International Journal of Engineering Technology Science and Research
IJETSR
www.ijetsr.com
ISSN 2394 – 3386
Volume 4, Issue 7
July 2017

Distributed Crawler for Detection and Removal of DUST using DUSTER

Jyoti G. Langhi
Department of Computer Engineering
Marathwada Mitra Mandal's
College of Engineering, Pune-52

Prof Shailaja Jadhav
Department of Computer Engineering
Marathwada Mitra Mandal's
College of Engineering, Pune-52

Abstract—WWW is commonly used medium to search infor-mation using Web crawlers. Web crawler collects many pages but some of them contains duplicate content.Different URLs with Similar Text are generally known as DUST. To improve the performance of search engines, a new method called DUSTER is used. DUSTER detects and removes duplicate URLs. It does not fetch their contents. The normalization rules are used which converts all duplicate URLs into the same canonical form. In DUSTER system crawler is used while in the proposed system we intend to use distributed crawler to improve the scalability and speed. Single crawler crawls single URL at a time while multiple URLs can be crawled using distributed crawler at the same time. DUSTER converts all the URLs into multiple sequence of alignments which generates candidate rules and rules of validation.Then the candidate rules filtered out according to their performance in a validation set and finally removes the duplicate URLs. Using this method reduction of large number of duplicate URLs is achieved.

Index Terms— Crawling, Dup-cluster, DUSTER, URL Normal-ization.

I. INTRODUCTION

A Web crawler fetches data from various servers. Gathering data from various sources around the world takes huge amount of time. Such a single process faces problems on the process-ing power of a single machine and one network connection. If the workload of crawling Web pages is distributed , the job can be performed faster. Many search engines run multiple processes in parallel.

On the web there are different URLs that fetches the same page. These similar URLs are known as DUST. Duplicate URLs occur because of many reasons. DUST detection is important task for search engine because Crawling these duplicate URLs is a waste of resources. This results in the poor user experience. The existing system focused on document content to remove Duplicate URLs. Generation of Dynamic web pages leads to Duplication of contents. DUSTER converts duplicate URLs into same canonical form which can be used by web crawlers to avoid DUST. Instead of processing all URLs the existing system uses random sampling.In DUSTER framework, multiple sequence alignment is used to obtain a general and smaller set of rules and to avoid duplicate URLs. Multiple sequence alignment can be used to identify similar strings, so that normalization rules can be derived. More general rules can be generated using multiple sequence alignment algorithm to remove the duplicate URLs with similar text.

To fetch the URLs from the web a crawler is used in an existing system.More than one crawler can be used in distributed web crawling. Each crawler in a system acts as seperate entity and does its own indexing. Distributed system can process a growing workload as we distribute the resources in the system.Data fetched by single crawler go by single physical link. If crawling process is distributed in several processes then it makes easy to build scalable system[3].

II. LITERATURE SURVEY

DUST can be detected using two methods. First method is content based method and another one is URL based method. Content based method fetches the whole paged and full content is inspected by comparing it using syntactic or semantic evidence. In URL based method, without examining the content of the page the duplicate URLs can be find out In the following paragraphs some URL-based methods are focused on.

International Journal of Engineering Technology Science and Research
IJETSR
www.ijetsr.com
ISSN 2394 – 3386
Volume 4, Issue 7
July 2017

In base paper [2] the DUSTER framework is proposed. DUSTER detects duplicate URLs and removes them.This method uses normalization rules that converts distinct URLs which refer to same content to a common canonical form. Normalization rules are generated to convert all duplicate URLs into same canonical form. This makes easy to detect them. The scalability and precision can be improved using other data sets.

S. Bal and G. Geetha [3] proposed a smart distributed web crawler. In this paper the authors suggested that use of distributed cralwer is faster than that of single crawler. Distributed crawler is used to improve the scalability.

