Near Duplicate Web Page Detection using NDupDet Algorithm

Nilakshi Joshi
Lecturer, Don Bosco Institute of Technology,
Kurla(W), Mumbai, India

Jayant Gadge
Associate Professor, Thadomal Shahani
Engineering College,
Bandra(W), Mumbai, India

ABSTRACT

Web is a system of interlinked hypertext documents accessed via Internet. Internet is a global system of interconnected computer networks that serve billions of users worldwide. The huge amount of documents on the web is challenging for web search engines. Web contains multiple copies of the same content or same web page. Many of these pages on the Web are duplicates and near duplicates of other pages. Web search engines face substantial problems due to such duplicate and near duplicate web pages. These pages enlarge the space required to store the index, increase the cost of serving results and so frustrates the users.

To assist search engines to provide search results free of redundancy to users and to provide distinct useful results on the first page, duplicate and near duplicate detection is required. The proposed approach will detect near duplicate web pages to increase search effectiveness and storage efficiency of search engine.

Keywords

NDupDet algorithm, Near duplicate web pages, search engine, Web, URL

1. INTRODUCTION

Due to the rapid growth of information on Internet, there is huge increase in availability of Web documents. Information from World Wide Web is accessible anytime anywhere through the voluminous web repository. This quick expansion of information sources present on the Web had created the necessity to the users to make use of automated tools to locate desired information [1]. There is a need to use this big volume of information efficiently for effectively satisfying the information need of the user on the Web. Search engines become the major breakthrough on the web for retrieving this information [2]. Due to presence of voluminous data on the Internet, the efficiency of the search engine faces considerable problems.

In response to the query from the user search engine generates list of web pages for evaluation. This list of web pages may contain duplicate and near duplicate web pages. Two web pages are termed as duplicate if the contents of both the web pages are identical. Near-duplicate web pages bear high similarity to each other, yet they are not bitwise identical [3]. User has to spend efforts in separately evaluating these duplicate and near duplicate web pages sequentially to recognize the required results. These duplicate and near duplicate web pages increase the index storage space or slow down or increase the serving cost thereby irritating users [4]. Duplicate and near duplicate web page detection nowadays has become a separate field of research. The purpose of this research is to detect redundant web pages to increase search effectiveness and also increase storage efficiency for search engines. Near-duplicate detection aids in presenting only unique web pages to the user.

Several applications are benefited by the identification of the near-duplicate detection in the field of plagiarism detection, spam detection and in focused web crawling scenarios. It is also useful for technical document management, Digital libraries and electronic publishing, Database cleaning [2] [3].

2. RELATED WORKS

Recently, the detection of duplicate and near duplicate web pages has gained popularity. The detection techniques for identification of duplicate and near duplicate documents are reviewed in this section [1]. Broder et al. [5] have suggested shingling technique for near duplicate detection. This technique extracted all sequences of adjacent words; if two documents contain the same shingles set they are treated as equivalent. If the shingles set of the documents overlaps, documents are considered as exact similar. However it has been noted that this method does not work well on small documents. This approach was a syntactic approach for near duplicate web page detection.

Zahra Eskandari Gharghe et al. [6] suggested an adaptation of shingling and super shingling for weighted shingles. In this approach when comparing two documents to detect whether they are near-duplicate or not, both (super) shingles and their corresponding weights are considered. All steps are taken exactly like traditional shingling and super shingling with the exception that a weight is maintained for each shingle or super shingle. This weight is a representation of the (super) shingle's frequency in the document. The results have shown an improvement in shingling's performance. This approach for near duplicate web page detection is a syntactic approach for web page detection.

Junping Qiu et al. [7] show another approach for near duplicate web page detection. The detection process involves the following three steps. This approach for near duplicate web page detection is based on URL.

1. Removal according to URLs. First, remove pages with the same URL in the initial set of pages to avoid the same page been download repeated due to repeat links.
2. Remove miscellaneous information in the pages and extract the texts. Pre-treatment the pages, remove the navigation information, advertising information, html tags, and other miscellaneous information on the pages, extract the text content and get a set of texts.
3. Detect with DDW algorithm. Use the DDW algorithm to detect similar pages. Include the non-duplicate set of pages meeting the search criteria into the search results, mark the duplicate pages and record the eigenvector and similarity.

Narayana et al. [8] presented an approach for the detection of near duplicate web pages in web crawling. First the crawled web pages are stored into the repository. Then from these web pages the near duplicate web pages are detected. To detect near duplicate web pages keywords are extracted from crawled pages. The similarity score between two pages is calculated on the basis of extracted keywords. The two documents are considered as near duplicates if its similarity score satisfies a threshold value.

