International Journal of Engineering &Technology, 7 (3.12) (2018) 361-365

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Removing Duplicate URLs based on URL Normalization and Query Parameter

Kavita Goel[1*], Jay Shankar Prasad[2], Saba Hilal[3]

[1]Research Scholar, Dept of Computer Science, MVN University, Palwal, India
[2]Dept of Computer Science & Engg., MVN University, Palwal, India
[3]http://sabahilal.blogspot.in/India
**Corresponding Author E-mail: [1]kavitagoyalmca@gmail.com, [2]jayshankar.prasad@mvn.edu.in*
[3]saba21hilal@gmail.com

Abstract

Searching is the important requirement of the web user and results is based on crawler. Users rely on search engines to get desired information in various forms text, images, sound, Video. Search engine gives information on the basis of indexed database and this database is created by the URLs through crawler. Some URLs directly or indirectly leads to same page. Crawling and indexing similar contents URLs implies wastage of resources. Crawler gives such results because of bad crawling algorithm, poor quality Ranking algorithm or low level user experience. The challenge is to remove duplicate results, near duplicate document detection and elimination to improve the performance of any search engine. This paper proposes a Web Crawler which performs crawling in particular category to remove irrelevant URL and implements URL normalization for removing duplicate URLs within particular category. Results are analyzed on the basis of total URL Fetched, Duplicate URLs, and Query execution time.

Keywords**:** *URL Normalization, Query Parameter, Categorization, Duplicate URLs, Execution time.*

1. Introduction

Web is rapidly growing in size, and thus chances of similar contents or duplicate contents become a problem for the user. To extract information from web, search engine has become an important tool for extracting that information [1]. Extracting information is done by a component called crawler [1, 18]. It is necessary to extract useful information from huge collection of web page, to extract information there are many techniques and tools [4]. Also the data is of different types (text, audio, pictures, multimedia) etc [1]. In other words, we can say that every web crawler uses different technique to extract links and try to give relevant results [7]. Also data can be extracted in number of ways: Content which include text and multimedia [1]; data mined on the basis of usage include server logs [2] and structure which includes analyzing information from link structure of web [3].

Web crawler is a program which automatically downloads the web pages and search engine stores it in repository for indexing [4].The process of crawling takes time to crawl and space to stores URLs in search engine. This Crawling process increases the traffic to great extent as huge number of URLs is visited by web crawler. Some of these URLs directly or indirectly lead to same page. In order to reduce traffic, administrator bound to implement robot.txt file. Robot.txt file is a specification of sites which not to be visited by crawler [1].

Presence of duplicate URLs, affects whole searching process which includes crawling, indexing and relevance factor [5]. WWW identifies web pages using the URL or URI [6]. Different URLs with same text are found on web. These URLs are recognized as DUST [7, 18].

Normalization is the process to determine that two syntactically different URLs are actually same or not [7, 19].URL normalization is used to remove duplicate results or web pages in crawling and hence searching results.

Many researchers have done work in the area of removing duplicate URLs or web pages but problem still persists [2, 8,12].

This paper contributes in the area of optimizing performance of web crawler by removing irrelevant and duplicate URLs. Irrelevant URLs are removed using categorized crawling and to remove duplicate URLs, URL normalization is implemented.

For experiment and comparison of duplicate and irrelevant URLs two crawlers are implemented.

Base crawler: It is based on Breadth First Search Algorithm. Base crawler simply follows the links when crawls the web and stores result in database. After completion of crawling it displays total time taken in crawling and total rows stored in database.

Proposed Crawler: Follows Breadth first search in addition to URL normalization for duplicity removal and Categorization crawling for removing irrelevant URLs.

A search interface is also provided which take keyword from user and generate results from database created by crawler. In this way, it can also be compared with existing search engines.

Crawler's efficiency is evaluated on the basis of Time taken in executing a query, number of rows fetched from database and number of duplicate records fetched from it.

Proposed crawler is based on categorized crawling and URL normalization. During crawling time, user specifies the category in which he/she wants to crawl a particular URL. For e.g. if user crawls a URL http://www.apple.com he will also specify the category in which apple will be crawled as Apple can be phone or it can be fruit. This way search results will be improved and crawler will search specific area of web which will result in

reduced crawling time and space in database. It will only store relevant records. This paper is organized as follows. In Section 2, the background of the research is presented. In Section 3, the design and implementation of proposed crawler is given. In Section 4, the experimental result based on comparison of base crawler and proposed crawler has been represented. Section 5 contains the conclusion of paper.

