International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 04 Issue: 11 | Nov -2017 www.irjet.net p-ISSN: 2395-0072

Removing Dust Using Sequence Alignment and Content Matching

Priyanka Khopkar[1], D.S.Bhosale[2]

[1]PG Student, Ashokrao Mane group of institution, Vathar
[2]Associate Professor, Ashokrao Mane group of institution, Vathar

---

Abstract - *World Wide Web is a most widely used medium to search information using Web crawlers. Some of the contents related to search query collected by the web crawlers include pages with duplicate information. Different URLs with Similar contents are known as DUST. To improve the performance of search engines, a new method called is proposed. The proposed method converts all the duplicate URL into multiple sequence of alignments and removes the duplicate URLs. The proposed method works on normalization rules which convert all duplicate URLs into a single canonical form. Using this method reduction of large number of duplicate URLs is achieved. Existing Methods Complexity is proportional to the number of specific rules generated from all clusters. In the proposed system, the URL normalization process is used which identifies DUST with fetching the content of the URLs.*

Key words—Crawlers, Dust

1.INTRODUCTION

The URLs which are having similar content are called as DUST (Duplicate URLs with Similar Text). Syntactically these URLs are different but having similar content. For example, in order to facilitate the user's navigation, many web sites define links or alternative paths to access a document. In addition, webmasters usually mirror content to balance web request load and ensure fault tolerance. Other common reasons for the occurrence of duplicate content are the use of parameters placed in distinct positions in the URLs and the use of parameters that have no impact on the page content, such as the session id attribute, used to identify a user accessing the content. Detecting DUST is an extremely important task for search engines since crawling this redundant content leads to waste of resources such as Internet bandwidth and disk storage. The DUST creates disturbance in results of link analysis algorithms and also results in poor user experience due to duplicate results. To resolve these problems, several authors have proposed methods for detecting and removing DUST from search engines. Initially efforts were focused on comparing document content to remove DUST, which was again a resource consuming process. However more recent studies proposed methods that inspect only the URLs without fetching the corresponding page content. These methods, known as URL-based de-duping, mine crawl logs and use clusters of URLs referring to (near) duplicate content to learn normalization rules that transform duplicate URLs into a unified canonical form. This information can be then used by a web crawler to avoid fetching DUST, including ones that are found for the first time during the crawling. The main challenge for these methods is to derive general rules with a reasonable cost from the available training sets. As observed in [6], many methods derive rules from pairs of duplicate URLs. Thus the quality of these rules is affected by the criterion used to select these pairs and the availability of specific examples in the training sets. To avoid processing large numbers of URLs, most of the methods employ techniques such as random sampling or by looking for DUST only within sites, preventing the generation of rules involving multiple DNS names. Because of these issues, current methods are very susceptible to noise and, in many cases, derive rules that are very specific. Thus, an ideal method should learn general rules from few training examples, taking maximum advantage, without sacrificing the detection of DUST across different sites.

People use search engines for searching information. But retrieved documents contains a large volume of duplicate documents. Hence there is need to improve the search results. Data filtering algorithms used by some of search engines which eliminate duplicate and partial duplicate documents to save time and effort. In proposed system, multiple sequence alignment and URLs matching methods are used. For any given set of sequences with more than two sequences, multiple sequence alignment is considered as a natural generalization of the pair wise alignment problem. This problem requires that all sequences to be of the same length and thus spaces are inserted at the appropriate places. In the proposed system, multiple sequence alignment is used to obtain a smaller and general set of rules to avoid duplicate URLs. Multiple sequence alignment is used for identifying identical patterns. This Multiple sequence alignment can be used to identifying similar strings, which can be used for deriving normalization rules.

The Duster has following objectives:

The main objective of the proposed work is to detect and remove duplicate url's.

1. To Remove duplicate URLs from web.
2. To Reduce Load of Server.
3. To improve the performance of search results
4. To improve the speed of server side.
5. To overcome the limitations of existing methods of de-duping.

Following Approaches are used for Duster:

1. Multiple Sequence Alignment Algorithm uses progressive alignment strategy that aligns most similar sequences at each stage and infers a new token set of rules from them.

2. A Url Multi Alignment algorithm is used that works on the consensus sequence on entire clusters that are formed by the tokenization of these URLs.

3. A Candidate Rules algorithm which takes a set of dup-clusters as input and generates a set of candidate rules as output.

