Abstract—Uniform Resource Locators (URL) were collected by web crawlers corresponding to pages with duplicate or near-duplicate contents. Crawl, store and use such duplicated data imply a waste of resources, the building of low quality rankings and poor user experiences. To manage this issue, a few investigations have been proposed to add and remove copy archives without getting their contents. To accomplish this proposed methods learn normalization rules to transform all duplicate URLs into the same canonical form. A testing part of this procedure determines an arrangement of general and exact rules. In this proposed model Precise Unique Key (PUK) DUST removing technique, a new approach to derive quality rules that take advantage of a multi-sequence alignment strategy, has been made. It demonstrates a full multi-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 was observed that it achieved larger reductions in the number of duplicate URLs than our best baseline, with gains of 85 and 150.76 per cent in two different web collection 002E.

Keywords—web technology, web crawling and normalization rules.

I. INTRODUCTION

Knowledge about DUST rules is valuable for search engines as the DUST rules allow for a canonical URL representation, thereby reducing overhead in crawling, indexing, and caching and increasing the accuracy of page metrics, like Page Rank. For example, in one crawl the number of URLs fetched would have been reduced. 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 in which pages are displayed without altering their contents. Without fetching a single web page, it is possible to discover likely DUST rules. We present an algorithm, Dust Buster, which discovers such likely rules from a list of URLs. These URL list can be obtained from many sources including a previous crawl or web server logs [1]. The rules are verified (or refuted) by sampling number of actual web page information on the web which is very huge in size. There is an essential to use this big volume of information efficiently for effectively satisfying the information need of the user on the Web. Search engines become the key development on the web for retrieving the information.

II. RELATED WORK

in World Wide Web the presence of duplicate documents adversely abets crawling, indexing and relevance, which is meant to be core building blocks of web search. In this paper, a set of techniques to mine rules from URLs are presented and utilize these learnt 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 specific rules from URLs belonging to each cluster [5]. Preserving each mined rules for de-duplication is not efficient due to the large number of specific rules. We of rules, which reduces the resource foot-print to be usable at web-scale. The rule extraction technique is robust against web-site specific URL conventions. We present a machine learning technique to generalize the set demonstrate the effectiveness of our techniques through experimental evaluation.

The main aim of the Terabyte track is to develop an evaluation methodology for terabyte-scale document collections. In addition, we are interested in efficiency and scalability issues, which can be studied more easily in the context of a larger collection [4]. 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 20 groups 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 data provides an outline of each task, summarizes the results, and outlines directions for the future. Further it is stated that background information on the development of the track could be found in the 2004 and 2005 track reports.

The efficient identification of near duplicates is important in many applications especially 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 have considerable problems due to duplicate and near duplicate web pages. These pages increase the space which is 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 [2].

III. PROPOSED SYSTEM

In this proposed technique PUK is used for getting the maximum results from high dimensional database systems from the different search engines database clusters [9]. The first step of this algorithm is to get the maximum URL pair wise category partition, which indicates high correlations among the clicked URL in a session in user click logs, and combine it with the clicked images visual information for inferring user image-search goals.
Then, this technique looks into the geographical location of the current session users and maintains the different set of URL click logs with tokens for 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 [6].

The secondary part of this technique is high density arbitrary data process. The clustering technique begins with an initial set of high weighted URL edges with redundant token numbers and iteratively refines this set so as to decrease the sum of squared errors [11]. PUK based multiple DUST removing is quite sensitive to the initial selection so it is usually rerun many times with different initializations in an attempt to find a good solution.

This analysis also performs 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 [12].

It illustrates how this knowledge of user search goals might be used to improve future web search engines.

A. URL Web data Category Metrics

Registration module allows new user who wants to access the service takes place. When a user wants to create account he/she can make use of this module. It involves collection of details such as his/her name, address, phone number, date of birth, city and the user is also asked to choose the website [10]. At the end of the phase user will be given user Id, password and website. All the details will be stored in to the database. Then the user views the log on details from the database

B. URL Pair-wise Tokenization

In this module, the user will be asked to give his user website. When the user chosen the website in a database server to receive user desired information by applying the query regarding the user request and the server responds with pairing log off details.

