LSBSM: A Novel Method for Identification of Near Duplicates in Web Documents

Lavanya Pamulaparty
Research Scholar, Department of CSE
JNTUH, Hyderabad, India
lavanya.post@gmail.com

Dr. C.V. Guru Rao
Department of CSE, S R Engineering College
JNT University, Warangal, India.

Dr. M. Sreenivasa Rao
Department of CSE, School of IT
JNT University, Hyderabad, India

***Abstract*—Documents that are 100% similar are termed to be duplicate documents and near duplicate documents (NDD) are not bitwise identical but strikingly similar. If the NDD papers are clustered then they almost share the same cluster. The existence of near duplicate web pages are due to exact replica of the original site, mirrored sites, versioned sites, and multiple representations of the same physical object and plagiarized documents. The proposed algorithm comprises of three phases – Transliterate phase, filtering and Location Sensitive Bitwise Similarity method (LSBSM). It is to identify the query page is how similar to all the records in the repository. We have analyzed the system using the parameters like precision, recall, f-measure and efficiency, the results showed improvement in the values when compared with systems using existing weighting schemes which clarifies the efficiency of the proposed system. Mainly the elapsed time for the identification of near duplicate web pages has reduced and accuracy has increased.**

Keywords- Location Sensitive Bitwise Similarity, Near-duplicate detection, Web Crawlers.

I. INTRODUCTION

The WWW has witnessed exponential growth of web documents. The huge amount of data is downloaded by web crawler and finding the useful data during runtime is a challenge for search engine retrieve the data and Detecting duplicate documents and near duplicate documents will help search engines to improve their performance. The Dennis survey which uncovers that approximately 30% of web contents are near duplicates [1]. Near duplicate web pages are not bit-wise indistinguishable to one another but rather they bear a striking similarity [2]. Duplicate documents can be easily detected. However, detecting near duplicates is much harder (Sood & Loguinov, 2011; Jiang & Sun, 2011) [3]. The detection of the near duplicate pages help the accompanying the topical crawling, enhances the nature of indexed lists and recognition on spam [3], [4], [5]. Elimination of near duplicates saves network bandwidth, reduces storage costs and improves the quality of search indexes. It also reduces the load and remote host that is serving such Webpages [8]. Mathematically, NDD can be said that : Given a set of existing n documents D={d1,d2,d3…..dn}, a similarity function like Jaccard coefficient or cosine between document feature vectors, hamming distance between document signatures, Euclidean distance or a Dice function for w€ {0,1} and a new document dnew. Finding all documents d € D such that w(d, dnew) >= t, where t is the similarity threshold for NDD on similarity function w.

The remainder of the paper is structured as follows. Section 2 reviews literature on the prior works of near duplicate detection. Section 3 provides details of preliminaries pertaining to the proposed solution for near duplicate detection. Section 4 presents the proposed solution that implements our NDD algorithm. Section 5 presents experimental results while section 6 concludes the paper besides providing directions for future work.

II. RELATED WORK

This section review literature on detecting near duplicate documents. Yang and Callan [11] applied near duplicate detection method to near duplicate comments that are in electronic format. Text clustering and retrieving algorithms are used to detect duplicates. Deng and Rafiei [13] used stable bloom filters to detect duplicates in streaming data. The stable bloom filter has proved to be accurate and time efficient when compared with its predecessors. Bern Stein and Zobel [16] applied duplicate detection mechanism for identifying co-derivative documents. They employed hash – based algorithm for identifying duplicate chunks in the given dataset. Yang and Kallan [21] used instance – level constrained clustering for near duplicate detection. They used content structure and document attributes in the process of clustering. Their method showed that the algorithm is as accurate as human experts. Foo and Sinha [1] proposed a method for near duplicate detection of redundant bit vectors as part of image detection mechanism. They achieved 91% precision and 98% recall. Chang and Wang [8] employed near duplicate detection t digital libraries. They used sentence level approach for duplicate detection. Their method showed high accuracy and efficiency.

Theobald et al. [5] proposed an algorithm named “SpotSigs” that make use of extracting signatures from documents for near duplicate detection. Mehtha et al. [18] proposed near duplicate detection method for detecting image spam. Their method also uses visual features besides duplicate detection. Their solution showed 98% accuracy in detecting image spam. Huang et al. [24] explored in achieving high precision and high recall in near duplicate detection. They used Longest Common Sequence (LCS) method to achieve this. Chu

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

and Lin [6] applied near duplicate detection technique for consumer photo management application. Their work is based on filtering approaches such as probabilistic latent semantic, region based and point based.

Fisichella et al. [9] proposed a method known as locality sensitive hashing which is meant for near duplicate detection incrementally. They applied the technique for finding near duplicate detection of images. Hartrumpe et al. [20] proposed a method based on shallow and parser for near duplicate detection. In-depth analysis of near duplicate texts is explored that are useful for question answering systems and search engines. Bueno et al. [4] explored Bayesian approach for near duplicate detection of images. The specialty of this approach is that it uses local descriptors which are supported by decision theory for flexibility. Stoica [10] proposed Delaunay Diagram Representations for duplicate detection of images. All existing methods for near duplicate detection focus on detection of near duplicates. Our Proposed work not only concentrates on the text but also on images.

