows the refined results in the form of near-duplicates and ranked data, and also the outcome for 100 documents , in which 49 duplicates were present but the above implementation detect 48 relevant duplicates which provides precision and recall to be 97.95%.

Table 3 shows the performance measures having number of documents , actual duplicates in a dataset, number of documents detected by software ,number of relevant documents, its precision and recall in percentages.

B. Graphs

The two graphs in figure 6 and figure 7 shows the Performances which is increasing with the increase in number of documents.

Table: 3 Performance Measures

No. of documents	Actual duplicates in dataset	No. of documents detected by software	No. of Relevant Documents out of the detected one	Precision(in Percentage%)	Recall(in percentage %)
20	9	9	8	88	88
25	12	12	11	91.66	91.6
30	14	14	13	92.85	92.8
35	17	17	16	94.11	94.1
40	19	19	18	94.73	94.7
45	22	22	21	95.45	95.40
50	24	24	23	95.83	95.83
55	27	27	26	96.29	96.28
60	29	29	28	96.55	96.54
65	32	32	31	96.87	96.87
70	34	34	33	97.05	97
75	37	37	36	97.29	97.2
80	39	39	38	97.43	97.43
85	42	42	41	97.61	97.6
90	44	44	43	97.72	97.72
95	47	47	46	97.87	97.8
100	49	49	48	97.95	97.9
			AVERAGE=	95.603529	95.57529

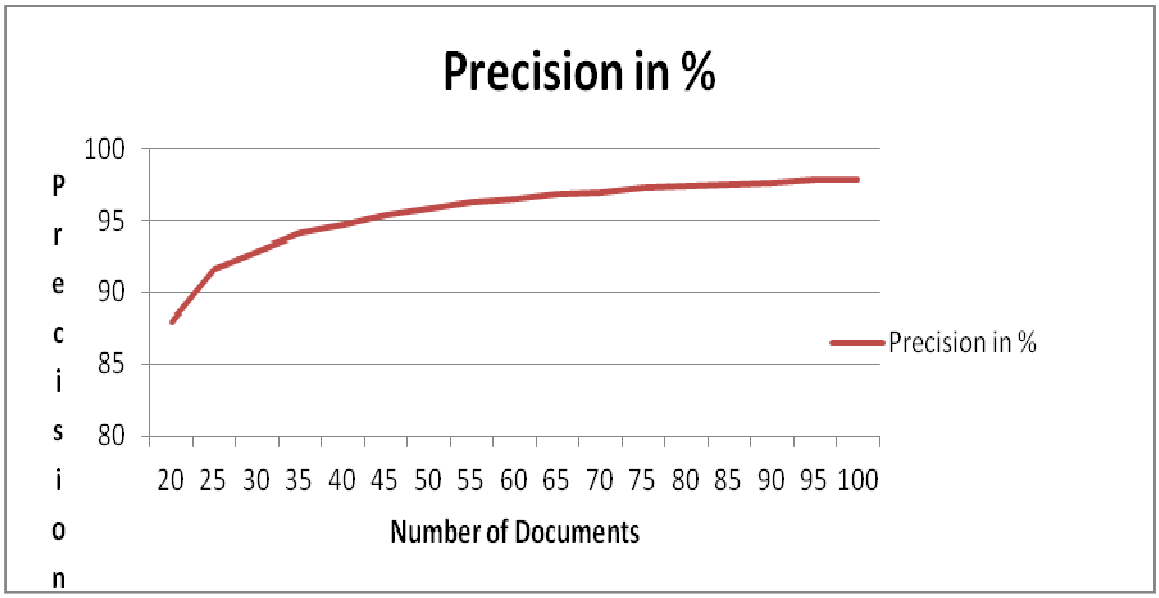

Figure6 : Graph of Precision

The graph of precision in figure 6 shows the exactness or quality of the concept. More the precision means getting more relevant results.

