Near-replicas of Web Pages Detection Efficient Algorithm based on single MD5 fingerprint

WANG DA-ZHEN[1,2] , CHEN YU-HUI[1]
(1.Department of Computer Science, Hubei University of Technology, Wuhan, P.R.C 430068)
(2.Department of Information Management, Wuhan University, Wuhan, P.R.C 430074)

http://www.newtelesoft.com

Abstract: - We consider how to efficiently compute the overlap between all pairs of web documents. This information can be used to improve web crawlers, web archives and in the presentation of search results, among others. Our experiments show that how common replication is on the web, and testified that our algorithm is better than others.

Key-Words : Near-replicas, MD5, Fingerprint

1 Introduction

Many documents are being replicated across the World Wide Web. For instance, there are several copies of .NET FAQs and Linux manuals on the net. Many of these copies are exactly the same, while in some cases the documents are "near" copies. For instance, documents may be older versions of some popular document, or may be in different formats (e.g.HTML, or Postscript), or may have additional buttons, links and images that make them slightly different from other versions of the document. Replication also occurs at a higher level than documents. In some cases, entire servers are mirrored across different countries and updated daily, weekly or on a monthly basis. The Linux Documentation Project (LDP) page with Linux manuals is one such instance, with over 180 mirror servers across the world. In some cases servers are exclusively dedicated to maintaining LDP pages and are therefore exact or near-replicas; in other cases there are several servers that replicate LDP manuals along with other manuals http://sunsite.unc.edu contains manuals on LDP, FreeDOS, JAVA, etc.) In this paper, we concentrate on computing pair wise document overlap; we put through a new algorithm and contrast with traditional algorithm, from the emulations we can similarly solve the pair-wise server overlap problem if we consider the server to be the union of all documents in the site and our new algorithm is better than old method.

1.1 Related work

Manber considered computing pair-wise document overlap in the context of finding similar files in a large file system [1]. Researchers at the DEC SRC Lab are developing a similar tool to "syntactically" cluster the web [2]. Heintze has developed the KOALA system for plagiarism detection [3]. Narayanan and Garcia-Molina has developed COPS [3] and SCAM [4] experimental prototypes for finding intellectual property violations. All the above techniques use variations of the same basic idea--compute "fingerprints" for a set of documents and store into a database. Two documents are defined to have significant overlap, if they share at least a certain number of fingerprints. The above systems have different target applications, and therefore differ in how they compute fingerprints of documents, and how they define the similarity measure between a pair of documents.

In this paper, we do not propose new notions of document similarity: We primarily concentrate on efficiently solving the "clustering" problem of computing overlap between all document-pairs simultaneously. This problem is especially hard when the number of documents is on the order of millions. The techniques we use to solve this problem are different from the techniques adopted by reference work [5,6,7,8]. Our new approach to solving the all-pairs document overlap problem is based on efficiently executing fingerprint's calculation [9]. As we will see, our techniques take orders of magnitude less space and time.

Current techniques of detection near-replicas Web pages are mostly base on content-based detection. It's generally said there are three methods of above techniques.

① Near-replicas of Web Pages Detection based on keywords.[1]

② Near-replicas detection by computing overlap between all document-pairs.[9]

③ Near-replicas detection by template noise filter.

In WebGather system, Zhang had given an algorithm using VSM to contrast different web pages based on single MD5 fingerprint, we can get that time complexity and space complexity is $O(N^2)$

and $O(N)$. The shortcoming of this method is that algorithm based on the keyword generations. It's difficult to generate accurate keywords of one web page. Ma had put through another method using text tiling first then contrast different paragraph fingerprint of web pages, this algorithm using text tag of HTML files divide files into differ block, then choice paragraph make MD5 fingerprint, we can get that time complexity and space complexity is $O(\mu m^2 N^2)$ and $O(mN)$ .The vital defect of this algorithm is burden of time complexity and space complexity, not suit the real situations.

II. Near-replicas of Web Pages Detection based on single MD5 fingerprint

The efficiency of the above implementation depends critically on how often fingerprints occur across documents. This is because above algorithm produces across-product of all document pairs that share a fingerprint and subsequent steps process the resulting cross-product. Hence if fingerprints tend to be common across documents, the output may be too large.

Roughly, our approach to computing the fingerprint of document by a given chunking as follows. We first compute document three maximum paragraphs that are exact copies using the by other web pagers. This step is useful in cases when the number of exact copies is very large, since it reduces the work required by the subsequent, less efficient steps. We then using these three chunks of the document and compute fingerprints for the document by MD5 method, and then we can detect the near-replicas of same web pages.

The details procedure of algorithm as follows:

① Segment Extraction: In web pages, we can easy get different segment of web page every segment by using HTML tags. For instance, in HTML file, a paragraph usually has <P> and </P> tag the identify it.

② Segment Sorting: Sorting the segments length array that distillation from step ① according to the length of differ segments. The big size segments usually representative the main content of web page, it's also can be seem as a noise filter procedure.

③ Chunk building: Get the first n number segments in segments array in step ② and concatenate it as a big chunk.(In out emulation, n is 3, adjust by differ users).

④ Filter stop words: We know that there many stop words in document such as blank id , interpunction id and etc because of editions by different authors, we can delete it is our program and get new chunk result.

