International Journal of Current Research and Modern Education (IJCRME)
ISSN (Online): 2455 - 5428
(www.rdmodernresearch.com) Volume I, Issue I, 2016

ELIMINATING NOISY CONTENTS IN WEB DOCUMENTS USING HYBRID DUSTER FRAMEWORK

A. Reena Risi* & R. Selvakumar**

* PG Scholar, Department of Master of Computer Applications, Dhanalakshmi Srinivasan Engineering College, Perambalur, Tamilnadu

** Assistant Professor, Department of Master of Computer Applications, Dhanalakshmi Srinivasan Engineering College, Perambalur, Tamilnadu

Abstract:

Web Page Noise Cleaning is one of the new research areas of study for removing the noise patterns of web pages for effective web mining. The World Wide Web contains large amount of web pages which are accessible by users. With conventional data or text, Web pages generally contain a large amount of noise information that is not part of the main contents of the web pages, e.g., advertisement banners, navigation bars, and disclaimer/copyright notices. The main objective of this area is removing such irrelevant information (i.e. Web Page Noise or Local Noise) in Web pages that can seriously harm Web mining task such as clustering and classification etc. In our work we focus on identifying and removing local noises in web pages to improve the performance of mining. A simple idea for detection and removal of noises a new DOM tree structure is proposed. After DOM tree construction, we can implement DUSTER framework to crawling the document using normalized rules. The result shows the remarkable increase in F score and accuracy is obtained. In this work, we focus on detecting and eliminating local noises in Web pages to improve the performance of Web mining that is Web page clustering and classification. Then our experimental results show that improved performance at the time of classification and clustering.

Introduction:

Data Mining:

Data mining (the analysis step of the "Knowledge Discovery in Databases" process, or KDD), a field at the intersection of *computer science* and *statistics*, is the process that attempts to discover patterns in large *data sets*. It utilizes methods at the intersection of *artificial intelligence*, *machine learning*, *statistics*, and *database systems* The overall goal of the data mining process is to extract information from a data set and transform it into an understandable structure for further use Aside from the raw analysis step, it involves database and *data management* aspects, *data preprocessing*, *model* and *inference* considerations, interestingness metrics, *complexity* considerations, post-processing of discovered structures, *visualization*, and *online updating*. Generally, data mining (sometimes called data or knowledge discovery) is the process of analyzing data from different perspectives and summarizing it into useful information - information that can be used to increase revenue, cuts costs, or both. Data mining software is one of a number of analytical tools for analyzing data. It allows users to analyze data from many different dimensions or angles, categorize it, and summarize the relationships identified. Technically, data mining is the process of finding correlations or patterns among dozens of fields in large relational databases.

Data:

Data are any facts, numbers, or text that can be processed by a computer. Today, organizations are accumulating vast and growing amounts of data in different formats and different databases. This includes:

- ✓ Operational or transactional data such as, sales, cost, inventory, payroll, and accounting
- ✓ Nonoperational data, such as industry sales, forecast data, and macro economic data
- ✓ Meta data - data about the data itself, such as logical database design or data dictionary definitions

