Scientific Journal Impact Factor (SJIF) : 4.364
International Journal of Modern Trends in Engineering and Research
ISSN (ONLINE) : 2349 - 9745 ISSN (PRINT) : 2393 - 8161

REMOVAL OF DUPLICATE URL USING MULTIPLE SEQUENCE ALIGNMENT METHOD

Shraddha Sarode[1] and B. S. Chordia[2]
[1]*P.G Student Computer Engineering, SSVPS BSD COE, Dhule*
[2]*Associate Professor Computer Engineering, SSVPS BSD COE, Dhule*

Abstract—Word wide web is the largest repository of information today so it has become a primary mean for generating and locating the information on the web. Search engine contains a large numbers of URL which corresponds to page with duplicate and near duplicate content. Such URLs are called as DUST. Due to this user is not satisfied by redundant URLs which leads to waste of resources such as bandwidth and disc storage, low quality ranking and poor experience .To overcome this problem a number of methods are used that find the duplicate documents without fetching their content. To perform this normalization rules are used to transform all duplicate URLs into the same canonical form. An important task is to derive more general and precise rules. A framework called DUSTER which derives rules by using multiple sequence alignment method is used which interns help to derive effective rules.

Keywords—Web Mining, Search engine, web crawling, duplicate detection, Uniform resource locator (URL).

I. INTRODUCTION

The existence of syntactically different URLs which are linking to the same web page is common on the web. These duplicate URL with similar text are known as DUST. The Occurrence of DUST in web is due to number of reasons such as webserver software often uses aliases and redirections which dynamically generate the same web page for various different URL request which in turn gives rise to de-duplication of web pages. These Duplicate URLs have created a serious problem to search engine from crawling indexing to result serving. Other common reasons for the existence of the duplicate content are use of parameters placed in distinct position in the URLs and the parameters that have no effect on the page content, as the seession_id, which is used to identify user connection. The detection of Duplicate URL is extremely important task because crawling such contents leads to wastage of resources such as Internet bandwidth and disk storage. It also creates disturbance in algorithms such as link analysis and poor user navigation [1].

To overcome such problem of DUST many authors have proposed methods for detecting and removing DUST. Initially some method focused on comparing the document to avoid duplicate which was again a resource consuming process. Recent studies proposed methods which inspect the URL without fetching their content respectively. These methods are known as URL based de-duping which mines the crawl log to identify duplicate URL with similar text. They perform normalization rules that convert a duplicate URL into canonical form. But constraints to these methods are they need processing of large number of URLs which are susceptible to noise.

Inspired by these methods DUSTER a new method is used which takes the advantage of multiple sequence alignment method. Multiple sequence alignment is used as a tool in molecular biology to find the similarity among the patterns. Multiple sequence alignment method helps to align the subsequences of the URL. Our work aims at removing the duplicate URL with similar text from websites using Multiple Sequence alignment method.

II. RELATED WORK

Many authors have proposed various techniques to remove the Duplicate URL from the web that is inspecting the URL without fetching their content.

Bar-Yossefet. et al.[2] proposed a algorithm called DUSTBUSTER which is used to detect DUST rules. This algorithm helps to detect DUST by finding the normalization rules that transform a given URL likely to have similar content. This technique was done using substring substitution method. For example if the last part of URL is "story_1259" it should be converted into valid rule as "story?id=1259" and "news.google.com" valid rules as"google.com/news"

Dust Buster mines dust effectively from previous crawl logs or web server logs, without examining the page details. But these substitution rules derived from algorithm were not able to capture many duplicate URL on the web.

In 2008 A. Dasguptaet.et al.[3] proposed a new method which was able to capture all previous substitution rules. In this method URL are divided into equivalence class.URL with same equivalence class have same content. These rewrite rules can then be applied to eliminate duplicates among URLs that are encountered for the first time during crawling, even without fetching their content .It helps for trapping duplicates much earlier in a search-engine workflow, which improve the efficiency of entire processing. The disadvantage of this method is that it was unable to capture many common duplicate URL.