Salha Alzahrani et al. [9] suggested a method on plagiarism detection using fuzzy semantic-based string similarity approach. The algorithm was designed using four main stages.

The first step i.e. pre-processing step includes tokenization, stemming and stop words removal for a given document. In the second step a list of candidate documents for each suspicious document using shingling and Jaccard coefficient is retrieved. Then suspicious documents are compared sentence-wise with the associated candidate documents. In this step the computation of fuzzy degree of similarity that ranges between two edges 0 and 1 is performed.0 is used for completely different sentences and 1 is used for exactly identical sentences. Two sentences are marked as similar (i.e. plagiarized) if they gain a fuzzy similarity score above a certain threshold. In the last step i.e. post-processing step single paragraphs/sections are formed by joining consecutive sentences.

3. ARCHITECTURE OF PROPOSED SYSTEM

Fig.1 shows the proposed architecture of the system. The architecture aims at detecting near duplicate webpage from the dataset.

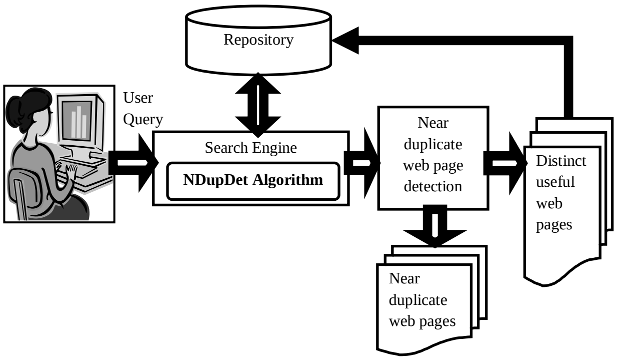

Fig 1: Proposed architecture of the system

User will provide query to the search engine. Search engine maintains its own repository. By applying near duplicate web detection the search engine will detect near duplicate web pages. Near duplicate web pages are not stored in the repository there by always maintaining the repository with distinct useful web pages.

Fig.2 shows the near duplicate web page detection process in detail.

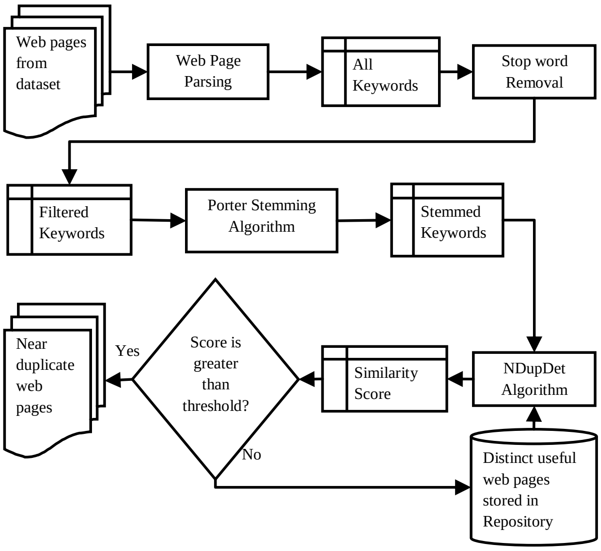

Fig 2: Near duplicate web page detection process

Web pages from the dataset undergo web page parsing for extracting keywords where HTML tags are removed from the web pages. Stop word removal is the process where the stop words like ‘a’, an’, the’, ‘in’ etc are removed from the keywords and provides filtered keywords to the stemming process [8]. Porter stemming algorithm[10] is used to stem the related keywords; for example ‘process’, ‘processes’, ‘processing’ are considered as ‘process’. The stemmed keywords and the existing data base of the web pages become input to the NDupDet algorithm. SimScore obtained as output of the NDupDet algorithm is considered for deciding if the web page is near duplicate. If the web page is not near duplicate then it is stored in the database for future reference else the web page is not stored in the database.

