International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Impact Factor (2012): 3.358

Reducing Duplicate Content Using XXHASH Algorithm

Rahul Mahajan[1], Dr. S.K. Gupta[2], Rajeev Bedi[3]

[1]M.Tech Student, Beant College of Engineering & Technology, Gurdaspur, Punjab, India

[2]HOD & Associate Professor (Computer Science & Engineering Department),
Beant College of Engineering and Technology, Gurdaspur, Punjab, India

[3]Assistant Professor (Computer Science & Engineering Department),
Beant College of Engineering and Technology, Gurdaspur, Punjab, 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. With the rapid growth of information and the explosion of Web pages from the World Wide Web, it gets harder for search engines to retrieve the information relevant to a user. However, the performance of a web search is greatly affected by flooding of search results with information that is redundant in nature. Removing redundant content is an important data processing operation in search engines and other web applications. The existing architecture of WWW uses URL to identify web pages. A large fraction of the URLs on the web contain duplicate (or near-duplicate) content. Web crawlers rely on URL normalization in order to identify equivalent URLs, which link to the same web pages. Duplicate URLs have brought serious troubles to the whole pipeline of a search engine, from crawling, indexing, to result serving. De-duping URLs is an extremely important problem for search engines, since all the principal functions of a search engine, including crawling, indexing, ranking, and presentation, are adversely impacted by the presence of duplicate URLs. In this we have proposed a new technique for reducing duplicate content during web crawling and saving only unique pages in the database.***

Keywords: Duplicate, Duplicate Content, Normalization, Web, Web Crawler

1. 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. The performance and reliability of web engines thus face huge problems due to the presence of enormous amount of web data. The voluminous amount of web documents has resulted in problems for search engines leading to the fact that the search results are of less relevance to the user. Search engines use crawlers to collect Web pages from Web Servers distributed across the Internet. Crawlers are the programs that automatically collect Web pages by starting with a Uniform Resource Locator (URL), downloading the Web page at that location and recursively retrieving all the pages pointed to by the hyperlinks on the page. Several previous studies [4, 3, 5] have established that a large fraction of the web consists of duplicate URLs — syntactically distinct URLs having similar content. These duplicate URLs adversely affect the performance of commercial search engines in various ways. In crawling, they waste valuable bandwidth, affect refresh times, and impact politeness constraints; in indexing, they consume unnecessary disk space; in link-based ranking, they impart disproportionate authority to undeserving URLs; in presentation, they pollute displayed search results and lead to a poor user experience. Crawler resources are wasted in fetching duplicate pages, indexing requires larger storage and relevance of results are diluted for a query An estimate by [6] shows that approximately 29 percent of web-pages are duplicates and the magnitude is increasing. De-duping URLs is an extremely important problem for search engines, since all the principal functions of a search engine, including crawling, indexing, ranking, and presentation, are adversely impacted by the presence of duplicate URLs.

2. Related Work

Conventional methods to identify duplicate documents involved fingerprinting each document's content and group documents by defining a similarity on the fingerprints. Many elegant and effective techniques using fingerprint based similarity for de-duplication have been devised [1, 9, 10]. [9, 10] also emphasized and showed results with large scale experiments. However, with the effectiveness also comes the cost of fingerprinting and clustering of documents. Recently, more cost-effective approach of using just the URLs information for de-duplication has been proposed, first by Bar-Yossef et.al. [11] and extended by Dasgupta et.al. [2]. The standard URL normalization mechanism has been studied and extended by [7] and [8]. Lee et al. extended the standard URL normalization mechanism with three additional steps, which include changing the path component of the URL into lower case, eliminating the last slash symbol at the non-empty path component and eliminating the default pages [8]. The default pages considered are default.html, default.htm, index.html and index.htm. In [7], four evaluation metrics were defined, which are URL consistency, URL applying rate, URL reduction rate and true positive rate. The elimination of the trailing slash symbol was also proposed to further extend the standard URL normalization mechanism. Another related work was done by Schonfeld et al. where they studied the patterns of how URLs are constructed within a particular website, followed by constructing DUST (different URLs with similar text) rules [11]. For an instance, the DUST rule ".co.il/story_" → ".co.il/story?id=" will replace ".co.il/story_" in all the URLs

with ".co.il/story?id=". Since each website may observe different formatting in its URLs, the DUST rules must be mined separately for each Website. As of June 2008, Netcraft charted 174 millions of websites available on the WWW [8]. As such, it is practically infeasible if DUST rules are to be mined from each website individually.