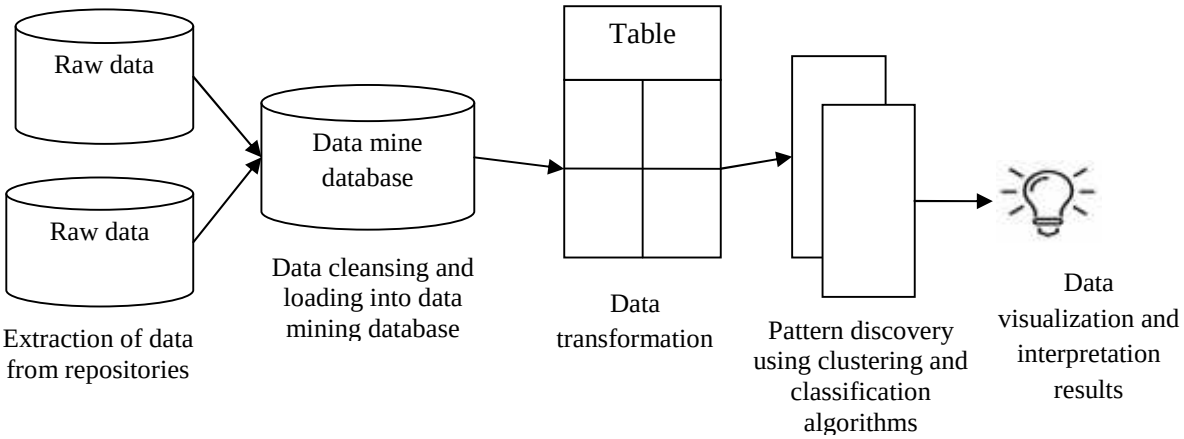


Figure 1: Process of Data Mining

Information:

The patterns, associations, or relationships among all this data can provide information. For example, analysis of retail point of sale transaction data can yield information on which products are selling and when.

Knowledge:

Information can be converted into knowledge about historical patterns and future trends. For example, summary information on retail supermarket sales can be analyzed in light of promotional efforts to provide knowledge of consumer buying behavior. Thus, a manufacturer or retailer could determine which items are most susceptible to promotional efforts.

Data Warehouses:

In computing, a data warehouse (DW or DWH) is a database used for reporting and data analysis. It is a central repository of data which is created by integrating data from multiple disparate sources. Data warehouses store current as well as historical data and are commonly used for creating trending reports for senior management reporting such as annual and quarterly comparisons. The data stored in the warehouse are uploaded from the operational systems (such as marketing, sales etc., shown in the figure to the right). The data may pass through an operational data store for additional operations before they are used in the DW for reporting. The typical ETL-based data warehouse uses staging, integration, and access layers to house its key functions. The staging layer or staging database stores raw data extracted from each of the disparate source data systems. The integration layer integrates the disparate data sets by transforming the data from the staging layer often storing this transformed data in an operational data store (ODS) database. The integrated data are then moved to yet another database, often called the data warehouse database, where the data is arranged into hierarchical groups often called dimensions and into facts and aggregate facts.

A data warehouse constructed from integrated data source systems does not require ETL, staging databases, or operational data store databases. The integrated data source systems may be considered to be a part of a distributed operational data store

layer. Data federation methods or data virtualization methods may be used to access the distributed integrated source data systems to consolidate and aggregate data directly into the data warehouse database tables. Unlike the ETL-based data warehouse, the integrated source data systems and the data warehouse are all integrated since there is no transformation of dimensional or reference data. This integrated data warehouse architecture supports the drill down from the aggregate data of the data warehouse to the transactional data of the integrated source data systems.

Data warehouses can be subdivided into data marts. Data marts store subsets of data from a warehouse. This definition of the data warehouse focuses on data storage. The main source of the data is cleaned, transformed, cataloged and made available for use by managers and other business professionals for data mining, online analytical processing, market research and decision support However, the means to retrieve and analyze data, to extract, transform and load data, and to manage the data dictionary are also considered essential components of a data warehousing system. Many references to data warehousing use this broader context. Thus, an expanded definition for data warehousing includes business intelligence tools, tools to extract, transform and load data into the repository, and tools to manage and retrieve metadata.