The work done by A. Agarwal and other authors [4] focuses that the basic and deep tokenization of URLs to extract all possible tokens from URLs which are mined by Rule generation techniques proposed by them for generating normalization Rules. Proposed system implements for giving output to the user efficiently and large-scale de-duplication of documents. Short Web pages does not work well and does not find out noise ratio on web pages. A new technique SizeSpotSigs [5] is used for effective near duplicate detection algorithm considering the size of page content in mining. Proposed system implements noise-content ratio to work better. The disadvantage of this technique is that the size of the core content of Web page does not automatically or approximately decided.

T. Lei, R. Cai and other authors [6] proposes top-down approach. In this paper, a new technique pattern tree based approach is used to learn URL normalization rules. In this training data set is created first. Then a pattern tree is gen-erated on basis of training data set. After that the duplicate nodes are identified from the pattern tree. And at last, the normalization rules are generated. As these normalization rules are directly applied on pattern rather than on every URL pair, the respective computational cost is low. The proposed system is help to user select deployable rules by removing conflicts and redundancies.

The list of retrieved document contains duplicated and near duplicate results. B. S. Alsulami and other authors [7] have done a survey on Near Duplicate Document Detection. The detection of Near Duplicate Document is the problem of finding all documents fast whose similarities are equal to or greater than the threshold which is given. There are two techniques: Near duplicate prevention and Near duplicate detection. The proposed system is used in to Technical support doc management, Plagiarism Detection, Web Crawling, Digital libraries and electronic publishing, Database cleaning, Files in a file system, E-mails applications.

The first URL-based method is DustBuster [8]. This tech-nique is used to avoid duplicate urls in crawling. A dynamic forum site, an academic site, a large news site, a small news site, the proposed system DustBuster mines dust effectively from URL List. It can reduce crawling overhead, increases crawl efficiency and reduces indexing overhead.

The authors in [9] presented machine learning technique to generalize the set of rules that reduces resource footprint to be usable at web scale. The basic and deep tokenization of URLs to ex-tract all possible tokens from URLs which are mined by their Rule generation techniques for generating normalization rules. The proposed system is used to measure the performance of URL dataset on key metrics.

Author Dasgupta [10] presented a new characterization of URL Rewrite rules because substitution rules were not cap-turing many duplicate URL transformations on web. Rewrite rules applied to remove duplicates. Here, the set of URLs are classified into classes. Clusters are formed according to classes of similar content then rewrite rules applied on the clusters. It improves the efficiency of entire process.

III. MATHEMATICAL MODEL

Mathematical model for the DUST removal system can be given as follows:

Let the system be S; The system S be a set of 6 tuples.

$$S = \{I, O, f_1, f_2, f_3, f_4\} \qquad (1)$$

Where,

I = Input, set of duplicate URLs.

It can be represented as set of ’n’ duplicate URLs

$$\{u_1, \ldots, u_n\} \qquad (2)$$

O = Output, set of URLs removing duplicate URLs.

It can be represented as set of m URLs, removing duplicate URLs

$$\{u_1, \ldots, u_m\} \qquad (3)$$

f_1 = It is a Multiple URL Alignment function Input given is Dup-cluster C

It gives Output tuple as (consensus, domains, support)

f_2 = Function for Candidate Rule Generation

Input given is: Training set T and S n duplicate clusters set It gives set of m Candidate Rules represented as CR = $r_1, \ldots, r_m$

f_3 = Function for URL Normalization

f_4 = Function for generating Validate Rules

Input given, VS- validation set, CR- n candidate rules set, f pr_{max}- maximum false positive rate , min_{supp}- minimum number of instances that are required.

It gives Output as the set of n valid rules denoted by VR.

Given X and Y are the sequences with m and n characters respectively. To measure the distance between the URL token sets a score function sf is defined. The scoring function, given by Equation , is the Jaccard similarity coefficient[2] which is commonly used to measure the overlap between two sets.

B. System Architecture

We have used existing DUSTER architecture for removing the duplicate urls of similar text. This system uses multiple sequence alignment to get a general and smaller set of nor-malization rules. Its complexity is proportional to the number of URLs to be aligned. DUSTER uses single crawler to fetch the urls. In the proposed system we will be using distributed crawlers instead of single crawler. This increases the speed of search and ultimately the scalability will also be increased.