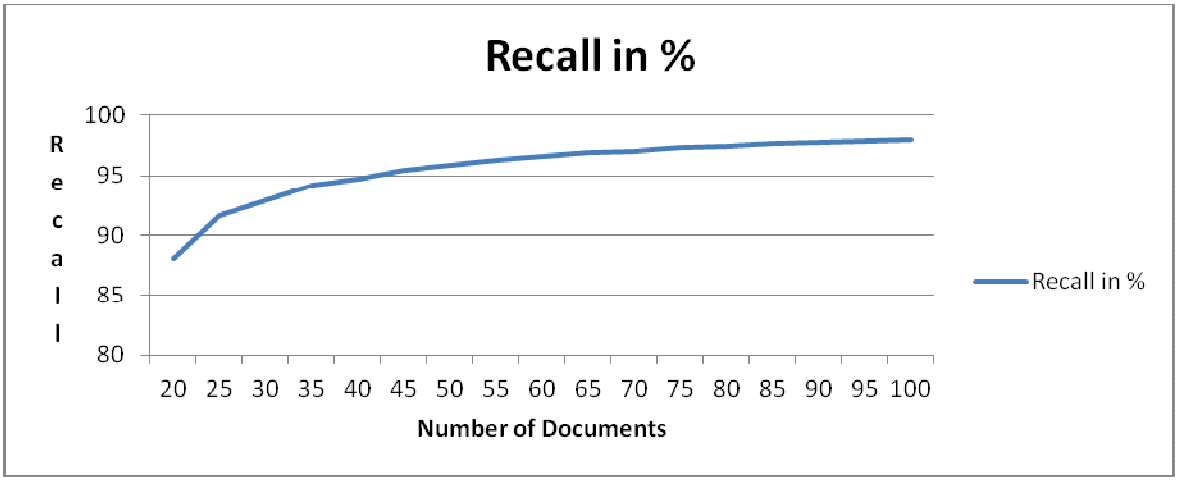

Figure7 : Graph of Recall

The graph of recall in figure 7 shows the completeness or quantity of the concept. More Recall means most of the relevant results will come.

C. Comparison of Experiments

a) When TDW Matrix based algorithm is used to detect the duplicates or near-duplicates the figure 8 shows the performance measures[7] i.e. Precision and Recall be 94.9% and 93.3% respectively. .

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**

Figure8 : Performance Measures of TDW Matrix Based Algorithm

b) When Web Provenance Technique is used to detect and eliminate the near-duplicates two concepts were used : a) DTM b) Provenance Matrix which uses cluster based approach.
The clusters of documents that are highly similar in both observations(i.e. DTM and Provenance Matrix) are classified as near-duplicates. From Fig. 9[6] and 10[6], the cluster of document which is highly similar in both observation 1 and 2 are Doc 2, Doc 5, Doc 6, Doc 7, Doc 8 and Doc 9 and Doc 10 since they are found to be highly similar on both the content and the Provenance factors.

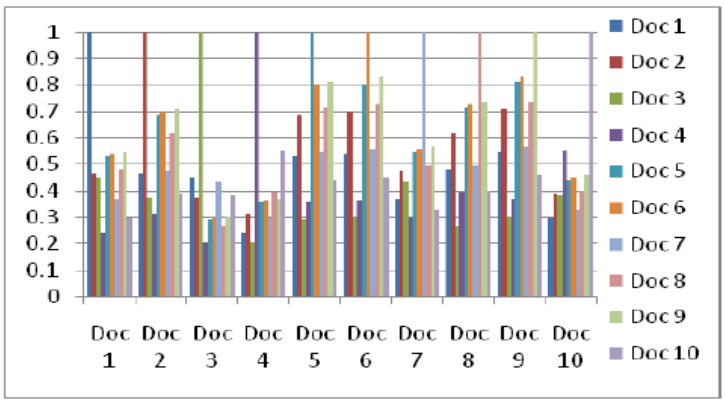

Figure9 : Comparison Based on DTM

c)When a hybrid model of Web Provenance and TDW Matrix Based Algorithm is considered , it will provide much more efficiency as compared to both individually. This is shown in Table 3(Performance Measures).

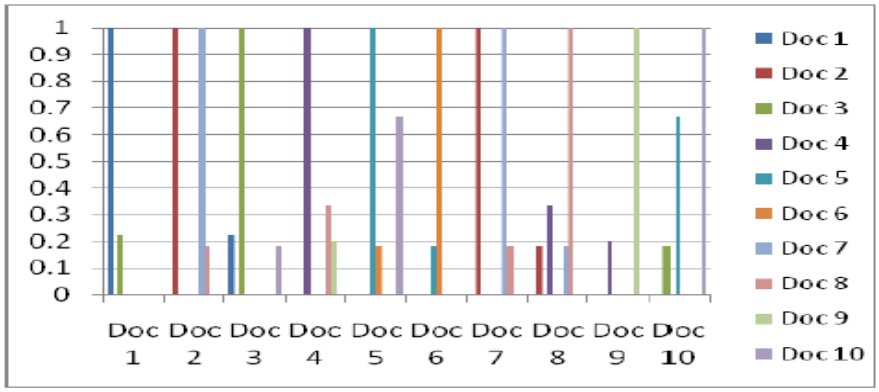

Figure 10 : Comparison Based on Provenance Matrix

In the Hybrid Model, first the pre-processing (tokenization, Lemmatization, Stop Word Removal) is done, then a Provenance Matrix is made for all documents shown in figure11 . This Provenance Matrix and three binary matrices will be stored in database . Then, TDW Matrix based algorithm will be implemented having three phases :