III. PROPOSED WORK

In this Paper, our proposed work has been carried out as for an input record r1, the near duplicate verification is done on set of n records stored in the repository {r1, r2, r3…..rn} and the ratio of similarity is also returned. The similarity verification is mainly based on the bit by bit comparisons. The duplicate and near duplicate detection is done by our novel method Location Sensitive Bitwise Similarity method (LSBS) is used to compute the difference. Our main objective is to find how to identify the query page is how similar to all the records in the dataset. A three stage approach is proposed.

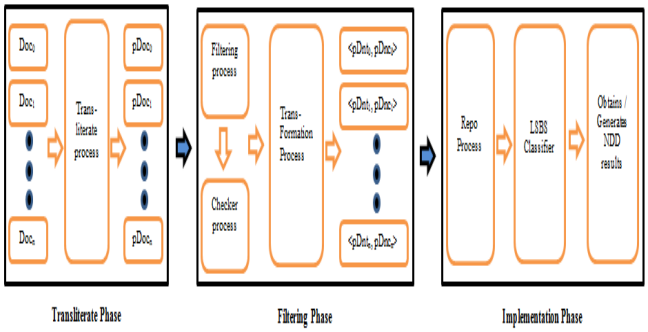

Figure 1. Three phase LSBS approach

The Document size Ds has also shown an important role in near duplicate detection. Our approach is considering the size of the new record set r1 with the whole record set Os= {d1, d2, d3, d4…dn}.The greater the difference of size the lesser chances of near duplicates.

One more advantage of the proposed approach is it provides a unique feature of searching near duplicates only within the relevant categories based on the document type like .pdf, .html, .doc, .txt etc. This method can be used as a tool for identification NDD in different categories.

In the first phase, called as Transliterate phase for the input query r the standard preprocessing methods like removal of whitespaces are applied and {x1,x2,x3…..xm} are retrieved where x is a sentence in a record set and m is the total number of sentences.

In the second phase, called as filtering phase the input file a text file is visualized in terms of the numbers by the ASCII equivalent and is converted to binary format to get the binary stream. For example 'R' uses 8 bits which is stored as 01010010. The document sizes are compared with the sizes of the individual files in the dataset and the decision is taken whether it has to be compared or not. Our approach concentrates not only on the text but also on images and special symbols. In the input page the text followed by images and special symbols are collected and converted to binary.

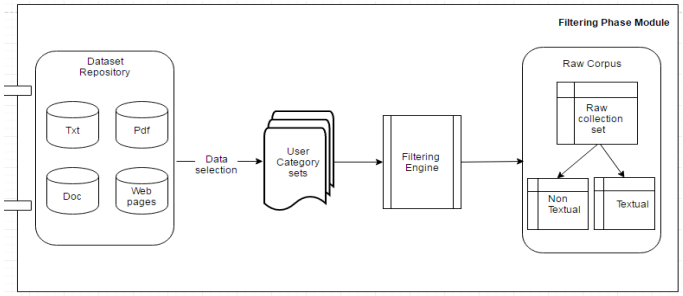

Figure 2. Filtering Phase

A. *Algorithm for Document Filtering Phase*

Input: The input page type can be any of the following Webpage, PDF Document, Word Document, Text Document, Records in the Repository (Record Set)

Output: Obtaining the filtered data from the documents in textual (Dnt) and non-textual (Dnc) content sets.

Steps:

1. Consider the given dataset Dn as

$$Dn = \sum_{t=1}^{n} Dnt$$

Dn <- *Total n number of docs considered in Dataset*

Dnt <- *Each doc considered in a given Dn*

2. In each Dn, extract

$$T_{LWS} <- \sum_{i=1}^{m} trim\left(\sum_{t=1}^{n} D(X)\right)_i$$

$\sum_{t=1}^{n} D(X)$ <- Identifying the lines in x document within the given D.

3. $Dnc <- \sum_{t=1}^{n} D(Sx)$

4. $Dnt <- \sum_{t=1}^{n} D(Tlws - Sx)$

$Tlws$ <- *trimmed line*

Sx* <- *Special Character line

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

In Filtering algorithm, Massive datasets of various categories were collected and maintained in the repository will be considered as databases. It will take only one type of categorical data as input at each run. Given dataset will be processed, trimmed and filtering the data within the considered dataset. Extract each and every document from the given dataset and trims the lines and additional spaces to obtain the raw corpus which stores in temporary dataset vectors. In the next step the raw corpus from temporary dataset will be categorized into the two vectors, One vector will parse the raw data and collects all the text oriented sets such as alphabets, numerical and so on where as the other vector collects non-textual data such as images.

