# Web Crawling DUST Removing using PUK (Precise Unique Key) Technique

S. ISWARYA, M.E.,
*Department of Computer Science and Engineering*
*A.V.C College of Engineering*
*Mannampandal-609305.*

Mr. K. TAMILSEVAN, M.Tech.,
*Department of Computer Science and Engineering*
*A.V.C College of Engineering*
*Mannampandal-609305.*

***Abstract*** **- A large number of URLs collected by web crawlers correspond to pages with duplicate or near-duplicate contents. To crawl, store,and use such duplicated data implies a waste of resources, the building of low quality rankings, and poor user experiences. To deal with thisproblem, several studies have been proposed to detect and remove duplicate documents without fetching their contents. To accomplish this,the proposed methods learn normalization rules to transform all duplicate URLs into the same canonical form. A challenging aspect of thisstrategy is deriving a set of general and precise rules. In this proposed model PUK (Precise Unique Key) DUST removing technique, a new approach to derive quality rules that takeadvantage of amulti-sequence alignment strategy. It demonstrates that a fullmulti-sequence alignment of URLs with duplicated content,before the generation of the rules, can lead to the deployment of very effective rules. By evaluating this method, it observed it achieved largerreductions in the number of duplicate URLs than our best baseline, with gains of 85 and 150.76 percent in two different web collections002E**

***Index Terms - Web technology, web crawling and normalization rules.***

## I. INTRODUCTION

Knowledge about dust rules can be valuable for search engines for additional reasons: dust rules allow for a canonical URL representation, thereby reducing overhead in crawling, indexing, and caching and increasing the accuracy of page metrics, like PageRank. For example, in one crawl we examined, the number of URLs fetched would have been reduced by . We focus on URLs with similar contents rather than identical ones, since different versions of the same document are not always identical; they tend to differ in insignificant ways, e.g., counters, dates, and advertisements. Likewise, some URL parameters impact only the way pages are displayed (fonts, image sizes, etc.) without altering their contents. Contrary to initial intuition, we show that it is possible to discover likely dust rules without fetching a single web page. We present an algorithm, Dust Buster, which discovers such likely rules from a list of URLs. Such a URL list can be obtained from many sources including a previous crawl or web server logs.The rules are then verified (or refuted) by sampling a small number of actual web pages. Information on the Web is very huge in size. 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 the information. Where, among users looking for information on the Web submit information requests to various Internet search engines. Search engines are critically important to help users find relevant information on the Web. Search engines in response to a user's query typically produces the list of documents ranked according to closest to the user's request. These documents are presented to the user for examination and evaluation. Web users have to go through the long list and inspect the titles, and snippets sequentially to recognize the required results. Filtering the search engines' results consumes the users' effort and time especially when a lot of near duplicate.

## II. RELATED WORK

Presence of duplicate documents in the World Wide Web ad-aversely abets crawling, indexing and relevance, which are The core building blocks of web search. In this paper, we Present a set of techniques to mine rules from URLs and Utilize these learnt rules for de-duplication using just URL Strings without fetching the content explicitly. Our tech-nique is composed of mining the crawl logs and utilizing clusters of similar pages to extract specific rules from URLs belonging to each cluster. Preserving each mined rules for de-duplication is not efficient due to the large number of specific rules. We present a machine learning technique to generalize the set of rules, which reduces the resource foot-print to be usable at web-scale. The rule extraction tech-Inquest is robust against web-site specific URL conventions. We demonstrate the electiveness of our techniques through Experimental evaluation.

We focus on URLs with similar contents rather than identical Ones, since different versions of the same document are Not always identical; they tend to differ in insignificant ways,e.g., counters, dates, and advertisements. Likewise, some URL parameters impact only the way pages are displayed(Fonts, image sizes, etc.) without altering their contents. Detecting DUST from a URL list. Contrary to initial intuition, we show that it is possible to discover likely dust rules without fetching a single web page. We present an algorithm, Dust Buster, which discovers such likely rules from a list of URLs. Such a URL list can be obtained from many sources including a previous crawl or web server logs.1 The rules are then verified (or refuted) by sampling a small number of actual web pages. At first glance, it is not clear that a URL list can provide reliable information regarding dust, as it does not include actual page contents.