2. Background

Search engines use web crawlers to collect documents from the web [1]. Search engines rely upon the performance of crawlers and crawler's performance depends upon many factors such as crawling time and space [4, 6]. As web is growing exponentially in size, high crawler's performance is required. The web has million and billion of pages and growing daily in size [4].In order to improve the process of crawling, many researchers have work in the area of improving crawler performance. Performance of crawler can be improved by improving challenges of web crawling [18, 19]. Shestakov [8] defines challenges of web crawling .Among many challenges one such challenge is detection and removal of near duplicate URLs and content. Many researchers have worked in the area of detecting and removal of near duplicate links. Ziv BarYossef [5], consider the problem of DUST (Different URLs with same text) and presents some rules which transforms given URL to others that may have same content [18]. Lay-Ki Soon [6], suggests method of identifying equivalent URLs by using metadata (page size and body text) of web pages along with basic URL normalization. Tao Lei [9], uses pattern tree approach to redefine the rewrite rules of URL normalization. Anirban Dasgupta [10], focused on de-duping of URLs by partitioning the URLs into equivalence class and writing rules that transform those URLs into same canonical form. Houqing Lu [11], improves the focused crawler works to obtain more relevant web pages based by building web page classifier on term weight approach. Hong-Wei Hao [12]**,** implemented topic specific news gathering system based on TF-IDF and LSI. Banu Wirawan Yohanes [13], uses genetic algorithm to optimize the process of web crawling and to select the relevant pages. Sotiris Batsakis [14], improves the performance of focused crawler using Hidden Markov Model. In [15], Salim Khalil implements multithreaded crawler based on R package. Crawler is optimized and performs content extraction and duplicate content detection. In [16], Xiaochen Zhang optimizes distributed crawler by modification of parameters and model under Hadoop platform. Fengyun Cao [17], defines a two level scheduling algorithm that considers network performance and web quality pages and improves crawler's efficiency in terms of time and space. Semantic similarity used for improving the performance of search engine [19].

3. Design and Implementation of Proposed Crawler

URL normalization and categorization are used to design proposed crawler that will reduce duplicity among the result.
URL normalization is a technique to recognize that two syntactically different URLs are actually different or not [19]. Syntactically means, if one URL starts with HTTP and other with HTTPS [19]. It considers it same URL and removes the one. If First URL contains default.aspx and next URL contains index.html, then these URLs are same and it removes one. Removes default port 80 and 443. Segments '.' And '..' can be removed. Trailing hash '/' and '/.' will be removed. Like these there are many other factors that are considered to remove duplicity among URL.
Categorization states that URLs with same features are placed together in categories usually for some specific purposes. Like all Urls related to sports will stored in sports category. Initially there are 10 categories defined in which URLs will be crawled and searched. Design and implementation of propose crawler based on above technique is defined below.

A. Design of Proposed Crawler

In order to start the process of crawling, category of website, name of website and URL is required.
Same type of URLs is combined into one category. Like websites of all educational institutes are grouped under education category. In the same way, 10 different categories are created. Category contains links of same type. Crawler will crawl only in category in which it is ask to crawl. For e.g., to crawl apple the category user want to crawl becomes helpful because it may be fruit or phone or laptop.
In order to remove duplicate in each category URL normalization is used.URL normalization specifies some rules which helps to identify that two syntactically different URLs are actually different or not.

1) Rules used in designing the proposed crawler are defined below:

Step 1: Handle Domain Duplicity
"WWW" and "non WWW" URL, HTTP and HTTPS URL, URL with port 80,443 and without port, URL ending in '.' Or '..',URL with single '/' and double '//' are treated as same URL during comparison.
Step 2: Sort the Parameter
Resulted URL will be sorted on the basis of query parameter. During Sorting all parameters will be converted into lower case and parameters with multiple values will also be sorted on the basis of key/pair value.
Step 3: Remove Duplicate Query Parameter
After sorting, duplicate query parameter will be removed from the URL. If value of one key/pair is equivalent to other key/pair then one will be removed.
Step 4: Remove Empty Parameter.
Once duplicate parameters are removed, the next step is to remove empty parameters and empty query from URL. It will remove BLANK values but 0 will be considered as a value.
Step 5: Remove Social Parameter
Social tracking parameter will also be removed from URL. Resulted URL will be compared with all other URLs and based on comparison result it will be saved or deleted.
To understand the above process, the algorithm is applied and explained as below:

Example:

http://www.mnc.com:80/../a/b/./c?d=1&e=3&f=4&D=2&f=4&g=0&h=?
1) After applying first step port 80 and '..' and '.' segments will be removed and resulted URL will be
http://www.mnc.com/a/b/c?d=1&e=3&D=2&f=4&f=4&g=0&h=?
2) In the next step, after applying sorting the resulting URL will be
http://www.mnc.com/a/b/c?d=1&d=2&e=3&f=4&f=4&g=0&h=?
3) After sorting, duplicate query parameter will be removed and resulted URL will be
http://www.mnc.com/a/b/c?d=1&d=2&e=3&f=4&g=0&h=?
4) After removing duplicates BLANK values will be removed and resulted URL will be
http://www.mnc.com/a/b/c?d=1&d=2&e=3&f=4&g=0
This is the final URL which will be stored in database.

2) The interface for proposed crawler is shown in fig. given below:

Fig. 1: Design interface of Proposed Crawler

B. Implementation of the Proposed Crawler

In order to fulfill the objectives; two crawlers are created i.e. Base crawler and Proposed Crawler. Base crawler is based on Breadth First Search Algorithm. Proposed crawler follows Breadth first search in addition to URL normalization and categorization. There are two databases used for the purpose. One is SE database that is used to store results given by base crawler. Another is SE_Proposed database that is used to store results fetched by proposed crawler.

To test the efficiency of above program, Search interface is also designed which is shown in Fig. below:

Fig. 2: Search Engine interface of crawler.

User will enter a keyword and select the field. Search engine will generate result based upon URLs stored by the crawler in the database. In proposed crawler, the entire URLs are stored in specific category in which keyword need to be searched, hence crawler will search in that category. It will save time in searching result and also it will generate related and relevant results to query.

4. Experimental Results

In this section experimental results for base crawler and proposed crawler are compared and described. To test the effectiveness of proposed algorithm, different categories are selected. The experimentation performed on different URLs in each category on same server. Below are the findings.

A) List of Categories**:** For experimentation of crawler 10 different categories are selected. The categories selected are mostly of user interest and it can be changed as per the requirement also.

B) List of Urls: In ten different categories numbers of different URLs are taken for experiment. Three to Four URLs are taken in each category. Table I show the categories and associated few URLs.

Table 1: List of Categories And URLS In Each Category

Category	URL
Education	https://collegedunia.com
	https://www.mvnrepository.com
	www.amity.edu
	https://www.mvneducation.com
	http://www.du.ac.in
	http://www.mvn.edu.in
Health	https://allayurveda.com
	http://www.eatingwell.com
	https://www.top10homeremedies.com
	http://www.7daygmdiet.com
	http://www.stylecraze.com
Sports	https://www.olympic.org
	www.worldbadminton.com
	https://www.britannica.com
	www.skysports.com
	http://www.cricbuzz.com
Science	www.physicsclassroom.com
	www.astronomy.com
	https://www.space.com
	https://wonderopolis.org
	http://www.nineplanets.org
Travel	https://www.oyorooms.com/
	https://www.booking.com
	https://www.goibibo.com
	http://www.transindiatravel.com
	www.mycozytrip.com
Politics	www.thehindu.com
	rajyasabha.nic.in
	www.livemint.coms
	https://www.indiabix.com
Business	http://www.moneycontrol.com
	http://www.bseindia.com
	https://www.surfexcel.in
	https://www.bigbasket.com
	https://grofers.com
	http://www.india-crafts.com
Others	https://www.tasteofhome.com
	www.countryliving.com
	www.healthandyoga.com
	https://www.target.com
	https://www.pestcontrolindia.com
World	https://www.worldbank.org
	http://www.un.org
	https://www.worldpoliticsreview.com/
	https://www.endpolio.org
Transport	http://www.besttransport.com
	https://www.redbus.in
	https://www.yatra.com
	www.travelyaari.com
	https://www.mapsofindia.com

C) Execution Time: Execution time is the time taken by search engine to return results in response to user query. It can be defined as the time interval between users enters the query and the time he gets the results. The less execution time is desired. The more execution time means poor performance of search engine.