In the third phase, The LSBS (Location Sensitive Bitwise Similarity) method name is abbreviated for our convenience. The LSBS is used to identify the near duplicate documents. The input record 'r' binary Data is converted into chunks of sizes 8 bits or 10 bits. The chunk size can be increased also per our requirement. We have considered a chunk size of 8 bits.

The current chunk size = Total characters/ (chunk size of 8 bits)

Current chunk1 = Txt0 (1:8)

Current chunk2= Txt1 (9:16)

….

Chunk n = Txtn (m-8: m)

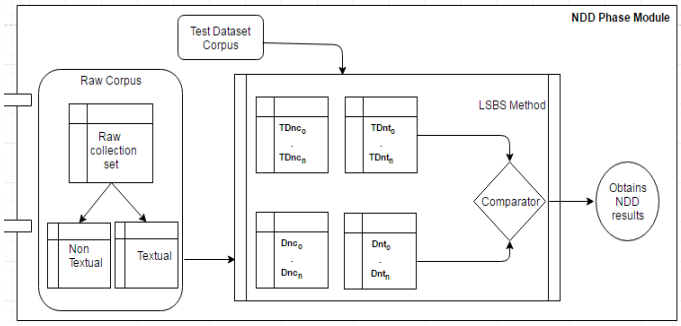

Figure 3. NDD Phase

B. *Algorithm for Near Duplicate Detection*

Input: The filtered input which any of the following Webpage, PDF Document, Word Document, Text Document, Records in the Repository (Record Set) and similarity threshold (t)

Output: Obtaining the LSBS featured sets for the given filtered content sets thereby results the NDD of documents.

Steps:

1. For a given filtered input, finding the chunks based on LSBS method as feature intervals and label the each featured chunks as cnk0,cnk1 … cnkn
2. Obtaining the chunks separately for Dnc and Dnt content sets
   a. Numbers of Chunks = (Dnt/Sz) V (Dnc/Sz)
   b. Sz <- chunk size (multiples of 8 bits, as we considered Sz as 8 bits only)
   c. e.g. cnk0 = txt0(0:7), cnk1 =txt1(8:15) … cnkn=txtn(n:m)
3. Identifying the similarity for all LSBS chunks obtained for Dnc and Dnt sets
   Featured Chunk (FC) <- LSBS_Method (Dtxti, cnki)
   The input document chunks are compared with every document chunks of the database.
4. LSBS method will give vote for each chunk if they are similar to the compared chunk in the repository. The numbers of votes are calculated.
5. Based on the threshold 't' given and similarity labels ratios, the NDD will be identified.
6. If the input document is an NDD then it is discarded otherwise added into the repository.
7. Steps 3 to 6 repeated for all documents in the database.

In NDD algorithm, it accepts two vectors as inputs to process further. Given vectors will be parsed and bit conversion will be made in the form of chunks up to the length of given vectors. Every feature conversion will be appropriately labeled and user considerable size chunks will be generated mostly in the multiple of eight. Here we considered the chunk size of eight bits. Every chunk in the vector will be compared with the given another dataset vectors as the bit-by-bit difference between the each chunk and the other chunks are found out and total number of bit difference is computed. The bit-by bit difference value is compared with the predefined threshold TB. If the bit difference value is less or equal to than the threshold, the input page is considered as a near duplicate page. If the difference value exceeds the threshold then it is not considered as NDD and it is added to the existing database.

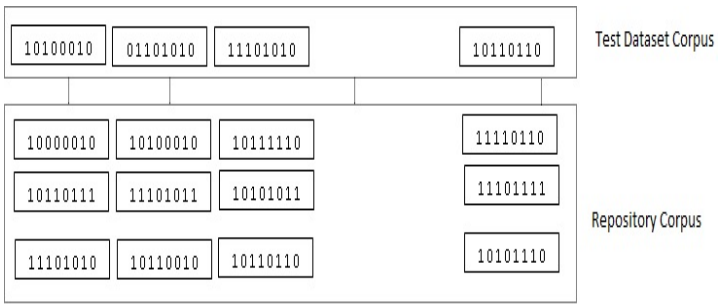

Figure 4. Dataset Repository

IV. Experimental Setup

After In this section, results of several experiments are presented. Comparisons are also done with other methods to demonstrate the effectiveness of our proposed approach. To conduct required experiments we used various datasets like Enron, RCV1, Reuters, and C50. The Enron document set (Enron email dataset, 2012) includes the mailbox e-mails of 150 users. Most of the users are executives of Enron. The Enron set contains 128,173 e-mails and takes 310 MB in size, as shown in Table 3. The RCV1 document set was edited and collected by Reuters. It includes 223,496 full English text news

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

from 20 August 1996 to 30 November 1996, as also shown in Table 1. Each news in RCV1 contains 109 words and 13 sentences on average.