All Rights Reserved @ 2016 IJARMATE

The primary goal of the Terabyte Track is to develop an evaluation methodology for terabyte-scaledocument collections. In addition, we are interested in efficiency and scalability issues, which can be studied more easily in the context of a larger collection.TREC 2006 is the third year for the track. The track was introduced as part of TREC 2004,with a single adhoc retrieval task. For TREC 2005, the track was expanded with two optional tasks:a named page finding task and an efficiency task. These three tasks were continued in 2006, with 20groups submitting runs to the adhoc retrieval task, 11 groups submitting runs to the named page finding task, and 8 groups submitting runs to the efficiency task. This report provides an overview of each task, summarizes the results, and outlines directions for the future. Further background information on the development of the track can be found in the 2004 and 2005 track reports [4, 5]..

Presence of duplicate documents in the World Wide Web adversely affects crawling, indexing and relevance, which are the core building blocks of web search. In this paper, we present a set of techniques to mine rules from URLs and utilize these rules for de-duplication using just URL strings without fetching the content explicitly. Our technique is composed of mining the crawl logs and utilizing clusters of similar pages to extract transformation rules, which are used to normalize URLs belonging to each cluster. Preserving each mined rule for de-duplication is not efficient due to the large number of such rules. We present a machine learning technique to generalize the set of rules, which reduces the resource footprint to be usable at web-scale. The
rule extraction techniques are robust against web-site specific URL conventions. We compare the precision and scalability of our approach with recent efforts in using URLs for de-duplication. Experimental results demonstrate that our approach achieves 2 times more reduction in duplicates with only half the rules compared to the most recent previous approach. Scalability of the framework is demonstrated by performing a large scale evaluation on a set of 3 Billion URLs, implemented using the MapReduce framework.

Information on the Web is very huge in size. 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 the information. Where, among users looking for information on the Web, 85% submit information requests to various Internet search engines. Search engines are critically important to help users find relevant information on the Web. Search engines in response to a user's query typically produces the list of documents ranked according to closest to the user's request. These documents are presented to the user for examination and evaluation. Web users have to go through the long list and inspect the titles, and snippets sequentially to recognize the required results. Filtering the search engines' results consumes the users' effort and time especially when a lot of near duplicate.

The efficient identification of near duplicates is an important in a many applications especially at that has a large amount of data and the necessity to save data from diverse sources and needs to be addressed. Though near duplicate documents display striking similarities, they are not bit wise similar. Web search engines considerable problems due to duplicate and near duplicate web pages. These pages increase the space required to store the index, either decelerate or amplify the cost of serving results and so exasperate users. Thus algorithms for recognition of these pages become inevitable.

## III. PROPOSED SYSTEM

In this proposed technique **PUK** (Precise Unique Key)for getting the maximum results from high dimensional database systems in the different search engines database clusters. The first step of this algorithm is to get the maximum url pairwise category partition, which indicates high correlations among the clicked urls in a session in user click through logs, and combine it with the clicked images visual information for inferring user image-search goals.

Then, this technique look into the geographical location of the current session users and maintain the different set of url click logs with tokensfor the various countries search for example if a person searching from the UK means it shows the high density point verification of geographical user search log and shows the results.

The secondary part of this technique high density arbitrary data process the clustering technique begins with an initial set of high weighted url edges with redundant token numbersand iteratively refines this set so as to decrease the sum of squared errors. PUK based multiple dust removing is quite sensitive to the initial selection of exemplars, so it is usually rerun many times with different initializations in an attempt to find a good solution.

This analysis also perform proposed feedback session consists of both clicked and unclicked URLs and ends with the last URL that was clicked in a single session. It suggests that so-called “navigational” searches are less prevalent than seeking” goal may account for a large fraction of web searches. It illustrate how this knowledge of user search goals might be used to improve future web search engines.