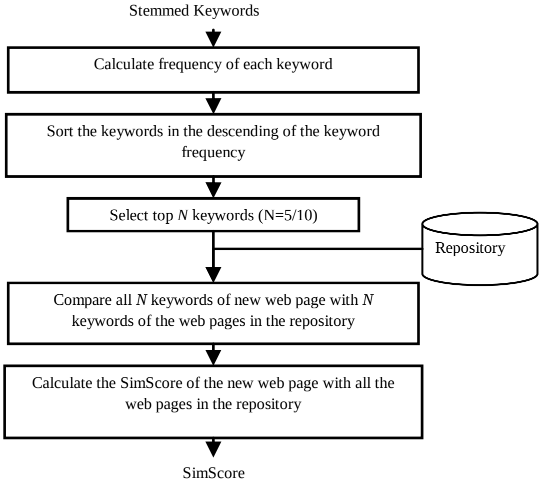
Stemmed Keywords
Calculate frequency of each keyword
Sort the keywords in the descending of the keyword frequency
Select top N keywords (N=5/10)
Repository
Compare all N keywords of new web page with N keywords of the web pages in the repository
Calculate the SimScore of the new web page with all the web pages in the repository
SimScore

Fig 3: NDupDet Algorithm

Fig.3 shows the details of NDupDet algorithm. NDupDet algorithm consists of three major steps. The first step is to calculate the frequency of each keyword in a web page w1. Then in the second step these keywords are sorted according to the descending order of their frequencies. The third step is to select top N keywords from this list. Value of N is set to 5 and 10. Whenever a new page w2 is coming and needs to be stored in the repository, its similarity with the web pages already existing in the repository needs to be compared. For this purpose only these top N keywords are used. SimScore is calculated using following formula

$$\text{SimScore} = \frac{\text{No. of common keywords in w1 and w2}}{\text{Top N Keywords under consideration}} \times 100$$

Where w1 is web page 1 and w2 is web page 2.

4. EXPERIMENTAL SETUP

For this experiment the query terms related to Operating System are selected. The five query terms selected are, “Process”, “Deadlock”, “Thread”, “Virtual Memory”, “Memory”.

Above mentioned keywords are given as query term one at a time to Google search engine. From first 5 pages of the search result, web pages are selected. The count of web pages considered in each datasets is represented in Table 1.

Table 1. Web page dataset details

Sr.No.	Web Pages data set	Count of web pages considered
1	Process	13
2	Deadlock	10
3	Thread	14
4	Virtual Memory	22
5	Memory	11
Total		**70**

5. RESULTS

Every dataset mentioned in the above section undergoes web page parsing, stop word removal and Porters Stemming algorithm processes. It is observed that after applying the mentioned steps above the no. of keywords are reduced to an extent. The final output that is stemmed keywords are provided to the NDupDet algorithm. The results are obtained after executing the NDupDet on the above dataset are shown in Table 2 and Table 3. As shown in Fig. 4 and Fig. 5, SimScore range is divided into 3 categories that 1% to 39%, 40% to 59% and 60% to 99%. To get Table 2 and Table 3 all the web pages from that dataset are checked against every page of that dataset.

Table 2. NDupDet results for N=5

Data set	No of web pages	SimScore for 5 Keywords		
		1% to 39%	40% to 59%	60% to 99%
Process	13	100%	46%	13%
Deadlock	10	90%	70%	10%
Thread	14	86%	57%	14%
Virtual Memory	22	95%	91%	22%
Memory	11	100%	91%	45%

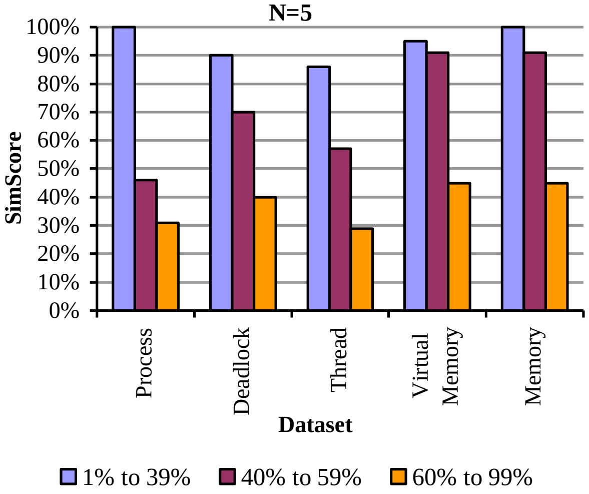

Fig 4: SimScore illustrations for N=5

When N=5, and SimScore >=60% as illustrated in the Fig.3, Maximum 45% of the web pages are detected to be near duplicate. There are no unique web pages identified, therefore possibility of detecting near duplicate web pages is high for this keyword count (N=5).