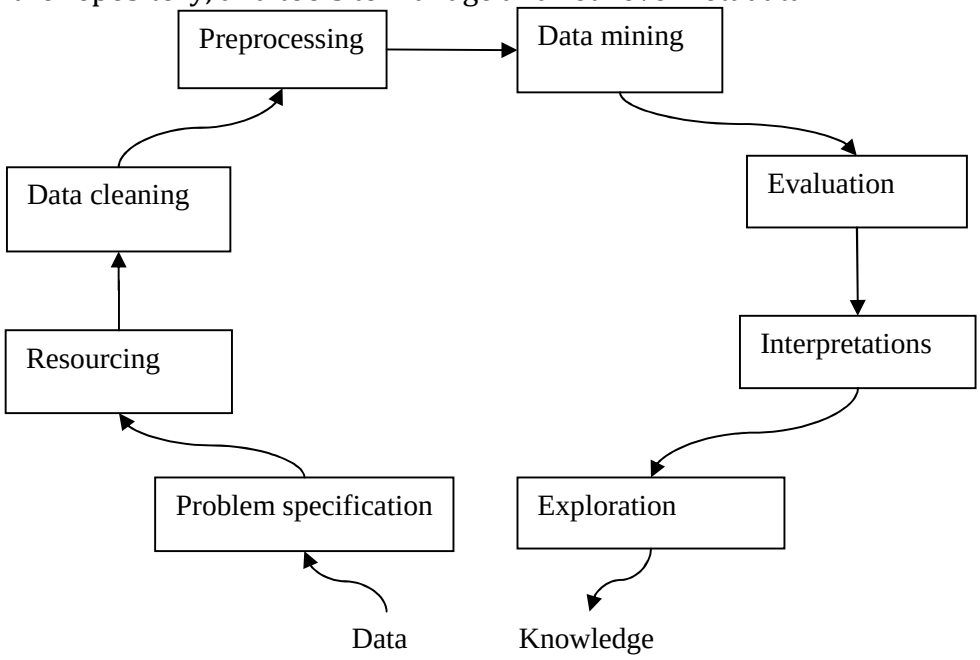


Figure 2: Levels of Data Mining

Dramatic advances in data capture, processing power, data transmission, and storage capabilities are enabling organizations to integrate their various databases into data warehouses. Data warehousing is defined as a process of centralized data management and retrieval. Data warehousing, like data mining, is a relatively new term although the concept itself has been around for years. Data warehousing represents an ideal vision of maintaining a central repository of all organizational data. Centralization of data is needed to maximize user access and analysis. Dramatic technological advances are making this vision a reality for many companies. And, equally dramatic advances in data analysis software are allowing users to access this data freely. The data analysis software is what supports data mining. It enables these companies to determine relationships among "internal" factors such as price, product positioning, or staff skills, and "external" factors such as economic indicators, competition, and customer demographics. And, it enables them to determine the impact on sales,

customer satisfaction, and corporate profits. Finally, it enables them to "drill down" into summary information to view detail transactional data.

Data Mining Elements:

- ✓ Extract, transform, and load transaction data onto the data warehouse system.
- ✓ Store and manage the data in a multidimensional database system.
- ✓ Provide data access to business analysts and information technology professionals.
- ✓ Analyze the data by application software.
- ✓ Present the data in a useful format, such as a graph or table.

Different Levels of Analysis:

Artificial Neural Networks: Non-linear predictive models that learn through training and resemble biological neural networks in structure.

Genetic Algorithms: Optimization techniques that use process such as genetic combination, mutation, and natural selection in a design based on the concepts of natural evolution.

Decision Trees: Tree-shaped structures that represent sets of decisions. These decisions generate rules for the classification of a dataset. Specific decision tree methods include Classification and Regression Trees (CART) and Chi Square Automatic Interaction Detection (CHAID). CART and CHAID are decision tree techniques used for classification of a dataset. They provide a set of rules that you can apply to a new (unclassified) dataset to predict which records will have a given outcome. CART segments a dataset by creating 2-way splits while CHAID segments using chi square tests to create multi-way splits. CART typically requires less data preparation than CHAID.

Nearest Neighbor Method: A technique that classifies each record in a dataset based on a combination of the classes of the k record(s) most similar to it in a historical dataset (where k 1). Sometimes called the k-nearest neighbor technique.

Rule Induction: The extraction of useful if-then rules from data based on statistical significance.