TABLE I. TYPES OF DOCUMENTS USED IN EXPERIMENTS

Datasets	No.of Doc	Size(MB)
Enron	128,173	310
C50	5000	228
RCV1	223,496	652

We collected Webpages using an online tool wget to download the pages in voluminous amount. Whenever any input query which can be of any type like .pdf, .html, .doc, .txt file is given and its corresponding datasets are considered. The threshold is considered as user input and the LSBSM algorithm is run. The similarity is greater than the threshold percentage then is considered to be NDD and is discarded otherwise is stored in the database. All the programs that follow were implemented as a compact Java prototype using matlab and run on an Intel Core i5 quad-core CPU 2.80 GHz with 8.00 GB RAM.

A. *Experiment - I*

The The Enron document set [6] contains many spam e-mails which are just slight modifications of the original e-mails. So the set contains many e-mails which are similar to each other. The Enron document set (Enron email dataset, 2012) includes the mailbox e-mails of 150 users. Most of the users are executives of Enron. Then we compute similarity degree sim(X, Y). If sim(X, Y)> T, X and Y are labeled to be near-duplicates. Otherwise, X and Y are labeled as non near-duplicates. T is the threshold percentage entered by the expert like sim(X, Y)> 0.7.

B. *Experiment - II*

Headings, The C50 dataset is considered for the text documents evaluation. The experimental evaluation was done for a small sample of 10 pages to 1000 pages with different sizes and the time elapsed was collected. The following table represents the same.

TABLE II. DOCUMENT ELAPSED TIME

Data Set	Type	#No.of Documents	Size	Time(Sec)	Accuracy
C50	.txt	10	50KB	3.38	95
C50	txt	50	300KB	14.78	95
Enron	.pdf	10	300KB	196.20	96
Enron	.pdf	50	36.5MB	846.02	95
Reuters	.pdf	10	5.56MB	30.19	97

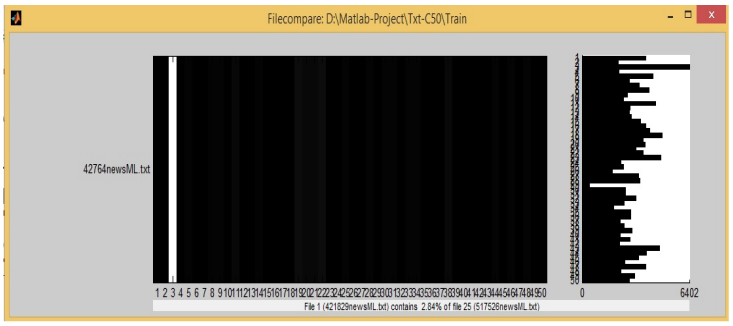

Figure.5. Training set computation

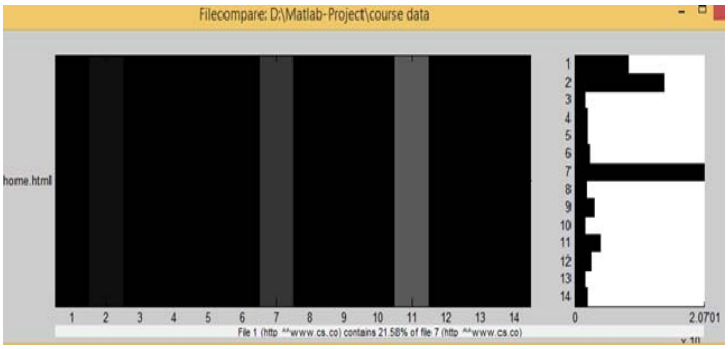

Figure 6. Computing the Near Dup document

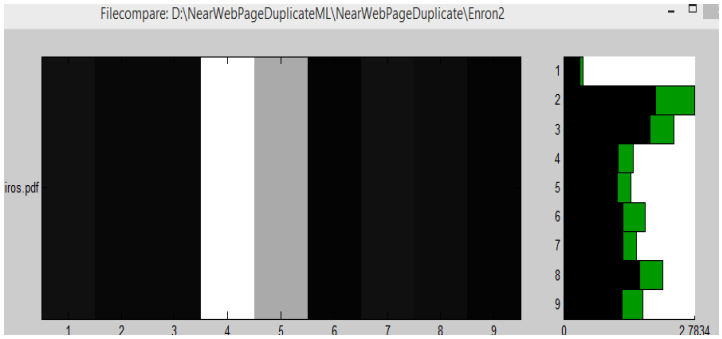

Figure 7. Identifying the Dup document

V. EVALUATION MEASURES

We the evaluations of the proposed approach are done using information retrieval effectiveness measures like precision and recall. Precision can be defined as the fraction of retrieved items that are relevant to all retrieved items or the probability given that an item is retrieved it will be relevant and recall as the fraction of relevant items that are retrieved to relevant items in the database or the probability given that an item is relevant it will retrieved [40]. The lower the values indicates bad performance of the system and the higher the values the more the user is encouraged to use the system due to the anticipation of getting more of the relevant search items. These evaluation measures are inter-dependent measures in that as the number of retrieved items increases the precision usually decreases while recall increases. In Table.3, the "Retrieved" documents are those that have been detected as duplicate by a duplicate detection algorithm, and the

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

"Duplicate" documents are really duplicates manually labeled by annotators.