⑤ Fingerprint generate: Using chunk result in step ④ and MD5 encoding algorithm, we can get the MD5 fingerprint of web page.

⑥ Fingerprint comparison: Compare fingerprint different web pages, if the fingerprint is same, we can draw conclusion they are near replicas web pages.

In our algorithm, concatenate the t big segment of web pages into one chunk and generate a MD5 finger print, it means m=1, we can get the time and space complexity is $O(N^2)$ and $O(N)$, it better than old algorithms. The follow is comparison of different methods:

Table 1 Time and Space Complexity

Algorithm	Time	Space
WebGather system	$O(N^2)$	$O(N)$
Garcia-Molina's method	$O(\mu m^2 N^2)$	$O(mN)$
Our method	$O(N^2)$	$O(N)$

III. Experiments

We random used 100,000 of web data for our experiments. We ran our experiments on a BigPC, 1G MBs of RAM and 1.6 GBs of swap space, running Red Hat 9.0C with dual processors. We computed fingerprints of each document using our chunking method.The follow is contrast of our algorithm and others in time cost and space cost.

In the follows, we using CB represent Garcia-Molina 's method, TW as WebGather system method, SMFA as our methods.

3.1 Time complexity

Table 2: Time

Pages	10000	100000	100000
TW	3M	110M	253M
CB	2M	420M	1386M
SMFA	2M	80M	140M

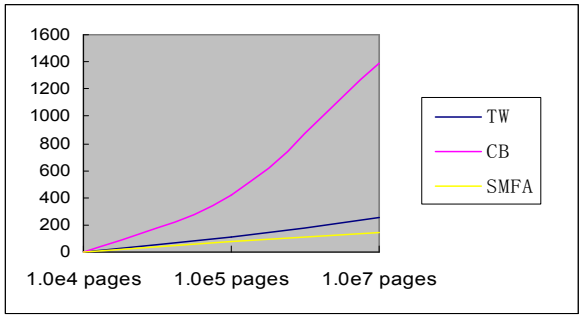

Figure 1 Time cost of different algorithm

We can see that the three algorithms have little difference when the test data set only contain 10000 web pages, however, there are huge difference when the scale of test data reach to 1000000 web pages. CB algorithm is not suit for near replicas detection when the scale is large. Our methods is better than TW algorithm is time cost.

3.2 Precision and Recall

As for precision and recall, we are using 1000000 random web pages as test dataset, the result as follows:

Table 3 Precision and Recall

Algorithm	Precision (%)	Recall (%)
TW	0.966	0.718
CB	0.974	0.813
SMFA	0.983	0.846

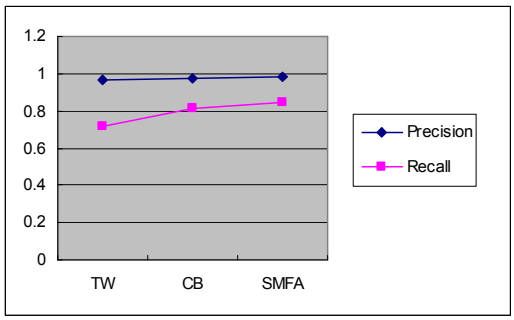

Figure 2 Precision and Recall

We can draw conclusion that precision of SMFA and CB algorithm is better than TW algorithm. The recall has little different in three algorithm.

IV. Conclusion

Based on the experiments we computed, we can see our algorithm is better than others. One proposal diminishes the time complexity and space complexity contrast to old algorithms. Furthermore, we testified that our algorithm is suit for huge data set.

References

1. Shivakumar and Garca-Molina,1998, "Finding near-replicas of documents on the web." WebDB 1998. pp: 204-212.

2. Junghoo Cho, Hector Garcia-Molina "Finding replicated Web collections." In Proceedings of 2000 ACM International Conference on Management of Data (SIGMOD), May 2000.

3. Andrei Broder, Steve Glassman, Mark Manasse, and Geoffrey Zweig. "Syntactic clustering of the Web." In Proceedingsof the 6th Int'l World Wide Web Conf.(WWW), pages 391—404,1997.

4. Z. Zhang, J. Chen, and X. Li, "A preprocessing framework and approach for web applications," Journal of Web Engineering, vol. 2, pp. 175--191, 2004.

5. Narayanan Shivakumar and Hector Garcia-Molina, "SCAM: A Copy Detection Mechanism for Digital Documents." In Proceedings of the 2nd International Conference on the Theory and Practice of Digital Libraries(DL'95), June 1995.

6. Lan Yi, Bing Liu, Xiaoli Li. "Eliminating Noisy Information in Web Pages for Data Mining." In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD-2003), Washington, DC, USA, August, 2003

7. J. Dean and M. Henzinger. "Finding Related Pages in the World Wide Web." In Proceedings of the 8th International World Wide Web Conference (WWW), 1999.

8. B. Davison. "Topical Locality," In the Web. Proceedings of SIGIR, 2000. [Haveliwala, et al., 2002] T. Haveliwala, A. Gionis, D. Klein, and P. Indyk., "Evaluating Strategies for Similarity Search on the Web." In Proceedings of the 11th International World Wide Web Conference (WWW), May 2002

9. WebGather System,Technical Report, http://e.pku.edu.cn.