Data Visualization: The visual interpretation of complex relationships in multidimensional data. Graphics tools are used to illustrate data relationships.

Data Mining Techniques: There are several major data mining techniques have been developed and used in data mining projects recently including association, classification, clustering, prediction and sequential patterns.

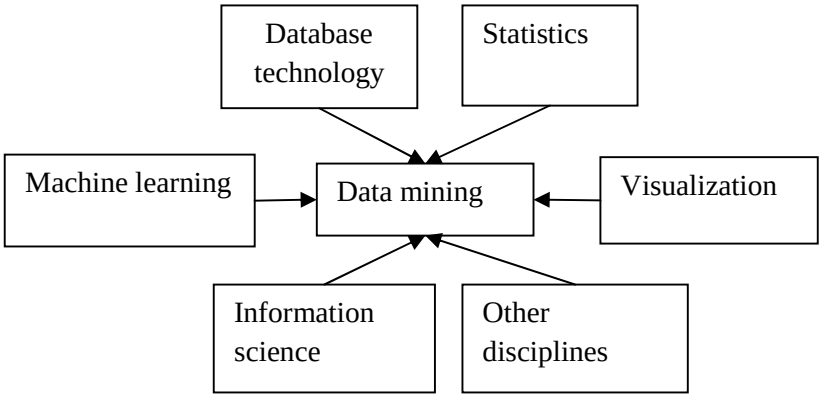


Figure 3: Techniques of Data Mining

Association:

Association is one of the best known data mining technique. In association, a pattern is discovered based on a relationship of a particular item on other items in the

same transaction. For example, the association technique is used in market basket analysis to identify what products that customers frequently purchase together. Based on this data businesses can have corresponding marketing campaign to sell more products to make more profit.

Classification:

Classification is a classic data mining technique based on machine learning. Basically classification is used to classify each item in a set of data into one of predefined set of classes or groups. Classification method makes use of mathematical techniques such as decision trees, linear programming, neural network and statistics. In classification, make the software that can learn how to classify the data items into groups. For example, can apply classification in application that "given all past records of employees who left the company, predict which current employees are probably to leave in the future." In this case, divide the employee's records into two groups that are "leave" and "stay".

Clustering:

Clustering is a data mining technique that makes meaningful or useful cluster of objects that have similar characteristic using automatic technique. Different from classification, clustering technique also defines the classes and put objects in them, while in classification objects are assigned into predefined classes. To make the concept clearer, can take library as an example. In a library, books have a wide range of topics available. The challenge is how to keep those books in a way that readers can take several books in a specific topic without hassle.

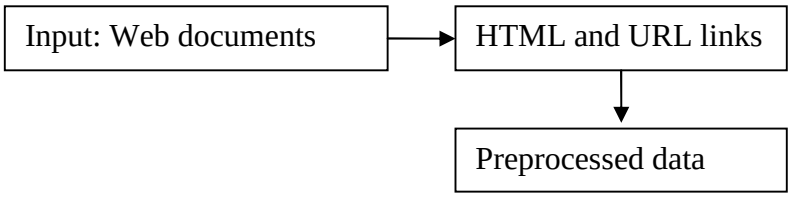
Implementation:

Modules Description:

Web Documents Acquisition:

Elimination of noisy and irrelevant contents from web pages has many applications, including web page classification, clustering, web featuring, proper indexing of search engines, efficient focused crawlers, cell phones and PDA browsing, speech rendering for the visually impaired, improving the quality of search results and text summarization. Thus cleaning web pages for web data extraction becomes crucial for improving the performance of information retrieval. A web document is similar in concept to a web page.

Every Web document has its own URI. Note that a Web document is not the same as a file: a single Web document can be available in many different formats and languages, and a single file, for example a PHP script, may be responsible for generating a large number of Web documents with different URIs. A Web document is defined as something that has a URI and can return representations (responses in a format such as HTML or JPEG or RDF) of the identified resource in response to HTTP requests. In this module, can get the datasets as web documents. A web document contains Content, HTML codes and so on. Then read the codes from upload web datasets. And perform preprocessing steps to tokenize the each code.