Experimental results in terms of search engine execution time for Base and Proposed crawler are shown in Table II and Table III respectively

Table 2: Time Taken by Base Crawler's Search Engine

Keyword	**Category**	**Execution time**
Mvn	NA	5.2 ms
Health	NA	6.7 ms
Yoga	NA	3.4 ms
Fitness	NA	6.2 ms
Olympic	NA	5.1 ms
Taste of home	NA	6.2 ms
Surf excel	NA	4.5 ms
Oyo rooms	NA	5.2 ms
Astronomy	NA	6.8 ms

Table 3: Time Taken by Proposed Crawler's Search Engine

Keyword	Category	Execution time
Mvn	Education	0.9 ms
Health	Health	1.6 ms
Yoga	Health	2.8 ms
Fitness	Health	2.2 ms
Olympic	Sports	3.4 ms
Taste of home	Other	4.8 ms
Surf excel	Business	2.3 ms
Oyo rooms	Travel	4.0 ms
Astronomy	Science	3.1 ms

D) Duplicate records: Duplicate records are such records which is available at more than one place on web. The Table IV given below depicts the total no. of rows fetched for a particular keyword. Also, it depicts number of duplicate entries in a keyword search. As no duplicate removal algorithm is implemented. Table V and VI shows number of records fetched in response to user query and total duplicate records fetched.

Table 4: Total Records and Duplicate Records Fetched byBase Crawler

Keyword	Category	Number of rows fetched	Number of duplicate records
Mvn	NA	13	2
Health	NA	32	5
Yoga	NA	8	0
Fitness	NA	5	0
Olympic	NA	18	2
Taste of home	NA	12	1
Surf excel	NA	6	0
Oyo rooms	NA	9	1
Astronomy	NA	19	1

Table 5: Total Records and Duplicate Records Fetched By Proposed Crawler

Keyword	Category	Number of rows fetched	Number of duplicate records
Mvn	Education	3	0
Health	Health	18	2
Yoga	Health	1	0
Fitness	Health	1	0
Olympic	Sports	8	1
Taste of home	Other	2	0
Surf excel	Business	1	0
Oyo rooms	travel	1	0
Astronomy	Science	14	1

E) Improvement of Proposed crawler over Base crawler

Table shows improvement in the performance of proposed crawler over base crawler. Two parameters are considered in terms of search query execution time and duplicate records. Search engine execution time is the time computed as the difference between the users enters the query and the time he gets the result. The next metric shows the reduction in number of duplicate results in Search query.

Table 6: Comparison of Execution Time and Duplicate URL Reduction between Base Crawler and Proposed Crawler

Keyword	Category	Search Query Execution time improvement (%)	Duplicate URL Reduction (%)
Mvn	Education	5.77	200
Health	Health	4.1	250
Yoga	Health	1.2	0
Fitness	Health	2.8	0
Olympic	Sports	1.8	100
Taste of home	Other	1.2	100
Surf excel	Business	1.9	0
Oyo rooms	travel	1.3	0
Astronomy	Science	2.1	100

5. Conclusion

URL normalization and categorized crawling implemented in crawler outperforms. It generates better results. The proposed duplicate removal algorithm is tested on 40 URLs and results are compared with base crawler. Due to limited hardware resources we cannot compare execution time with existing crawlers and search engines, so base crawler is designed to show the results of experiment. The experiment shows that the results of proposed crawler are better than base crawler in terms of execution time, memory utilization and duplicate records. In future work categorization and normalization can be improved to generate more relevant and less duplicate results.

References

[1] M. Shoaib and A. K. Maurya. "URL ordering based performance evaluation of Web crawler", International Conference on Advances in Engineering & Technology Research, Unnao, pp. 1-7. 2014 doi: 10.1109/ICAETR.2014.7012962.

[2] Jiang, J. Pei. And H.Li. "Mining search and browse logs for web search: A Survey", Journal of ACM Transactions on Intelligent Systems and Technology, vol.4, no. 4, 2013.