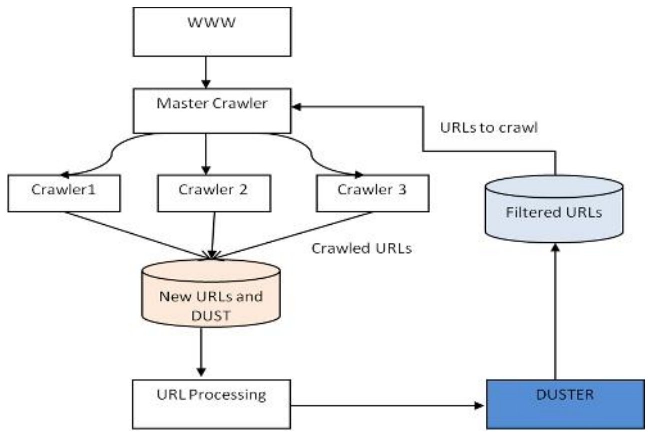


Fig. 1. Architecture of DUSTER with Distributed Crawler)

$$sf(X_i; Y_j) = \frac{j x_i^T Y_j j}{x_i Y_j} \quad (4)$$

if S

$$9(x_i; y_j)\ 2\ X_i \quad Y_j\ j\ (x_i) = (y_j) \quad (5)$$

-1 otherwise

IV. SYSTEM ARCHITECTURE / SYSTEM OVERVIEW

A. Hardware and Software Requirements

Hardware Requirements

Processor : Pentium IV (and onwards).

Memory (RAM) : 1 GB RAM.

Hard disk : 40GB

Software Requirements

Operating System : Microsoft operating system (Win-dows 7)

Coding Language : Java

Tool : Eclipse(Indigo/Luna)

Database : MySQL

less susceptible to noise[2].

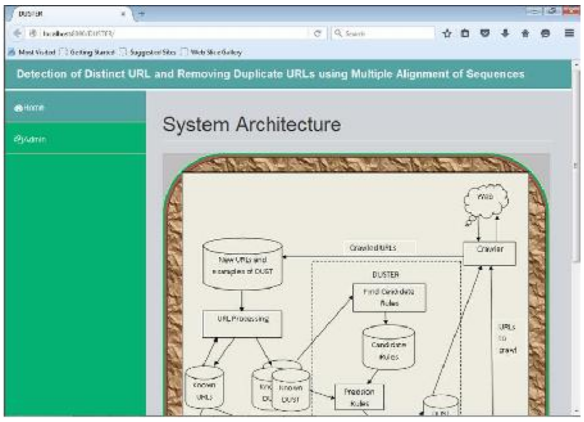
Phases of DUSTER- The proposed method DUSTER is divided in two main steps as mentioned below, Step 1: Generation Candidate rules : In this step, The multi-sequence alignment algorithm is applied first in dup-clusters to align all the URLs and obtains consensus sequences for each dup-cluster. Then the candidate rules are generated from these consensus sequences. A heuristic is used to ensure the efficiency of the method for large clusters [1].

Step 2: Validating candidate rules: In this phase according to performance the candidate rules get filtered out in a validation set[1].

V. IMPLEMENTATION STATUS

We have created an application DUSTER with Distributed Crawlers. When we run this application the front page gets displayed. On the front page we have displayed the system architecture of DUSTER. Fig.2 displays the front page of an application.

International Journal of Engineering Technology Science and Research
IJETSR
www.ijetsr.com
ISSN 2394 – 3386
Volume 4, Issue 7
July 2017


Detection of Distinct URL and Removing Duplicate URLs using Multiple Alignment of Sequences

Home
Admin

System Architecture

Fig. 2. Front Page of DUSTER application

In this architectural diagram, first the URLs are crawled by distributed crawlers then they are merged with the set of already known URLs. It forms a new set of known URLs. During crawling, by following canonical tags, the crawlers can identify some examples of DUST. Finally, a new set of known DUST is formed. We have divided the set of known DUST into multiple sets. Final set of known DUST is taken by DUSTER and use it to find candidate rules and validate rules, by splitting it into training sets and validating sets. The final rules are then used to normalize the known URLs which yields a new (and reduced) set of URLs to be crawled. This reduced set of URLs and DUST rules are passed to the crawler finally.