H.S. Koppulaet.et al. in 2010 proposed [4] a technique to mine rules from URLs and utilize these rules for de-duplication using just URL strings without fetching the content explicitly. The technique is made of extracting the crawl logs and using clusters of related pages to mine detailed rules from URLs which belongs to each cluster. It presents deep and basic tokenization of URLs to mine all possible tokens from URLs which are extracted by rule generation techniques for generating normalization rules. Problem with this method was not publicly available and it was not described with enough detail because it uses a bottom up approach in which the normalization rules are learned by inducing local duplicate pairs to more general form.

In 2010 Lie et. al[5] rethought the problem of URL normalization from a global perspective and proposed a top down URL pattern tree (UPT) based approach, which is remarkably different from existing approaches.(UPT) is built from clusters of duplicate URLs for a targeted website. The pattern tree helps to leverage the statistical information from all the training samples to create the learning process further strong and reliable.Figure1 shows pair wise bottom up approach there are four different URL. The values of the token T3 and T5 are generalized to '*'. Unfortunately this approach does not work up to the mark as the pattern tree construction algorithm should be accelerated further as it contributes to bottleneck.

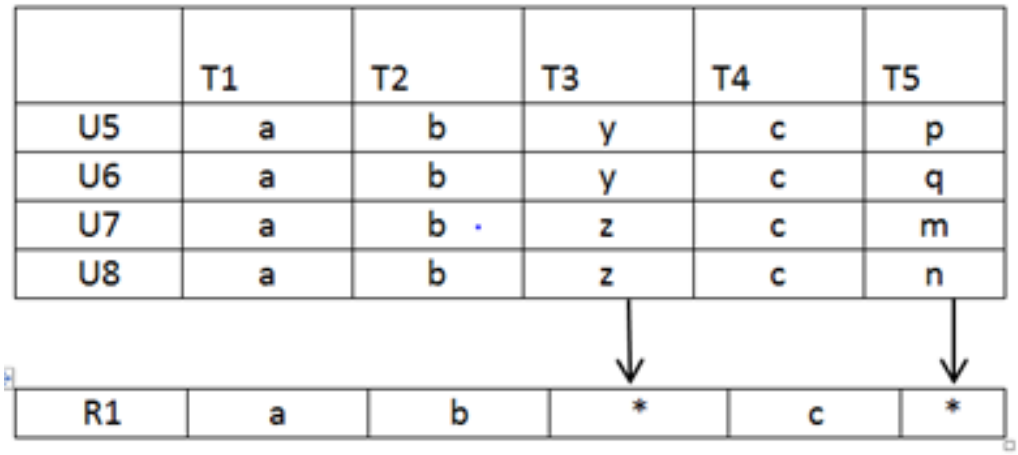

Figure 1. Pairwise Bottom-up Strategy [5]

Sérgio Anibal de Carvalhoet.et al.in 2005 proposed [13] a method called sequence alignment in 2005.This method gives detail description of how sequence alignment is done. Here sequences are compared to identify similarities and differences between them. Sequence alignment means the

relation between two strings. The characters in substring may be continuous while in subsequence may be non-continuous. For example “abc” is a subsequence but not a string of ‘axbxcx’.In general distance between the two sequence is amount of work done to convert one sequence into another.

The idea of aligning two sequences of possibly different size is to write one on top of the other, and break them into smaller pieces by inserting spaces in one or the other so that identical subsequences are eventually aligned in a one-to-one correspondence. For sequence alignment consider an example of two sequences A and B respectively, A=ACAAGACAGCGT and B=AGAACAAGGCGT. Alignment of both sequences is done as below:

A = ACAAGACAG-CGT
| || | | | | ||
B = AGAACA-AGGCGT

The main objective is to match the subsequence as far as possible. In the above example there are nine matches shown by vertical bar. However, if the sequences are not identical, mismatches are likely to occur as different letters are aligned together. Sequence alignment is a way of transforming one sequence into another.

III. SYSTEM ARCHITECTURE

Figure 2 shows the system architecture of the proposed system. In this system user has to provide a file which is obtained from crawl log or web server log. The file contains a set of URLs which are requested by the client. These URL are then processed and identified by the system whether they are duplicate or not. Duplicate URLs with similar text are grouped into cluster. These clusters of URLs are then processed by using URL processing technique which is discussed as below.