Precision = P = A/A+B - (1)

Recall = R = A/A+C - (2)

TABLE III. FALSE ALARM – MISS RATE STRUCTURE

	Duplicate	Not Duplicate
Retrieved	A	B
Not Retrieved	C	D

We combined precision and recall values with F-measure [RIJ1979].It is the weighted harmonic mean of precision and recall.

The F-measure used in the study is given as follows:

$$F = 2 * P * R / P + R \quad - (3)$$

Accuracy: The accuracy is the perhaps the most intuitive performance measure. It is simply the ratio of correctly predicted observations i.e. Proportion of the true positives and true negatives. The input datasets of type txt, pdf, html, and doc are given to the LSBM approach to evaluate its accuracy. The computed values are plotted as a graph give in Figure 6. From the analysis of the above graphs we can infer that accuracy is good for the symmetric datasets like pdf, txt and doc where the class distribution is 50/50 and the cost of false positives and false negatives are roughly the same.

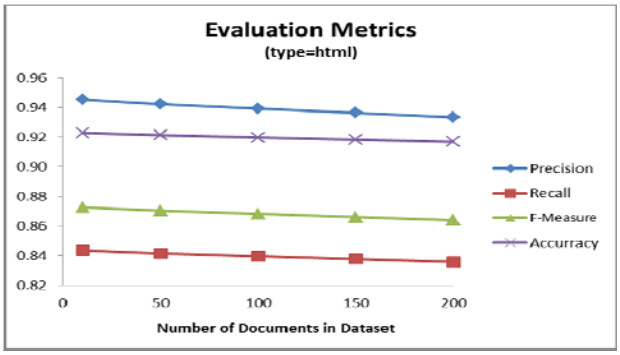

Figure 8.1 Metrics computations with html datasets

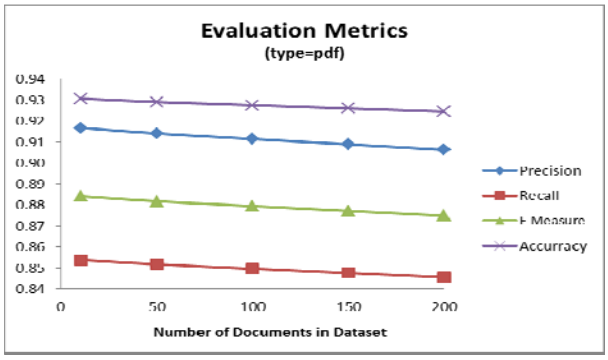

Figure 8.2 Metrics computations with pdf datasets

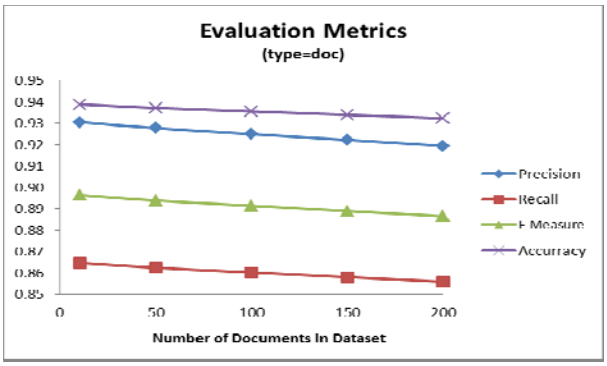

Figure 8.3 Metrics computations with doc datasets

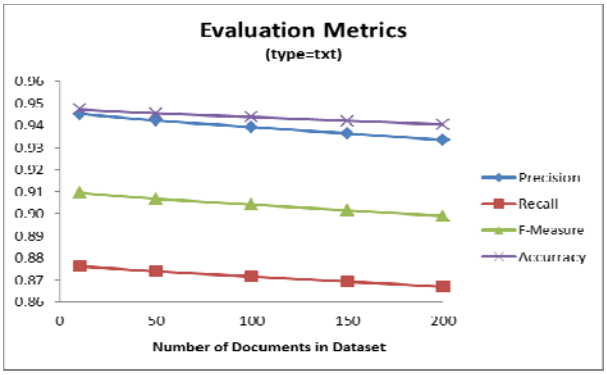

Figure 8.4 Metrics computations with txt datasets

Computation Time: For our proposed approach the computation time is considered as the time taken to compute the consolidated similarity score between the query and the reference web pages or documents in the database. . So, time incurred usually vary with the number of webpages or documents in the database. More number of documents in the database means that the new webpage have to be compared with more number of web documents. The time response for each is plotted in the Figure7 and the Table 4 gives the sample details. In the proposed approach the time taken to identify near duplicates is very less compared with the existing methods like Manku.