4. Validate Rules algorithm which takes a set of candidate rules as input and outputs a set of valid rules.

B. Architecture

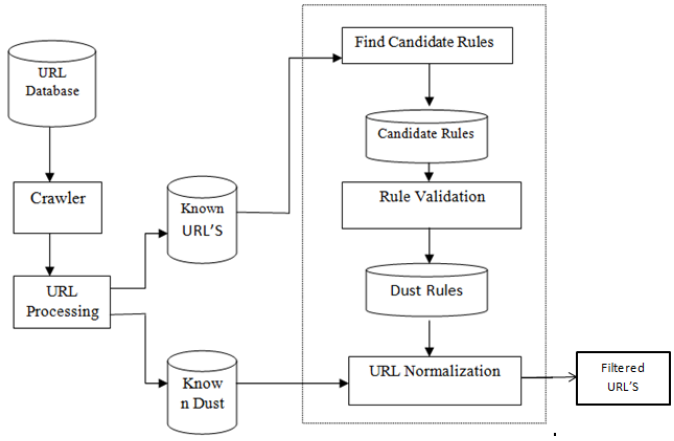

Fig 1: System Architecture

As shown in above figure, once a new set of URLs is crawled, it is merged with the already known URLs to form a new set of known URLs. These URLs are then categorized into known URLs set and known DUST set. During crawling, the crawler is also able to identify examples of DUST by canonical tags. As a result, a new set of known DUST is also available. Further this set can be enriched by processes such as those based on content signature, and manual inspection. Once the final set of known DUST is crawled, it is used to find and validate rules. It does it by splitting it in training and validating sets. The resulting rules are then used to normalize the known URLs which generates, a new reduced set of URLs to be crawled. The proposed architecture consists of two modules.

1.URL Dataset Visualization

Admin can login and manage the categories of products. The admin user can add/update product information, images and description. Also, the admin has right to approve the order requested by visitor, generate invoices and pass to dispatch team. Admin module has the option to add/update advertisements. Also, admin can provide the product details, advertise details and order details to visitor through web services.

2. Tokenization and Clustering

Basic tokenization is the process of parsing URL to extract tokens. The protocols, hostnames, query arguments and the path components are also extracted from the specified standard delimiters. Firstly, clusters are formed with the URLs in the datasets. Then, anchors are selected from the URL clusters formed in the previous step. The selected anchors are validated and if the anchors are found to be valid, then the child pattern is generated. If the anchors are not valid, they do not generate child pattern. Then, the process of generating tokenized key value pairs and associates them to the original URL in order to generate deep tokenized URLs is known as Deep tokenization. The URL encodings are learnt by a specialized technique that doesn't require any supervision. This process is iterative defined and conducted as per the decision tree generated. The process of cluster formation with the URLs are known as Clustering. It is the basic step of module 2 in which the cluster is formed and is produced to the rule generalization module. The URLs which consists of more similarity in the web page content is termed as a duplicate cluster. The rules are generated for all the URL pair present in the duplicate clusters.

3. Tokenization and Clustering

Pair wise rule generation module is designed for generating pair wise rules from the URLpairs of the duplicate clusters. The transformational rules are framed in this module. This is the critical part of the work which decides the efficient working of de-duplication process.

Here target URLs are used for generating transformation. The clusters have few URLs which are closest to the normal URL. Out of these URLs, one is selected as source URL. The source URL is changed frequently based on the study. Now, using these source and target URLs the pair wise rule is generated. The learning of these pair wise rules generated through URLs out of duplicate cluster happens which are further generalized so as to normalize the unseen URLs as well.

In the process of rule generalization, one of the cluster is selected from the cluster groups. A key is selected from the previously selected cluster. We know that all keys have information gains, so the key selection is made by studying the maximum information gains possessed by the key. Finally the transformation process of transmitting the source to the target is performed. Generalization is performed by generating a decision tree. This tree is constructed with the

selected keys and their branch is formed with the key's matching pair else it is branched out with a wildcard. Number of linear rules generated by the generalization technique. Only after rule generalization, the new values can be accommodated. The decision tree based generalization enabled the work to be error proof and robust. The so generated decision tree follows bottom-Up approach. The rules are used in online mode and hence the memory requirement to store these rules are minimal. The contexts as well as transformation format are generated by the rule generalization process. Thus it provides a compatible contexts as well as target URLs. The feasibility is improved as the iteration counts high. During the initial phase, the frequency is generated for each key. Then context generalization is performed. The generalization is performed over the contexts.

4. Comparison

This is the ultimate step to present the non-redundant, de-duplicated data for the users. With two data sets in hand, the number of de-duplicated data are estimated. The comparison module is presented, where the maximum number of duplicates are detected. Thus the matches are identified and are produced for display without any duplicated data. There is a steady improvement as the proposed pair wise rule generalization is an iterative algorithm.