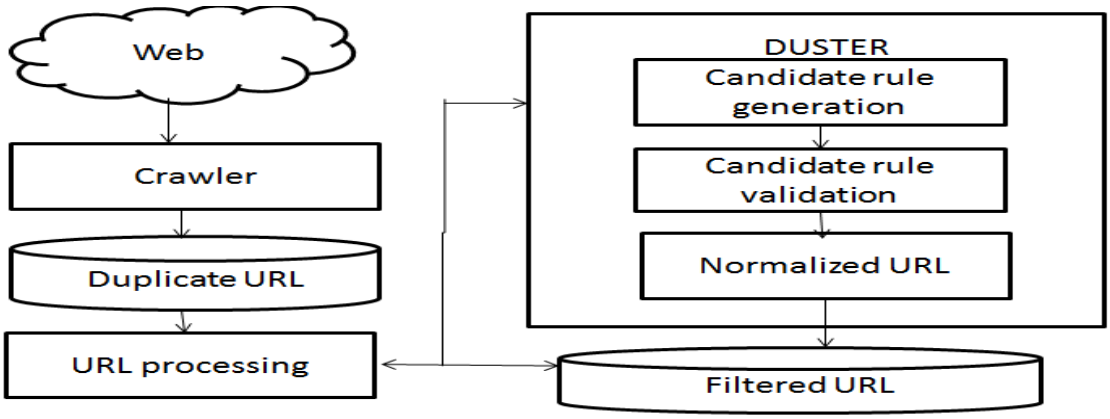

Figure 2. System Architecture

3.1. URL Processing

The main aim of this phase is to generate the consensus sequence from the Dup-cluster of URL. The detail description of URL processing is given in Figure 3.

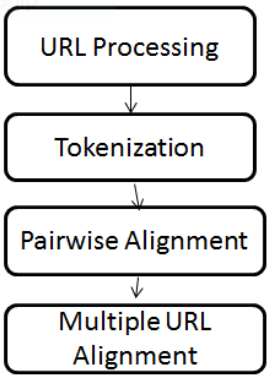

Figure 3. URL Processing

The first step of URL processing is Tokenization. Each URL to be aligned is initially parsed according to grammar G. This process, referred to as tokenization, which decomposes the URL into a sequence of singleton set called as URL tokens.
For example, URL u=http://example.com/1.htms
Then token set of URL is as below
S={http},{:},{/},{/},{example},{.},{com},{/},{htms}
After the tokenization URL are aligned using pairwise URL alignment method.

In pairwise URL alignment two URLs are taken in which mapping of two URL with another similar pair of URL with same characters, in same order with possible inserted spaces is done [11].Alignment process can be described by using a matrix S of size (m+1)*(n+1) so that S cells are filled as follows [10].

$$S_{i,j} = \left\{ max \begin{pmatrix} 0 \\ S_{i-1,j-1} + Sf(X_i, Y_j) \\ S_{i-1,j}, \\ S_{i,j-1} \end{pmatrix} if\ i = 0\ or\ j = 0\ otherwise \right\} \quad (1)$$

Where $sf\ (X_i, Y_j)$ is a scoring function that describes the similarity between pair of URL. This function gives points for matching and penalties for gap. The basic idea of scoring function is as follows (1) if the token set contains at least one token in common than the score is higher. (2)if token set contains token in common but at different position ,than score is high but smaller than in first case.(3)if token set has no token in common but token of same type than the score is small than in case 1.(4)score value indicate penalty in other case. Larger score value indicate better alignment.

After pairwise URL alignment Multiple URL alignment is done which is a progressive alignment process. Multiple URL Alignment process continues until it gives rise to a final consensus sequence. A consensus sequence is created by aligning the URLs in each cluster by using Multiple URL alignment algorithm which is discussed in section IV.