3. Problem Definition

However, there are many syntactically different and yet equivalent URLs, which point to the similar web pages. For examples, two pairs of equivalent URLs which have been identified by our proposed URL signatures are http://weavebytes.com/wbmain/internship.php equivalent to http://weavebytes.com/wbmain/indst_training.php. These pairs would not be identified by the standard URL normalization mechanism since they are syntactically different. These syntactically different but equivalent URLs have motivated us to explore the possibility of eliminating these syntactically different and yet equivalent URLs using the body content of the corresponding web pages. In this paper, the metadata considered are the URL and the body text of the web pages. In above example only URL normalization is not enough as syntactically different URL can lead to same page contents. So, we need a technique that not only take into account URL but also something additional that will allow the crawler to identify whether the contents of two different URL are equal or not. For this reason a technique was invented called URL signature.

4. Proposed Work

The goal of our proposed URL signature is to reduce the processing of redundant web pages downloaded through equivalent URLs that are syntactically different. We have implemented signature on both URL and on the body text of the web page. To reduce the time taken to generate the signature, its compulsory to reduce the size of body text because crawler efficiency is not only depends to retrieve maximum number of relevant pages but also to finish the operation as soon as possible. We have implemented XXHASH algorithm in our approach which identify duplicate pages in minimum span of time resulting in faster efficiency of crawler.

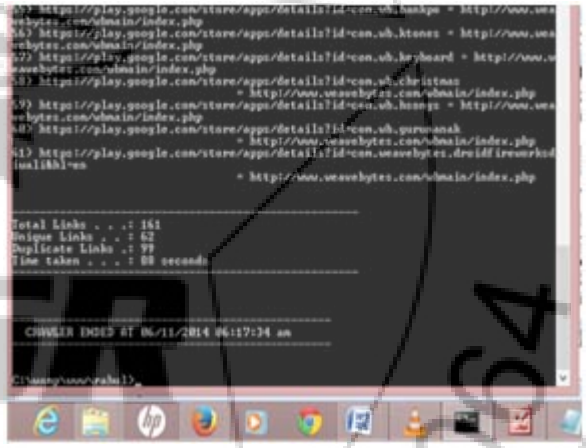
5. Algorithm

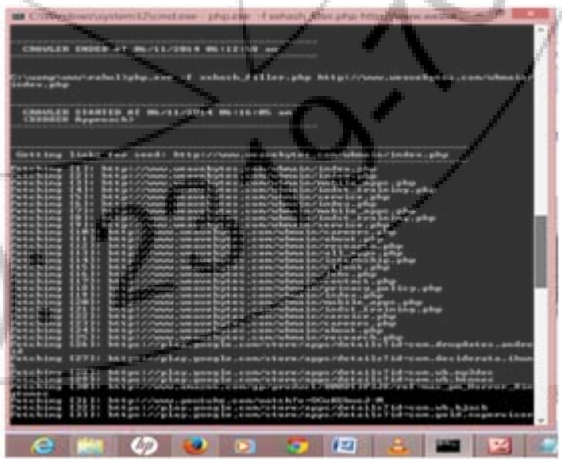
1. Our crawler starts with a seed URL.
2. The crawler fetches the corresponding web page from WWW and parse that page in order to extract the links present in that web page. These links which are fetched will be stored in the queue called frontier queue.
3. The URL which are present in the frontier queue will be normalized using standard normalization and those URL which are redundant will be removed.
4. Now we will generate a signature of the URLs by implementing XXHASH algorithm. If the signature of the URL exist in the database, the URL will be reject and will not be further processed. If not, crawler will fetch the web page and the signature will be updated in the database.
5. After fetching the web page , we will generate another signature of the page content of the web page that will allow the crawler to identify whether the contents of two different URL are equal or not. If the signature of the page content exists in the database, the URL will be rejected and will not be further processed. If not, the signature will be updated in the database.
6. After performing two way duplicacy check, the URL which are left are unique in nature and will be saved in the database.
7. This two way duplicacy check helps us to save only unique pages resulting in saving valuable bandwidth.