Table 3. NDupDet results for N=10

Data set	No of web pages	SimScore for 10 Keywords		
		1% to 39%	40% to 59%	60% to 99%
Process	13	100%	31%	15%
Deadlock	10	100%	60%	30%
Thread	14	93%	29%	0%
Virtual Memory	22	95%	73%	9%
Memory	11	100%	73%	0%

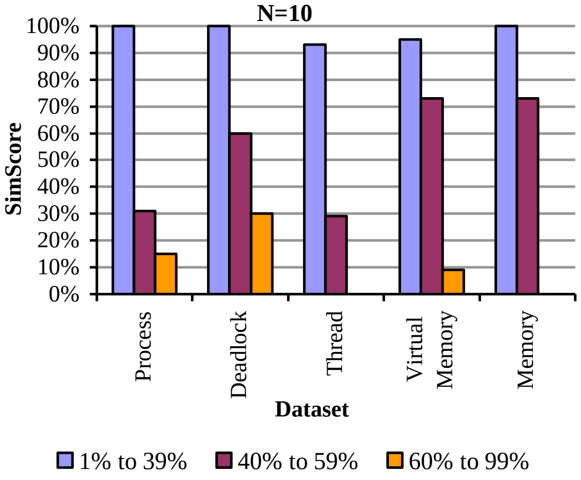

Fig 5: SimScore illustrations for N=10

When N=10, and percentage similairty >=60% as illustrated in the Fig.4, Maximum 30% of the web pages are detected to be near duplicate. 35% of the web pages are detected to be unique web pages, therefore possibility of detecting near duplicate web pages is low for this keyword count (N=10).

6. CONCLUSION AND FUTURE WORK

Web pages with similarity score of 0% represent totally unique documents. Web pages with similarity score of 1% to 39% are not similar and needs to be maintained in the database for future reference. Web pages with similarity score of 40% to 59% are suspicious near duplicates and needs to be maintained in the database for future reference.

Furthermore only top N (N=5 or N=10) keywords are stored in the repository for each web page instead of all keywords; memory space for the repository has got reduced. Since web pages having SimScore greater than 60% are not stored in the repository, by detecting them as near duplicate web pages, the search efficiency is also achieved.

Web pages which are near duplicates may appear closer to each other in search results, but provide little benefit to the user. The future work can be research for more robust and accurate method for near duplicate detection and elimination on basis of the detection. To increase the accuracy of the NDupDet Algorithm a weighing scheme can also be applied to the Top N keywords.

The proposed approach does not consider the position of the keyword in the document. These positional filtering can also be used for finding the near duplicate web page detection. Also after successful completion of the near duplicate distinct useful web pages and near duplicate detection the web pages can be ranked using ranking algorithm for keeping useful results on the first 3-5 pages of the search results for user's reference. Further to increase more accuracy these near duplicate web pages can be eliminated from the search results.

7. REFERENCES

[1] J Prasanna Kumar, P Govindarajulu ,"Duplicate and Near Duplicate Documents Detection: A Review" European Journal of Scientific Research ISSN 1450-216X Vol.32 No.4, pp.514-527,2009

[2] Bassma S. Alsulami, Maysoon F. Abulkhair, Fathy E. Eassa, "Near Duplicate Document Detection Survey",International Journal of Computer Science & Communication Networks,Vol 2(2), 147-151,2010

[3] Midhun Mathew, Shine N Das, T R Lakshmi Narayanan, Pramod K Vijayaraghavan, "A Novel Approach for Near-Duplicate Detection of

[4] Web Pages using TDW Matrix", International Journal of Computer Applications (0975 – 8887)Volume 19– No.7, April 2011

[5] 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

[6] Zahra Eskandari Gharghe, Behrouz Minaei Bidgoli,"Weighted shingling: an adaptation of shingling for weighted shingles",2009 IEEE

[7] Junping Qiu and Qian Zeng, Detection and Optimized Disposal of NearDuplicate Pages, 2nd International Conference on Future Computer and Communication, Vol.2, pp: 604-607, 2010.

[8] V.A. Narayana, P. Premchand and A. Govardhan, "Effective Detection of Near-Duplicate Web Documents in Web Crawling", International Journal of Computational Intelligence Research, Volume 5, Number 1, pp. 83–96, 2009.

[9] Salha Alzahrani, Naomie Salim, "Fuzzy Semantic-Based String Similarity for Extrinsic Plagiarism Detection Lab Report for PAN at CLEF", 2010

[10] M.F. Porter, "An algorithm for suffix stripping Program", 14 no. 3, pp 130-137, July 1980.