a) Rendering Phase
b) Filtering Phase
c) Verification Phase

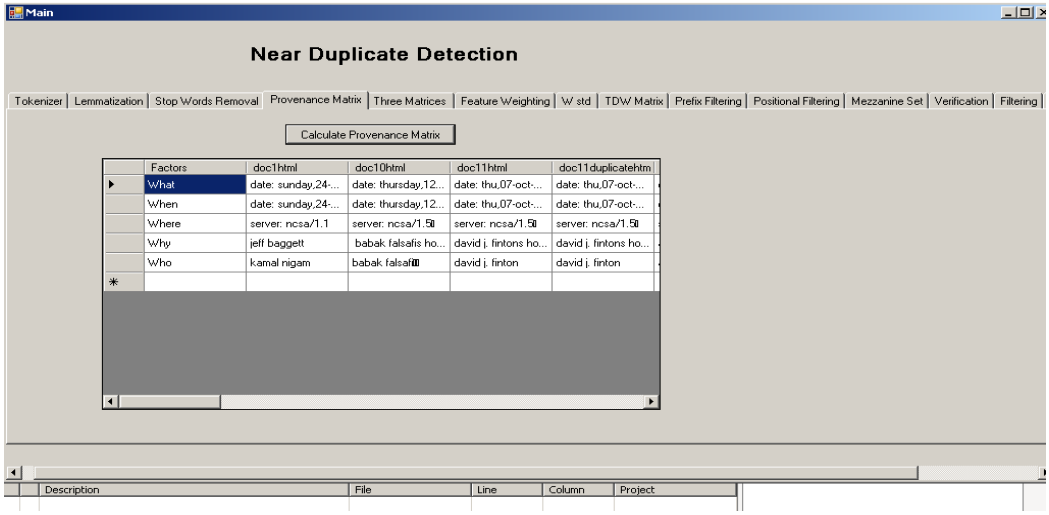

Figure11 : Provenance Matrix

Following Figure12 will show the feature weighting in Rendering Phase:

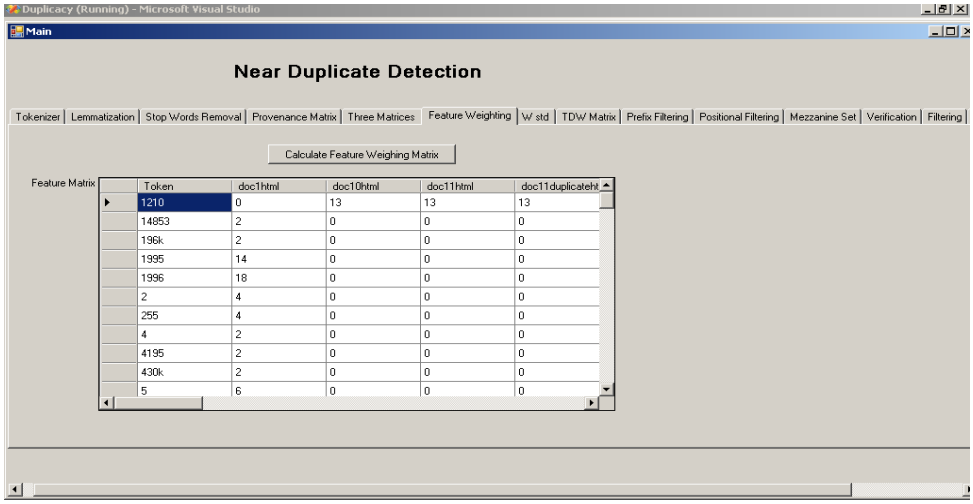

Figure 12: Feature Weighting

Filtering Phase helps in reducing the candidate sets and the final Phase of the TDW Matrix based algorithm is Verification Phase which is shown in figure 13 .