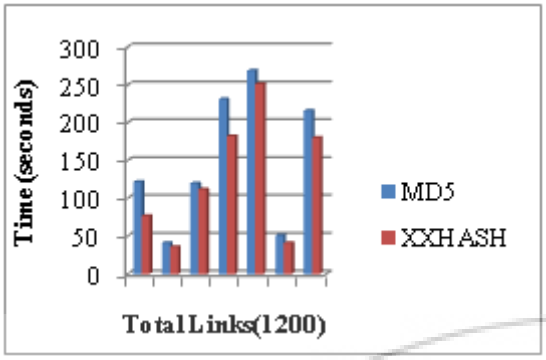
6. Experimental Setup

The main objective of the experiment reported in this paper is to evaluate the performance of a web crawler with we have embedded URL signature to identify duplicate pages in less amount of time. We have implemented our work on machine which consists of Intel i3 processor, 4 GB RAM, 500 GB hard disk. We have used PHP and WAMP Server as our tool for conducting our work. We have tested our work on sample websites.

7. Results







8. Conclusion and Future Work

Web crawlers rely on the standard URL normalization which transforms the URLs into a canonical form in order to eliminate equivalent URLs which link to redundant web pages. However, only redundant web pages identified by syntactically equivalent URLs could be avoided. Being aware of such limitation, we have proposed to incorporate the semantically meaningful metadata of web pages linked by the URLs to reduce the overhead caused by processing these redundant web pages multiple times. In our proposed method, we construct two URL signature one on the URL and other on the body text of the web page to represent the downloaded web pages. We have implemented a new hash function XXHASH which is extremely fast as crawler efficiency is not only depends to retrieve maximum number of relevant pages but also to finish the operation as soon as possible. For future work, we also plan to explore the possibility of incorporating other metadata of web pages to dynamically construct the URL signatures. In our experiment, all the URLs in our dataset link to web pages which have web contents in ASCII characters. Hence, another interesting future direction would be to investigate the suitable hashing methods for web pages which contain unicode characters.

References

[1] Broder. On the resemblance and containment of documents. In *SEQUENCES '97: Proceedings of the Compression and Complexity of Sequences 1997*,page 21, June 1997.
[2] Dasgupta, R. Kumar, and A. Sasturkar. De-duping urls via rewrite rules. In *KDD '08: Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 186-194,August 2008.
[3] Pereira, R. Baeza-Yates, and N. Ziviani, "Where and how duplicates occur in the web" In Proc. 4th Latin American Web Congress, pages 127–134, 2006.
[4] D. Fetterly, M. Manasse, and M. Najork, "On the evolution of clusters of near-duplicate web pages" In Proc. 1st Conf. on Latin American Web Congress, page 37, 2003.
[5] G. S. Manku, A. Jain, and A. D. Sarma, "Detecting near-duplicates for web crawling", In Proc. 16th WWW, pages 141–150, 2007.
[6] Kim, S. J., Jeong, H. S., and Lee, S. H., "Reliable Evaluations of URL Normalization", in Proceedings of the 2006 International Conference on Computational Science and its Applications (ICCSA), Glasgow, May 2006, pp. 609 – 617.
[7] Lee, S. H., Kim, S. J., Hong, S. H., "On URL Normalization", in Proceedings of the 2005
[8] International Conference on Computational Science and its Applications (ICCSA), Singapore, May 2005, pp. 1076 – 1085.
[9] M. Henzinger. Finding near-duplicate web pages: a large-scale evaluation of algorithms. In *SIGIR '06:Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 284{291, August 2006.
[10] G. S. Manku, A. Jain, and A. D. Sarma. Detecting near-duplicates for web crawling. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 141{150, May 2007.
[11] Z. Bar-Yossef, I. Keidar, and U. Schonfeld. Do not crawl in the dust: different urls with similar text. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 111-120, May 2007.