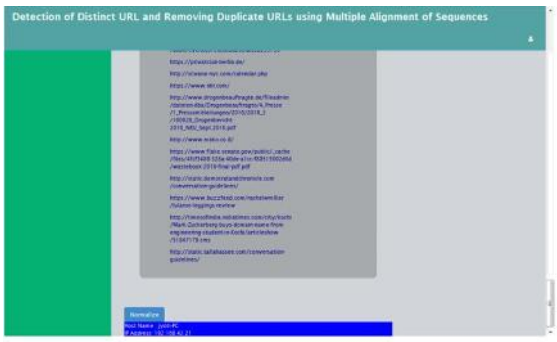
Master crawler assigns the URLs amongst different crawlers. For assigning new URL static assignment or dynamic assignment strategies can be used. In this way server balances the load of crawlers.

1) Multiple Sequence Alignment- To pick out similarities and differences among strings/sequences Multiple Sequence Alignment is used. Similarities and differences can be used to choose fixed and variable substrings in the URLs which helps to derive rules of normalization. The multiple sequence alignment methods find patterns in all the available strings. So this method can find more general rules and avoids problems related to pair-wise rule generation and finding rules across sites. Thus, a full multi-sequence alignment of duplicate URLs can make the learning process more robust and

At left two options are kept, Home and Admin. Admin have all the rights to add, modify or delete any content in the application. We will be providing User Login also so that user can search for the URL and they can get the filtered URLs only. When Admin logins into he can view a search bar where he can put his keyword or URL to be searched. Then all the URLs linked to that keyword get displayed. This is as shown in fig.3

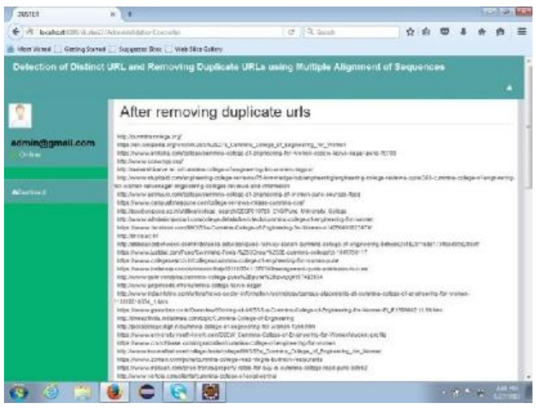
After clicking on the button the Duplicate Clusters get formed according to their similarity. Then the Candi-date Rules are generated for clusters. Which retrieves do-main,transformation and sequences. After this the Rules are formed for every cluster. Normalization is done on this. Duplicate URLs are removed out and the set of original URLs is left out.

After this the known URLs and known DUST are seperate out. Single crawler is used here. The distributed crawler is being used to improve the scalability.


Detection of Distinct URL and Removing Duplicate URLs using Multiple Alignment of Sequences

Normalize

Fig. 3. Crawling Results


Detection of Distinct URL and Removing Duplicate URLs using Multiple Alignment of Sequences

admin@gmail.com

After removing duplicate urls

Fig. 4. Original URLs

VI. SYSTEM ANALYSIS

The dataset used to analyze the system is Real time set of URLs. This data set contains around 100 web pages crawled from the google search engine. No restrictions were there regarding content duplication or quality. The DUSTER[2] method is used for the comparison. The URLs crawled by distributed crawlers are more than that of crawled by single URL. So our method improves the scalability. The time required to remove DUST for our method is comparively lesser than that of DUSTER method.

Fig. 5 shows the proposed system performance ratio after removing duplicate URLs in cluster-wise ratio. The X axis represents different duplicate clusters whereas Y axis denotes the time required to remove the DUST. This clearly shows that our method requires less time means the proposed method is faster than that of the existing system.

VII. EXPECTED OUTCOME

The system DUSTER detects the duplicate URLs with similar text (DUST) and removes duplicate URLs from the dup-cluster using distributed crawlers. The scalability will be improved also the search becomes faster. The precision will be improved as we will be applying precision rules.