TABLE IV. COMPUTATION HISTORY LOG FOR DATASETS W. RT. THRESHOLD

Dn	Similarity Threshold	System Elapsed time(sec)			
		Txt	doc	Pdf	html/webpages
10	50	2.73	6.99	412.27	2.74
	60	2.73	7.68	443.93	3.79
	70	2.71	7.99	578.01	5.36
	80	2.66	8.35	614.64	6.47
	90	2.63	9.46	767.76	7.71
50	50	14.79	13.53	824.46	10.43
	60	14.72	14.90	886.48	11.79
	70	14.69	15.90	912.88	13.92
	80	14.68	16.80	932.90	14.43
	90	14.55	18.11	946.79	14.88

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

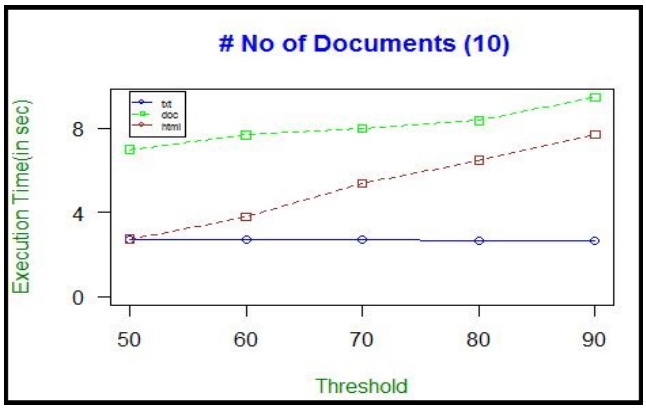

Figure 9.1 Elapsed time for the considered 10docs in datasets

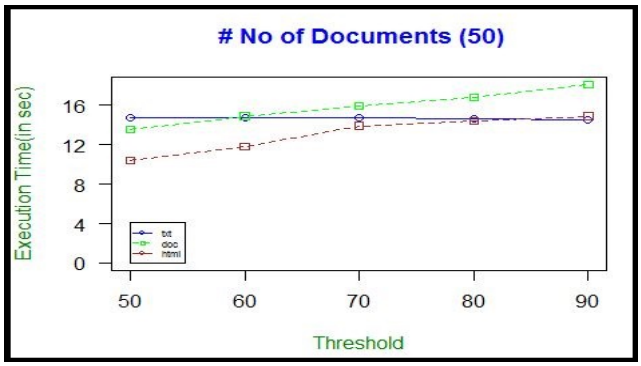

Figure 9.2 Elapsed time for the considered 50docs in datasets

VI. Effectiveness Measures

We tested the effectiveness of the algorithm using a corpus of various categorical documents. In order to perform this experiment, we extracted this many items from Enron, RCV1 and Reuters datasets from a respective open source sites and forums. We inspected each document from the respective category and with various in size as well.

With the same infrastructure testbed, we performed the experiments varying the corpus sizes and number of documents within the relative size. We found at performance of our algorithm comparatively as the best value at various threshold levels. The performances of LSBS decreases with higher values of corpus since more corpus require more memory.

We compared the effectiveness of our solution with other existing methods, i.e. SpotSig [11], Simhash [42] and Imatch [41].We executed these algorithms using the Java implementation provided by the authors of SpotSig [11], setting the default parameters. Figure 8 reports the performance values for several assigned threshold values.

We performed separated runs for each corpus, starting each run with an empty set. The more the hourly documents corpus grow, the more comparisons the algorithm performs, and the more the computation time increases, achieving the value of 0.8 seconds with 1000 number of documents within the corpus.

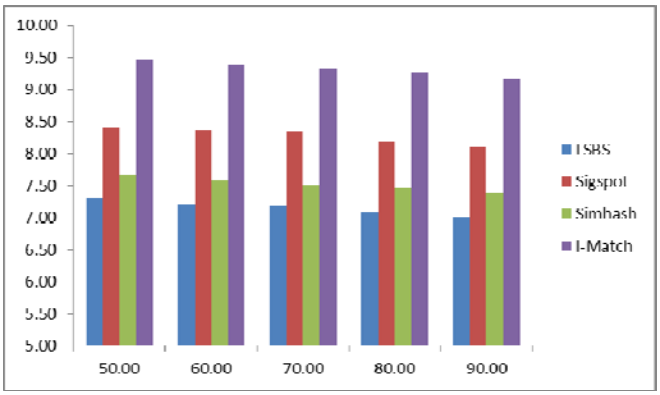

Figure 10. Elapsed time comparison with considered methods

CONCLUSION

The human community is dependent on internet for communication and information. It is consisting of voluminous amount of digital documents which are copied and pasted or modified continuously. During web crawling these are affecting space to store indexes, time, bandwidth and redundancy is frustration to the user. However identification is not at all an easy task. In this paper we have proposed a three phase efficient approach for detecting duplicate and near duplicate pages using Location Sensitive Bitwise Similarity method(LSBS).The main advantage of the proposed approach is it not only considers text but also images and hyperlinks. Our method can be applied to .doc, .docx, .pdf, .html documents. The experimental results have proved improved precision, Recall and F-measure. The execution time and accuracy for various types of documents are recorded. In future we would like to develop it as a tool for identification of duplicate and near duplicate documents which can be used in the study of data analytics.