C. Implementation steps

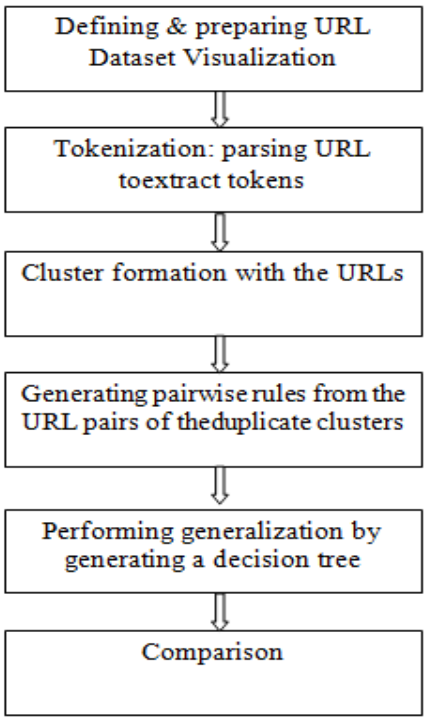

Fig 2 : Implementation steps

D. SCOPE OF THE WORK

The main goal of the Duster technique is to remove duplicate url,s from web.

1.To remove duplicate url,s from web using multiple sequence alignment.

2.Presents an approach that improve performance of search result and improve the speed of server.

3.Present a system to improve the speed and accuracy of recommendation system in big data application.

2.RESULT EVALUATION

Comparative analysis:

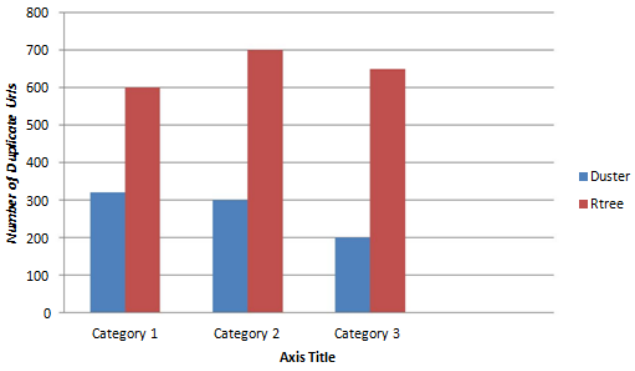

Fig 3. Comparative Results

3. CONCLUSION AND FUTURE WORK

In this work, we presented, a new method to address the DUST problem, that is, the detection of distinct URLs that correspond to pages with duplicate or near duplicate content. This Technique learns normalization rules that are very precise in converting distinct URLs which refer the same content to a common canonical form, making it easy to detect them. To achieve this, we applies a novel strategy based on a full multi-sequence alignment of training URLs with duplicate content.

As future work, we intend to improve the scalability and precision of our method, as well as to evaluate it using other datasets. For its scalability, we intend to provide a comprehensive comparison among strategies to cope with very large dup-clusters.

4.REFERENCES

[1] "Url normalization for de-duplication of web pages" by A. Agarwal, H. S. Koppula, K. P. Leela, K. P. Chitrapura, S. Garg, P. K. GM.

[2] "Near duplicate document detection survey" by B. S. Alsulami, M. F. Abulkhair, and F. E. Eassa.

[3] "Do not crawl in the dust: Different urls with similar text*" by Z. Bar-Yossef, I. Keidar, and U. Schonfeld.

[4] "De-duping urls via rewrite rules" by A. Dasgupta, R. Kumar, and A. Sasturkar.

[5] "Learning url patterns for webpage deduplication" by H. S. Koppula, K. P. Leela, A. Agarwal, K. P. Chitrapura, S. Garg, and A. Sasturkar.

[6] "A pattern tree-based approach to learning url normalization rules" by T. Lei, R. Cai, J.-M. Yang, Y. Ke, X. Fan, and L. Zhang.

[7] "SpotSigs: robust and efficient near duplicate detection in large web collections" by M. Theobald, J. Siddharth, and A. Paepcke.

[8] "Overview of the TREC 2004 terabyte track," by C. L. A. Clarke, N. Craswell, and I. Soboroff, " in Proc. 13th Text Retrieval Conf., 2004, pp. 2–3.

[9] "A novel method for rapid multiple sequence alignment based on fast Fourier transform,"by K. Katoh, K. Misawa, K. Kuma, and T. Miyata. (2002).

[10] "Duplicate and near duplicate documents detection: A review" by J. P. Kumar and P. Govindarajulu.