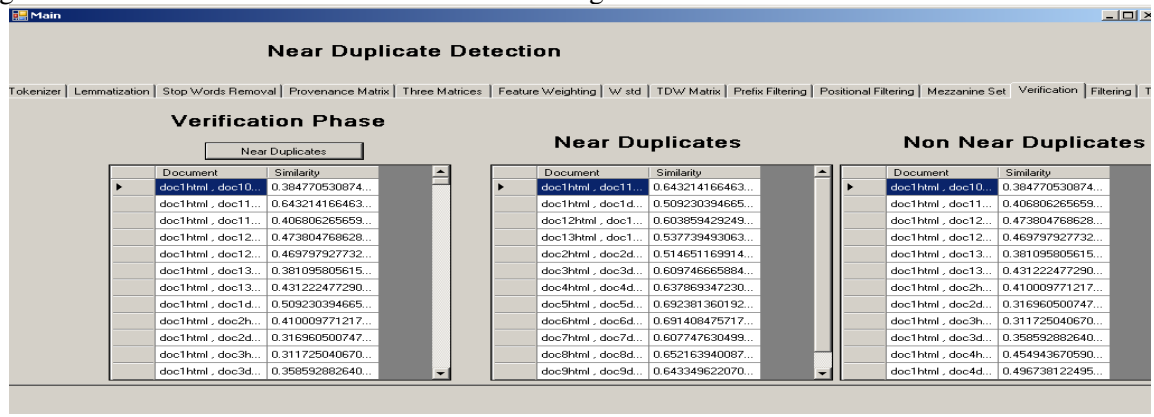

Figure13 : Verification Phase

This Verification Phase shows the Near-duplicates and Non-Near Duplicates.
After this Verification Phase , the near-duplicates were filtered according to Algorithm Filtering near-duplicates described in Proposed work. After this filtering, the trustworthiness calculation is done based on factors described in [6] and the refined results are given the form of figure8.

VI. CONCLUSION AND FUTURE SCOPE

In this paper, the proposed work is the hybrid model of Web Provenance and TDW-Matrix based algorithm which combines content, context, semantic structure and trust based factors for classifying and eliminating the results as original or near-duplicates. The approach used is the Web Provenance concept to make sure that the near duplicate detection and elimination and trustworthiness calculation is done using semantics by means of provenance factors (Who, When ,Where ,What ,Why , and How) and TDW Matrix based Algorithm concept aims at helping document classification in web content mining. So, it is concluded that the refined results are in the form of near-duplicates and ranked data, and also the outcome for 100 documents, in which 49 duplicates were present but the above implementation detect 48 relevant duplicates which provides precision and recall to be 97.95%.So, the experiments proved above that this work has better performance than both the methods individually.
In future, a further study will be made on the characteristics and properties of Web Provenance in near duplicate detection and elimination and also on the calculation method of trustworthiness in varied web search environments and varied domains. As a future work, the architecture of the search engine can be designed or a web crawler, based on web provenance for the semantics based detection and elimination of near-duplicates. Also, the ranking can be done based on trustworthiness values in

addition to the present link structure techniques which are expected to be more effective in web search. Also, further research can be extended to a more efficient method for finding similarity joins which can be incorporated in a focused crawler.

REFERENCES

[1] Broder A, Glassman S, Manasse M, and Zweig G(1997), Syntactic Clustering of the Web, In 6th International World Wide Web Conference, pp: 393- 404.
[2]Aleksander Kolcz, Abdur Chowdhury, Joshua Alspector(2004), Improved Robustness of Signature Based Near-Replica Detection via Lexicon Randomization,Copyright ACM.
[3] BarYossef, Z., Keidar, I., Schonfeld, (2007),U, Do Not Crawl in the DUST: Different URLs with Similar Text, 16th International world Wide Web conference, Alberta, Canada, Data Mining Track, pp: 111 – 120.
[4] Salha Alzahrani and Naomie Salim(2010), Fuzzy Semantic-Based String Similarity for Extrinsic Plagiarism Detection.
[5] Chuan Xiao, Wei Wang, Xuemin Lin(2008), Efficient Similarity Joins for Near-Duplicate Detection, Proceeding of the 17th international conference on World Wide Web, pp 131 – 140. April.
[6]Y. Syed Mudhasir, J. Deepika, S. Sendhilkumar, G. S. Mahalakshmi(2011), Near-Duplicates Detection and Elimination Based on Web Provenance for Effective Web Search in International Journal on Internet and Distributed Computing Systems. Vol: 1 No: 1
[7] Midhun Mathew, Shine N Das ,TR Lakshmi Narayanan, Pramod K Vijayaraghvan(2011), A Novel Approach for Near-Duplicate Detection of Web Pages using TDW Matrix, IJCA, vol 19-no.7,April
[8] Tanvi Gupta, Asst. Prof. Latha Banda(2012), A Novel Approach to detect near-duplicates by refining Provenance Matrix, International Journal of Computer Technology and Applications, Jan-Feb , vol(3),pp-231-234.

BIOGRAPHY

Tanvi Gupta received her B.E. degree in Computer Science from Maharashi Dayanand University in 2010 and her M.Tech degree in Computer Science from Lingaya's University Faridabad. Her areas of interest includes Web Mining, Text Mining, Network Security.

Latha Banda received her bachelor's degree in CSE from J.N.T University, Hyderabad, master's degree in CSE from I.E.T.E University, Delhi and currently pursuing her Doctoral Degree. She has 9 years of experience in teaching. Currently, she is working as an Associate Professor in the Dept. of Computer Sc. & Engg. at Lingaya's University, Faridabad. Her areas of interests include Data Mining, Web Personalization, Web Mining and Recommender System.