IV. ALGORITHMS

Algorithm 1:-Multiple URL Algorithm
Input:-A dup-cluster with n duplicate URL
Output:-Consensus sequence
Step 1:- A priority Queue is created for all the pair of the URL
Step 2:- Each tuple in the Queue is composed of two types of sequences X and Y and a consensus Sequence obtained by them and the alignment score alignment score
Step 3:- By using Queue it is possible to find the tuple with most similar pair of URL
Step 4:- Two sequences are removed from the set of sequences to be aligned and added to aligned sequence
Step 5:- Align the consensus sequence in the tuple with all remaining sequence to be aligned
Step 6:- Align all URL and reduced them to a single sequence of Token set
After the MUR alignment algorithm the output is send to the DUSTER Framework as shown in figure 2 above.

DUSTER framework [1] has two mains phases (1) candidate rule generation where the Multi-sequence alignment algorithm generates the candidate rule from dup-cluster. (2)Valid rules are selected according to their performance in validation set. Both the phases are discussed as below with their corresponding algorithms [1].

Phase 1: Candidate Rule Generation
Algorithm 2: Candidate Rule Generation
Input:-Training set with Duplicate Cluster
Output:-Set of Candidate Rules

Step 1:-Take a set of Dup-Cluster as input and generate a set of candidate as output also create two tables Rules table (RT) and Candidate Rules Table(CRT).
Step 2:-URL are randomly selected and aligned by Multiple URL alignment algorithm.
Step 3:- Rules are generated and added to CRT.
Step 4:- Rules are then grouped according to context and transformation.
Step 5:-Rules are same if they have same context and transformation.
Step 6:- If the Bucket size exceeds the minfreq then the Rule is considered as candidate rule.
Step 7:-Rules with same context and transformation are unified with by Union of Domain and Support of each tuple in bucket.
Step 8:-Rules are added to CRT and returned

In this phase conversion of the consensus sequence to the rules takes place. Consensus sequence is converted to rule according to token set classification.URL component plays different roles when designing URL for a website. Therefore goal is to distinguish which component should be kept, removed or generalized.

In the next phase rules are selected based on their performance in validation set. The main aim of this phase is to select a valid rule generated in previous phase which is done by two predefined thresholds false positive rate (fpr_{max}) and minimum support (min_{supp}).If the false positive rate is greater than fpr_{max} and the minimum support is smaller than min_{supp}than the rule is discarded [1].

Phase 2:-Validating Candidate Rules
Algorithm 3:-Validating Candidate Rules
Input:-Set of candidate rules
Output:-Set of valid rules
Step 1:- Create two tables Canonical Table (CT) and Rule Table (RT)
Step 2:-Set of URL's affected by candidate Rule is build.
Step 3:-when Rule matches with the URL in set the URL is normalized and added with its canonical form to CT.
Step 4:-URL with same canonical form are grouped in bucket to calculate Support
Step 5:- Instance of Rule which are not DUST are counted to calculate fpr.
Step 6:-If Rule passes to predefined criteria added to RT otherwise discarded.

URL normalization

URL normalization is the process by which URL's are modified and standardized in a consistent manner. The goal of the normalization process is to transform a URL into a normalized URL. So that it is possible to determine if two syntactically different URLs may be equivalent. The rule obtained from the validation set act as a signature to represent all URL that have very similar content.

V. EXPERIMENTAL SET UP AND RESULTS

The experiment is performed on live dataset collected from proxy web server log using squid proxy server of our institute. The log was made available in custom format which includes:-client IP address, request method (GET, POST), request URL from client, http status code and the total size of the request received. The log file includes around 60412 URL from which URLs with http status code 200 are considered and others were ignored which then includes 21000 URL.

Three parameter were taken minimum support min_{supp} is set as 10, false positive rate fpr was set as 10 and minimum frequency min_{freq} as 10.Figure 4 shows the graph for 21000 total URL, and system find out Normalized URL which are 7000 and 11000 DUST URL. Presence of Duplicate URL with similar text consumes more memory and time. Removal of such Duplicate URL with similar text using multiple sequence alignment as a backbone and the Duster frame work saves a large amount of memory and as redundant content are removed.