- URL Web data Category Metrics
- URL Pair wise Tokenization
- Multiple Sequence Alignment of Pair wise DUST
- DUST Removal using PUK

### URL Web data Category Metrics

In this module registration of new user who wants to access the service takes place. When a user wants to create

All Rights Reserved @ 2016 IJARMATE

account they can make use of this module. It involves collection of details such as their name, address, phone number, date of birth, city and the user is also asked to choose the website .On the end of the phase user will be given user Id, password and website. All the details will be stored in to database. Then the user view the log of details from the database.

**URL Pair wise Tokenization**

In this module, the user will be asked to give his user websites .When the user choosing the website in a database for getting user desired information .The database show the log of details and a server pairing that log of details.Then show a paired details to user.

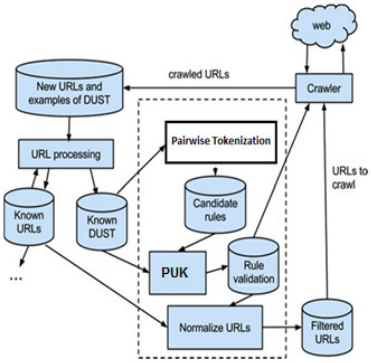
**Multiple Sequence Alignment of Pair wise DUST**

The user after giving his credentials, he can able to do use the services provided by the websites. He can exit the application by clicking exit button. Thus by enhancing the security features, the services can be used by the valid persons and the integrity of the system is thus maintained.A user choosing particular website from the database

**DUST Removal using PUK**

In this module, removing DUST from a user websites using PUK.The user after giving his credentials, he can able to do use the services. Server using a PUK algorithm for removing dust in a log of websites. Thus by enhancing the security features, the services can be used by the valid persons and the integrity of the system is thus maintained. The system is implemented with session validations. So no one is given the access to move direct to the page without login.

**Architecture**




## IV. CONCLUSION

Thus this system 'DUST REMOVEING MODULES' has solved all the problems existed in previous systems. Since this system has log of websites then pairing the websites. Thus the user feels free to use the websites and he can be sure that his credentials have been protected. As the system gives the opportunity to change his websites.User view the original websites.The system is simple and user-friendly and they can avail the services easily.

## V. FUTURE WORK

In this project has developed to store the log details into the server without any duplication but the timings of the server can take the more time to do the every operation in the database so in future work we have to reduce the loading timings and efficient in the serverdb as well as we have to take the server log in country wise also because it is global server log maintain we follow this algorithm in the every country server.

## VI. RESULT AND DISCUSSION

We use two document collections in our experiments: GOV2. Dataset consists of a snapshot of the resources fetched from 25,205,179 individual documents from US government domains in 2014. According to the TREC track information some duplicate documents have already been removed from GOV2. The GOV2 TREC datasetcontains about 3.42 million duplicate URLs divided into about 1.43 million dup-clusters. These documents wereGrouped by creating a small fingerprint of their content and hashing the URLs with identical fingerprints into the same clusters is a collection of over 150 million web pages crawled from the Brazilian domain using an actualBrazilian crawling system. This crawling was performed from September to October, 2014, with no restrictionsregarding content duplication or quality. To identify groups of duplicate URLs in WBR10, we adopted the sameapproach used by the authors in [11]. Thus, we scanned the collection to find out the web sites which explicitly indicatethe canonical URLs in their pages. By doing this, we identified about 3.95 million duplicate documents in fora total of about 1.14 million dup-clusters. Although is six times larger than GOV2, it has only 15 percent moreDUST identified. This was expected since webmasters are not obliged to identify canonical URLs.