C. Multiple Sequence Alignment of Pair wise DUST

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

D. DUST Removal using PUK

In this module, removing DUST from a user websites using PUK, the user after giving his credentials is able to use the services. Server uses a PUK algorithm for removing DUST in a log on websites [7]. The system is implemented with session validations. So no one is given the access to move direct to the page without login.

E. Web Crawler Performance Report

The Admin can view all the reports of the user. The admin is having separate login as he holds the power of viewing all the details for a particular user and even for the entire user. The user can able to view details for the usage of services. User views the original website from the database.

Architecture is shown in figure 1. Generally users looking for information on the Web submit information requests to various Internet search engines [13]. Search engines are 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 the closest to the user’s request. These documents are given to the user for examination and evaluation. Web users must 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 user’s effort and time especially when a lot of near duplicate content [3].

IV. RESULT AND DISCUSSION

Two document collections are used in our experiments. GOV2 Dataset consists of a snapshot of the resources fetched from 25,205,179 individual documents from US government domains of 2014. Some duplicate documents have already been removed from GOV2. The GOV2 TREC dataset contains about 3.42 million duplicate URLs divided into about 1.43 million dup-clusters. These documents were grouped 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 actual Brazilian crawling system.

This crawling was executed from September to October, 2014, with no restrictions regarding content duplication or quality. To identify groups of duplicate URLs in WBR10, we adopted the same approach used by the authors in. Thus, we
scanned the collection to find out the web sites which explicitly indicate the canonical URLs in their pages. By doing this, we identified about 3.95 million duplicate documents in for a total of about 1.14 million dup-clusters. Although it is six times larger than GOV2, it has only 15 per cent more DUST identified. This was expected since web masters are not obliged to identify canonical URLs [8]. Results are shown in Table I and Table II and graphical representations of results are shown in Figure 2 and Figure 3.

TABLE I: RESULT OF EXISTING METHOD

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Method</th>
<th>Candidates</th>
<th>Valid</th>
<th>Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>GOV2</td>
<td>R(Fanout-10)</td>
<td>7097</td>
<td>2242</td>
<td>31.6%</td>
</tr>
<tr>
<td></td>
<td>R(tree)</td>
<td>2458</td>
<td>718</td>
<td>29.21%</td>
</tr>
<tr>
<td>Duster</td>
<td></td>
<td>1685</td>
<td>1332</td>
<td>79.05%</td>
</tr>
</tbody>
</table>

TABLE II: RESULT OF OUR PROPOSED METHOD

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Method</th>
<th>Candidates</th>
<th>Valid</th>
<th>Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>GOV2</td>
<td>R(Fanout-10)</td>
<td>7097</td>
<td>3482</td>
<td>47.6%</td>
</tr>
<tr>
<td></td>
<td>R(tree)</td>
<td>2458</td>
<td>988</td>
<td>39%</td>
</tr>
<tr>
<td>Duster</td>
<td></td>
<td>1685</td>
<td>2857</td>
<td>89.05%</td>
</tr>
</tbody>
</table>

Everything has been considered in the project to satisfy users need. These two methods have been chosen due to their performance in previous experiments, which indicates that they represent the best options found in the literature for de-duplicating URLs.

V. CONCLUSION

Thus this system has solved all the problems existed in previous systems. Since this system has log of websites it is pairing the websites. Thus the user free to use the websites and he or she can be sure that his credentials have been protected. The user is capable of viewing the original websites. This system is very simple and user-friendly, and user can avail the services easily.

It has been attempted to store the log details into the server without any duplication but the server can take more time to do the operation in the database. Hence to reduce the loading timings and to enhance the efficiency of server database as well as we have to maintain country wise server log which is also called as global server log.

REFERENCES