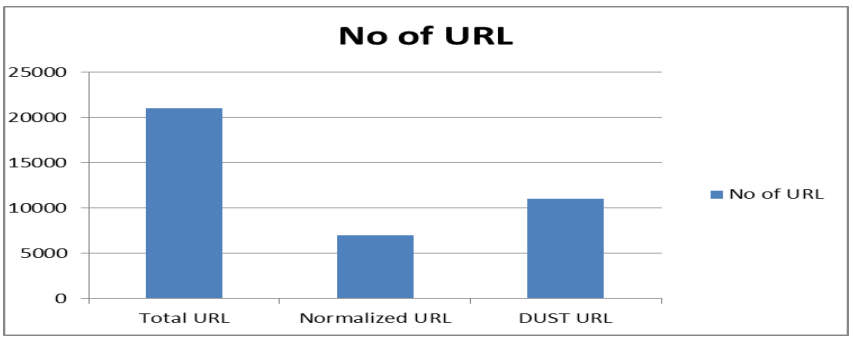

Figure 4. Number of Normalized URL

VI. CONCLUSION

The use of Multiple Sequence Alignment method helps to find the Duplicate URL's with similar text reaching the same Web page. It also save the resources such as Bandwidth, time and memory also help the user in better navigation performance also it helps to find duplicate URL in web search database. The use of DUSTER framework helps to generate more effective rules accurate normalization rule.

REFERENCES

[1] Marco Cristo, Edleno S. de Moura, and Altigran da Silva Kaio Rodrigues, "Removing DUST Using Multiple Alignment of squences," IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, vol. 27, pp. 2261-2274, August 2015.
[2] I. Keidar, and U. Schonfeld Z. Bar-Yossef, "Do Not Crawl in the DUST:Different URLs with Similar Text," ACM Trans. Web, vol. 3, pp. pp. 3:1–3:31, Janauary 2009.
[3] A. Sasturkar A. Dasgupta, R Kumar, "De-duping urls via via rewrite rules," in Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, pp. 186-194, 2008.
[4] K. P. Leela, A. Agarwal, K. P. Chitrapura, S. Garg, H. S. Koppula, "Learning url patterns for webpage deduplication," in Proc. 3rd ACM Int. Conf. Web Search Data Mining, pp. 381-390, 2010.
[5] R. Cai, J.-M. Yang, Y. Ke, X. Fan, and L. Zhang, T. Lei, "A pattern tree-based approach to learning url normalization rules," in Proc.19th Int. Conf. World Wide Web, pp. 611-620, 2010.
[6] M. F. Abulkhair, and F. E. Eassa, B. S. Alsulami, "Near duplicate document detection survey," Int. J. Comput. Sci. Commun. Netw, vol. 2, pp. 147-151, 2012.
[7] H. S. Koppula, K. P. Leela, K. P. Chitrapura, S. Garg, A. Agarwal, "Url normalization for de-duplication of Web pages," in Proc. 18th ACM Conf. Inf.knowl. Manage, pp. 1987-1990, 2009.
[8] Arvind Jain,Anish Das Sarma Gurmeet Singh Manku, "Detecting NearDuplicates for Web Crawlog," Session: Similarity Search, pp. 141-149, 2007.
[9] V. Rajapriya, "A LITERATURE SURVEY ON WEB CRAWLES," Computer Science and Mobile Applications, vol. 2, pp. 36-44, May 2014
[10] K. W. L. Rodrigues, M. Cristo, E. S. de Moura, and A. S. da Silva,"Learning url normalization rules using multiple alignment of sequences," in Proc. 20th Int. Symp. String Process. Inf. Retrieval, 2013, pp. 197–205.
[11] Alisha Gupta Satwinder Kaur, "A Survey on Web Focused Information Extraction and Algorithm," RESEARCH IN COMPUTER APPLICATIONS AND ROBOTICS, vol. 3, no. 4, pp. 19-23, April 2015.
[12] M. F. Abulkhair, and F. E. Eassa, B. S. Alsulami, "Near duplicate document detection survey," Int. J. Comput. Sci. Commun. Netw, vol. 2, pp. 147-151, 2012.
[13] Sérgio Anibal de Carvalho Junior, "Sequence Alignment algorithm,"Depatment of computer science school of physical science and engineering Kings college London,pp.1-45,2005