References

[1] Dennis Fetterly, Mark Manasse, and Marc Najork. “On the Evolution of clusters of Near Duplicate Web Pages”. In proceedings of the first Latin American Web Congress, LA-WEB 2003, IEEE, ISBN)-7695-2058-8/03. 25, 32

[2] [2] Xiao, C, Wang, W. Lin, X. Xu Yu, J., "Efficient Similarity Joins for Near Duplicate Detection", Proceeding of the 17th international conference on World Wide Web, pp: 131-140., 2008.

[3] Conrad, J. G., Guo, X. S, and Schriber, C. P, “Online Duplicate Document Detection: Signature Reliability in a Dynamic Retrieval Environment”, Proceedings of the Twelfth International Conference on Information and knowledge Management, New Orleans, LA, USA, pp. 443- 452, 2003.

[4] Fetterly. D, Manasse, M, and Najork, M, “On the Evolution of Clusters of Near-Duplicate Web Pages”, Proceedings of the First Conference on Latin American Web Congress, pp.37.2003.

[5] Monika Henzinger, "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, pp: 284-291, 2007.

[6] Enron email dataset. (2012). <http://www.cs.cmu.edu/enron/>.

[7] Jun Jie Foo and Ranjan Sinha. (2007). Using Redundant Bit Vectors for Near-Duplicate Image Detection. LNCS. 0 (0), p472-484.

[8] Marco Baroni Æ Silvia Bernardini Æ Adriano Ferraresi Æ Eros Zanchetta. (2009). The WaCky wide web: a collection of very large

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

linguistically processed web-crawled corpora. Lang Resources & Evaluation. 0 (0), p210-226.

[9] Tanvi Gupta1 and Latha Banda. (2012). A HYBRID MODEL FOR DETECTION AND ELIMINATION OF NEAR- DUPLICATES BASED ON WEB PROVENANCE FOR EFFECTIVE WEB SEARCH. ISSN. 4 (1), p192-205.

[10] Lucas Moutinho Bueno, Eduardo Valle, Ricardo Torres. (2011). BAYESIAN APPROACH FOR NEAR-DUPLICATE IMAGE DETECTION.FAPESP. 0 (0), p1-4.

[11] Martin Theobald Jonathan Siddharth Andreas Paepcke. (2008). SpotSigs: Robust and Efficient Near Duplicate Detection in Large Web Collections. SIGIR. 0 (0), p1-8.

[12] Wei-Ta Chu and Chia-Hung Lin. (2010). Consumer Photo Management and Browsing Facilitated by Near-Duplicate Detection with Feature Filtering. -. 0 (0), p1-28.

[13] Matt Thomas, Bo Pang, and Lillian Lee. (2005). Get out the vote: Determining support or opposition from Congressional floor-debate transcripts. EMNLP. 0 (0), p1-10.

[14] Hung-Chi Chang1 and Jenq-Haur Wang2. (2007). Organizing News Archives by Near-Duplicate Copy Detection in Digital Libraries. ICADL. 0 (0), p410-419.

[15] Marco Fisichella, Fan Deng, and Wolfgang Nejdl. (2010). Efficient Incremental Near Duplicate Detection Based on Locality Sensitive Hashing. LNCS. 0 (1), p152-166.

[16] Adina Raluca Stoica,Annandale-on-Hudson. (2011). Delaunay Diagram Representations for Use in Image Near-Duplicate Detection. -. 0 (0), p1-85.

[17] Hui Yang, Jamie Callan. (2005). Near-Duplicate Detection for eRulemaking.-0 (0), p1-9.

[18] Henry S. Baird. (2014). Difficult and Urgent Open Problems in Document Image Analysis for Libraries. IEEE. 0 (0), p1-8.

[19] Fan Deng,Davood Rafiei. (2006). Approximately Detecting Duplicates for Streaming Data using Stable Bloom Filters. SIGMOD. 0 (0), p1-15.

[20] Bassma S. Alsulami, Maysoon F. Abulkhair, Fathy E. Eassa. (nd). Near Duplicate Document Detection Survey. ISSN. 2(2) (0), p147-151

[21] Y. Syed Mudhasir, J. Deepika, S. Sendhilkumar, G. S. Mahalakshmi. (2011). Near Duplicates Detection and Elimination Based on Web Provenance for Effective Web Search. Ijidcs. 1 (1), p22-31

[22] Yaniv Bernstein, Justin Zobel. (2006). Accurate discovery of co-derivative documents via duplicate text detection$. Elsevier. 0 (0), P595-609.

[23] Leonardo S. de Oliveira, Zenilton K. G. do Patrocínio Jr, Silvio Jamil F. Guimarães. (2013). Searching for Near-duplicate Video Sequences from a Scalable Sequence Aligner. -. 0 (0), P1-5.

[24] Bhaskar Mehta, Saurabh Nangia, Manish Gupta. (2008). Detecting Image Spam using Visual Features and Near Duplicate Detection. -. 0 (0), P497-506.