Algorithm:

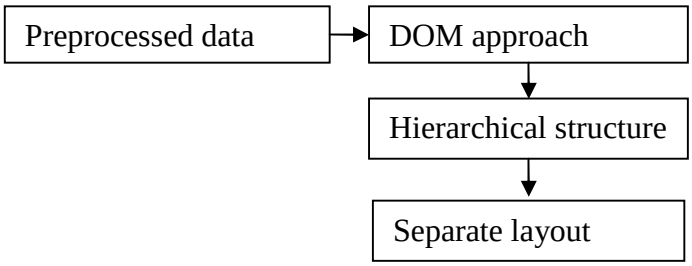
Algorithm 1 MultipleURLAligment (C)

Input: A dup-cluster $\mathcal{C} = \{u_1, \ldots, u_n\}$ with n duplicate URLs
Output: A tuple $\pi = (consensus, domains, support)$.

Let $\mathcal{Q}$ be a priority queue in which tuples $\sigma = (x, y, consensus_{xy}, scoring)$ are sorted in descending order according to the alignment $scoring$.
$Domains = \emptyset$; $Sequences = \emptyset$; $Support = \emptyset$; $Aligned = \emptyset$;
for all pairs of distinct urls u_1, u_2 in $\mathcal{C}$ **do**
  $Support = Support \cup \{(u_1, u_2)\}$
  $Domains = Domains \cup \{domain(u1)\} \cup \{domain(u2)\}$
  $x = tokenize(u1)$
  $y = tokenize(u2)$
  $Sequences = Sequences \cup \{x\} \cup \{y\}$
  $\sigma = PairURLAlignment(x, y)$
  add σ to $\mathcal{Q}$
end for
while $\mathcal{Q}$ is not empty **do**
  Pop the first tuple σ from $\mathcal{Q}$
  if $\sigma.x \notin Aligned$ and $\sigma.y \notin Aligned$ **then**
    $Aligned = Aligned \cup \{\sigma.x\} \cup \{\sigma.y\}$
    $Sequences = Sequences - Aligned$
    for all sequences s in $Sequences$ **do**
      $c = PairURLAlignment(\sigma.consensus, s)$
      add c to $\mathcal{Q}$
    end for
    $Sequences = Sequences \cup \{\sigma.consensus\}$
  end if
end while
Let s be the unique consensus sequence in $Sequences$
return $\pi = (s, Domains, Support)$

DOM Implementation:

We can upload web documents as HTML pages and implement DOM tree. The Document Object Model (DOM) specification is an object-based interface developed by the World Wide Web Consortium (W3C) that builds an XML and HTML document as a tree structure in memory. An application accesses the XML data through the tree in memory, which is a replication of how the data is actually structured. The DOM also allows the user to dynamically traverse and update the XML document. It provides a model for the whole document, not just for a single HTML tag. The Document Object Model represents a document as a tree. DOM trees are highly transformable and can be easily used to reconstruct a complete webpage. DOM tree is a well defined HTML document model. Some HTML tags do not include a closing bracket.

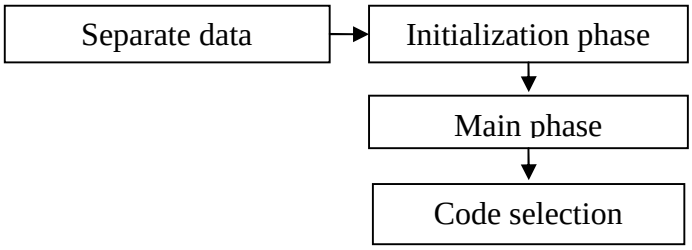


In order to analyze a web page, we first check the syntax of HTML document because most HTML Web pages are not well-formed. And then we pass web pages through an HTML parser, which corrects the markup and creates a Document Object Model (DOM) tree. The system must know the maximum level of DOM tree to choose the good choice of threshold level. Therefore, the system traverses the whole DOM tree to get the maximum depth of DOM. For the training data set, we picked the best suited threshold level up by setting various threshold levels. Then, the system chooses the suitable threshold level for test data set by using these known pair of series. The system

estimates the nature of the relationship between the maximum level and threshold level based on linear regression analysis. A regression is a statistical analysis assessing the association between two variables.