Existing Method

| Data Set | Method | Candidates | Valid | Rate |
|---|---|---|---|---|
| GOV2 | R(Fanout -10) | 7097 | 2242 | 31.6% |
| | R(tree) | 2458 | 718 | 29.21% |
| | Duster | 1685 | 1332 | 79.05 |

Our Proposed Method.

| Data Set | Method | Candidates | Valid | Rate |
|---|---|---|---|---|
| GOV2 | R(Fanout -10) | 7097 | 3482 | 47.6 % |

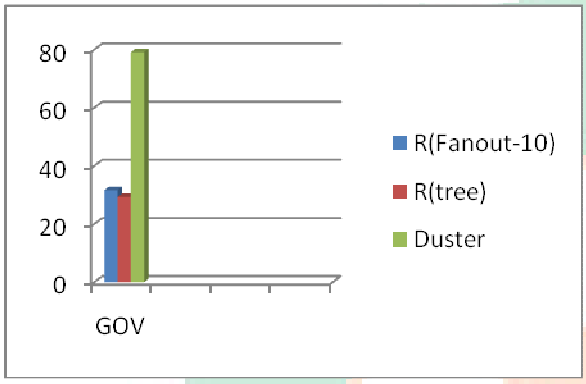
All Rights Reserved @ 2016 IJARMATE

| | | | | |
|---|---|---|---|---|
| | **R(tree)** | **2458** | **988** | **39 %** |

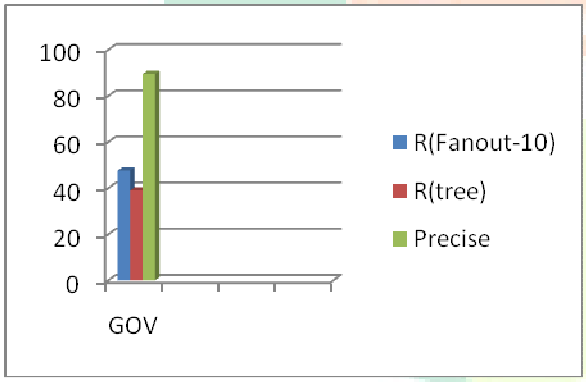
Everything will be consider in our project in our users need.

**Graph**

In our earlier system works like




In our project has work like that:




These two methods were chosen due to their performance in previous experiments, which indicate they represent thebest options found in literature for de-duplicating URLs.

## REFERENCES


[1] S. Abiteboul, M. Preda, and G. Cobena.Adaptive on-line page importance computation. In WWW '03:Proceedings of the 12th international conference onWorld Wide Web, pages 280{290, May 2003.

[2] 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 internationalconference on World Wide Web, pages 111{120, May 2007.

[3] T. Berners-Lee, L. Masinter, and M. McCahill. Uniform resource locators (url), 2012.

[4] Z. Bar-Yossef, I. Keidar, and U. Schonfeld. Do not crawl in the DUST: different URLs with similar text. TechnicalReport CCIT Report #601, Dept. Electrical Engineering, Technion, 2006.

[5] K. Bharat and A. Z. Broder. Mirror, Mirror on the Web: A Study of Host Pairs with Replicated Content. Computer Networks, 31(11–16):1579–1590, 2009.

[6] Theobald, M., Siddharth, J., and Paepcke, A. 2008. Spotsigs: robust and efficient near duplicate detection in large web collections. In SIGIR. 563–570

[7] Krishnamurthy KoduvayurViswanathan and Tim Finin, Text Based Similarity Metrics and Delta for Semantic Web Graphs, pp: 17-20, 2010

[8] JunpingQiu and QianZeng, Detection and Optimized Disposal of NearDuplicate Pages, 2nd International Conference on Future Computer and Communication, Vol.2, pp: 604-607, 2010.

[9] E. Uyar. Near-duplicate News Detection Using Named Entities. M.S. Thesis, Department of Computer Engineering, Bilkent University, Ankara, Turkey, 2009.

[10] Ranjna Gupta et. al. Query Based Duplicate Data Detection on WWW (IJCSE) International Journal on Computer Science and Engineering Vol. 02, No. 04, 2010, 1395-1400


All Rights Reserved @ 2016 IJARMATE