[25] Midhun Mathew, Shine N Das, T R Lakshmi, Pramod K. (2011). A Novel Approach for Near-Duplicate Detection of Web Pages using TDW Matrix. International Journal of Computer Applications. 9 (7), P16-21.

[26] Sven HARTRUMPF, Tim VOR DER BRÜCK, Christian EICHHORN. (2010). Detecting Duplicates with Shallow and Parser-based Methods. IEEE. 0 (0), p1-8.

[27] Hui Yang, Jamie Callan. (2006). Near-Duplicate Detection by Instance-level Constrained Clustering. -. 0 (0), P1-9.

[28] Fan Deng, Davood Rafiei. (ND). Estimating the Number of near Duplicate Document Pairs for Massive Data Sets using Small Space. -. 0 (0), P1-10.

[29] Ismet Zeki Yalniz, Ethem F. Can, R. Manmatha. (2011). Partial Duplicate Detection for Large Book Collections. -. 0 (0), P1-6.

[30] "Doe, John" "Mary K. Smith. (March 2, 2004). EnronDataset. Available: https://www.cs.cmu.edu/~./enron/. Last accessed 10th march 2015

[31] David D. Lewis. (2005). RCV1 (Reuters Corpus Volume 1). Available: http://www.daviddlewis.com/resources/testcollections/rcv1/. Last accessed 10th march 2015.

[32] Yuen-Hsien Tseng. (Nov. 8, 2002). Tools for Reuters-21578 Text Categorization Dataset. Available: http://lins.fju.edu.tw/~tseng/Collections/Reuters-21578.html. Last accessed 10th march 2015

[33] Wu, Y. et al (26, 3 2012). Efficient near-duplicate detection for Q&A forum. Retrieved from http://aclweb.org/anthology-new/I/I11/I11-1112.pdf

[34] Maosheng Zhong, Yi Hu, Lei Liu and Ruzhan Lu, A Practical Approach for Relevance Measure of Inter-Sentence, Fifth International Conference on Fuzzy Systems and Knowledge Discovery, pp: 140-144, 2008

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

[36] A. Broder, S. Glassman, M. Manasse and G. Zweig. Syntactic clustering of the web. In Proc. of the 6th International World Wide Web Conference, Apr. 1997

[37] Krishnamurthy Koduvayur Viswanathan and Tim Finin, Text Based Similarity Metrics and Delta for Semantic Web Graphs, pp: 17-20, 2010

[38] Salha Alzahrani and Naomie Salim, Fuzzy Semantic-Based String Similarity for Extrinsic Plagiarism Detection, 2010

[39] Lavanya Pamulaparty, Dr. C.V Guru Rao, Dr. M. Sreenivasa Rao. (2015). XNDDF: Towards a Framework for Flexible Near-Duplicate Document Detection Using Supervised and Unsupervised Learning.Procedia Computer Science. 48 (5), p228 – 235.

[40] C. D. Manning, et al., Introduction to Information Retrieval: Cambridge University Press, 2008.

[41] A. Kolcz et al. Improved robustness of signature-based near replica detection vialexicon randomization. In KDD 2004.

[42] P. Indyk et al. Approximate nearest neighbors: towards removing the curse ofdimensionality. In STOC 1998.

AUTHORS PROFILE

Lavanya Pamulaparty, Research Scholoar, Department of CSE, JNTUH, Hyderabad, obtained her Bachelor's degree in computer science from Nagpur University of KITS, Nagpur, India, and Masters Degree in Software Engineering from School of Informatics from JNT University Hyderabad, India, and Pursuing the PhD degree in computer science and engineering from JNT University. Her research interests include information storage and retrieval, Web Mining, Clustering technology and computing, performance evaluation and information security. She is a senior member of the ACM, IEEE and Computer Society of India

Dr. Guru Rao C. V received his Bachelor's Degree in Electronics &Communications Engineering from VR Siddhartha Engineering College, Vijayawada, India. He is a double post graduate, with specializations in Electronic Instrumentation and Information Science & Engineering. He is a Doctorate holder in Computer Science & Engineering from Indian Institute of Technology, Kharagpur, India. With 24 years of teaching experience, currently he is the Professor & Principal, SR Engineering College Warangal, India. He has more than 25 publications to his credit. He is a life member of Indian Society for Technical Education,Instrumentation Society of India and member of Institution of Engineers, Institution of Electronics & Telecommunications Engineers and Institution of Electrical & Electronics Engineers (USA).

Dr. M Sreenivasa Rao, Professor, School of Information Technology, JNT University, Hyderabad, obtained his Graduation and Post-graduation in Engineering from JNTU, Hyderabad and Ph D from University of Hyderabad. Over 28 Years of IT Experience in the Academia& industry. As a Dean of the MS IT Program, in association with Carnegie Melon University, USA. De-signed and conducted post graduations level MSIT program. Guided more than 10 research students in JNTU, and continuing the research in IT.

https://sites.google.com/site/ijcsis/
ISSN 1947-5500