Efficiency of the system will depend upon multiple sequence alignment algorithms. These algorithms provide high quality. The performance is superior of the DUSTER system which uses distributed crawlers. The performance is better when DUSTER generates smallest number of candidate rules and highest rate of valid rules.

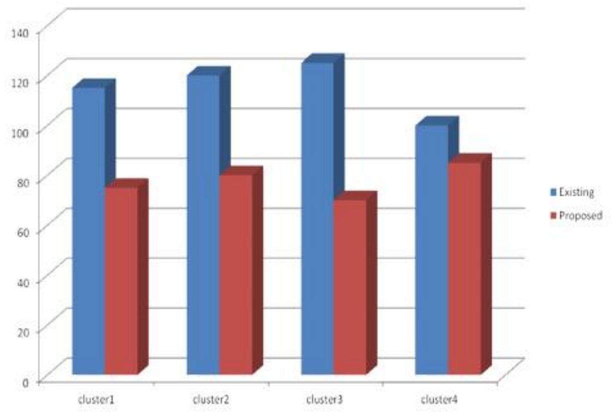


Fig. 5. Proposed system performance ratio

VIII. CONCLUSION

In this Project, we make an attempt to use DUSTER method with distributed crawler to address the DUST prob-lem faster. DUSTER learns the rules of normalization for converting distinct URLs which was easily detected. The proposed system is faster than the existing and scalability is also improved.Clusters may include false positives due to the approximate similarity measures.

ACKNOWLEDGMENT

We express our deepest gratitude to the college authorities for technical guidance and Infrastructure. Lastly we wish to thank the researchers and reviewers for their contributions because of which we could complete this work.

REFERENCES

[1] Jyoti G. Langhi,Prof. Shailaja Jadhav,"Detection and Removal of DUST: Duplicate URLs with Similar Text Using DUSTER",International Journal of Innovative Research in Computer and Communication Engineer-ing,Vol. 5, Issue 1, January 2017

[2] Kaio Rodrigues, Marco Cristo, Edleno S. de Moura, and Altigran da Silva, "Removing DUST Using Multiple Alignment of Sequences.", IEEE Transactions On Knowledge and Data Engineering,VOL. 27, NO. 8, AUGUST 2015

[3] Swaroop Kaur Bal,G.Geetha,"Smart distributed web crawler",ICICES 2016

[4] H. S. Koppula, K. P. Leela, A. Agarwal, K. P. Chitrapura, S. Garg, and A. Sasturkar, Learning url patterns for webpage deduplication, in Proc. 3rd ACM Int. Conf. Web Search Data Mining, 2010, pp. 381390.

[5] X. Mao, X. Liu, N. Di, X. Li, and H. Yan, "Sizespotsigs: An effective deduplicate algorithm considering the size of page content",in Proc. 15th Pacific-Asia Conf. Adv. Knowl. Discovery Data Mining, 2011

[6] T. Lei, R. Cai, J.-M. Yang, Y. Ke, X. Fan, and L. Zhang, "A pattern tree-based approach to learning url normalization rules", 19th Int. Conf. World Wide Web, 2010

[7] B. S. Alsulami, M. F. Abulkhair, and F. E. Eassa, "Near duplicate document detection survey", Int. J. Comput. Sci. Commun. Netw. vol. 2, no. 2, pp. 147151, 2012

[8] Z. Bar-Yossef, I. Keidar, and U. Schonfeld, "Do not crawl in the dust: Different urls with similar text", ACM Trans. Web, vol. 3, no. 1, pp. 3:13:31, Jan. 2009.

[9] A. Agarwal, H. S. Koppula, K. P. Leela, K. P. Chitrapura, S. Garg, P. Kumar GM, C. Haty, A. Roy, and A. Sasturkar, "Url normalization for de-duplication of web pages", ACM Conf. Inf. knowl. Manage, 2009

[10] A. Dasgupta, R. Kumar, and A. Sasturkar, "De-duping urls via rewrite rules", ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2008, pp. 186194.