[3] L. Getoor, "Link Mining: A New Data Mining Challenge" .SIGKDD explorations, pp. 1-6, 2003,

[4] K. S. Kim, K. Y. Kim and K. H. Lee, et al. "Design and implementation of web crawler based on dynamic web collection cycle", The International Conference on Information Network 2012, Bali, 2012, pp. 562-566, 2012 doi: 10.1109/ICOIN.2012.6164440.

[5] Agarwal, S H. Koppula and KP. Leela, et al. "URL normalization for de-duplication of web pages", in Proceedings of the 18th ACM conference on Information and knowledge management, Hong Kong, China,2009, pp.1987-1990.

[6] L.k. Soon and S.H. Lee. "Enhancing URL Normalization Using Metadata of Web Pages", 2008 International Conference on Computer and Electrical Engineering, Phuket, pp. 331-335. 2008 doi: https://doi.org/10.1109/ICCEE.2008.112.

[7] Z .B Yossef , I. Keidar and U. Schonfeld,U "Do Not Crawl in the DUST: Different URLs with Similar Text", in proceedings of the 16th international conference on World Wide Web (WWW '07), ACM, New York, NY, USA ,2007, pp.111-120 doi: http://dx.doi.org/10.1145/1462148.1462151.

[8] Shestakov. "Current challenges in web crawling", in, Lecture notes on computer science, F. Daniel, P. Dolog, Li Q Ed. Springer-Verlag: Berlin, Heidelberg, 2013, pp. 518-521.

[9] T. Lei, R. Cai and J.M.Yang, et al. "A pattern tree-based approach to learning URL normalization rules", in Proceedings of the 19th international conference on World Wide Web (WWW'10), New York, NY, USA, 2010, pp. 611-620. doi: http://dx.doi.org/10.1145/1772690.1772753.

[10] A.Dasgupta, R. Kumar and A. Sasturkar. "De-duping URLs via rewrite rules", in Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '08), ACM, New York, NY, USA, 2008, pp. 186-194. doi: https://doi.org/10.1145/1401890.1401917.

[11] H. Lu, D. Zhan and L. Zhou, et al. "An Improved Focused Crawler: Using Web Page Classification and Link Priority Evaluation", Mathematical Problems in Engineering, vol. 2016, 2016 doi:10.1155/2016/6406901.

[12] H. W. Hao, C. X. Mu and X. C. Yin,et al. "An improved topic relevance algorithm for focused crawling", IEEE International Conference on Systems, Man, and Cybernetics, Anchorage, AK, 2011, pp. 850-855. doi: 10.1109/ICSMC.2011.6083759.

[13] B. Yohanes, P. Handoko, H K. Wardana. "Focused Crawler Optimization Using Genetic Algorithm". Telkomnika, 2011, doi: 10.12928/telkomnika.v9i3.730.
[14] S. Batsakis, G.M. E. Petrakis and E. Milios "Improving the performance of focused web crawlers", Data & Knowledge Engineering, 2009, Vol. 68, no.10, pp 1001-1003.ISSN 0169-023X, https://doi.org/10.1016/j.datak.2009.04.002.
[15] S. Khalil and M. Fakir "R Crawler: An R package for parallel web crawling and scraping" , Software, vol. 6, 2017, pp. 98-106, ISSN 2352-7110, https://doi.org/10.1016/j.softx.2017.04.004.
[16] X. Zhang and M. Xian. "Optimization of Distributed Crawler under Hadoop". MATEC Web of Conferences, 2015. doi: 10.1051/matecconf/20152202029.
[17] Cao F, Jiang D ,Singh J P. "Scheduling Web Crawl for Better Performance and Quality",2003.[Online].Available: ftp://ftp.cs.princeton.edu/reports/2003/682.pdf. Accessed Jan 29, 2018.
[18] K. Rodrigues, M. Cristo and E. S. de Moura et al. "Removing DUST Using Multiple Alignment of Sequences," in IEEE Transactions on Knowledge and Data Engineering, vol. 27, no. 8, pp. 2261-2274, Aug 2015.doi: 10.1109/TKDE.2015.2407354.
[19] Purnamasari, L.Y. Banowosari, R.D. Kusumawati, et al. 2017. "Semantic Similarity for Search Engine Enhancement". Journal of Engineering and Applied Sciences , Vol 12. 2017, pp. 1979-1982.