Duster Framework Construction:

In this module implement DUSTER frame work to identify similarities and differences among strings/sequences. These similarities and differences can be explored to determine fixed and mutable substrings in contents, which help to derive normalization rules. As multiple sequence alignment methods find patterns involving all the available strings, the method is able to find more general rules and avoids problems related to pair wise rule generation, and the problem related to finding rules across sites. Thus, a full multi-sequence alignment of duplicate URLs, which is performed before rules are generated, can make the learning process more robust and less susceptible to noise.



Algorithm

Algorithm 2 GenerateCandidateRules ($\mathcal{TS}$)

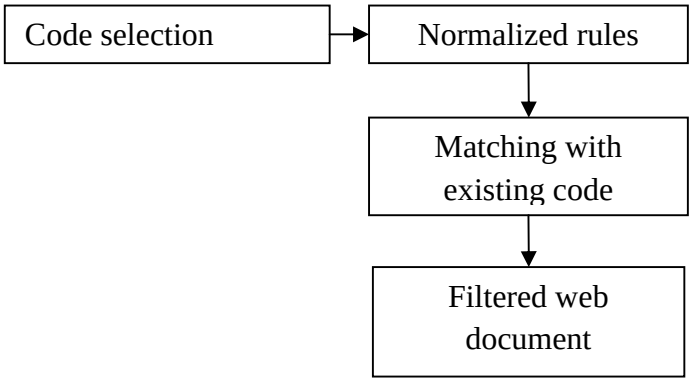
Input: Training Set $\mathcal{TS} = \{c_1, \ldots, c_n\}$ with n duplicate clusters
Output: Set of m candidate rules $\mathcal{CR} = \{r_1, \ldots, r_m\}$

```
Create table RT (contcxt, transformation, domains, support)
Create table CRT (context, transformation, domains, support)
for all cluters c_i ∈ TS do
    T = selectKRandomlyURLsFrom(c_i)
    π = MultipleURLAligment(T)
    r = generateRule (π.consensus)
    add (r.context, r.transformation, π.domains, π.support) to RT
end for
group tuples in RT into buckets by (context, transformation)
for all buckets B do
    if (|B| >= min_freq) then
        D_omains = ∅; S_upport = ∅;
        for all tuples t ∈ B do
            D_omains = D_omains ∪ t.domains
            S_upport = S_upport ∪ t.support
        end for
        α = the first tuple in B
        add (α.context, α.transformation, D_omains, S_upport) to CRT
    end if
end for
return a set CR of rules created from CRT
```

Web Content Prediction:

Web document classification or document categorization is a problem in library science, information science and computer science. The task is to assign a document to one or more classes or categories. This may be done "manually" (or "intellectually") or algorithmically. The intellectual classification of web documents has mostly been the province of library science, while the algorithmic classification of web documents is used mainly in information science and computer science. The problems are

overlapping, however, and there is therefore also interdisciplinary research on document classification. This process is known as testing process. Used validation rules algorithm; we can classify web documents based on supervised clusters.



Algorithm:

Algorithm 3 ValidateRules ($\mathcal{VS}$, $\mathcal{CR}$, fpr_{max}, min_{supp})

Input: $\mathcal{VS}$: validation set, $\mathcal{CR}$: Set of n candidate rules, fpr_{max}: maximum false-positive rate that can be tolerated, min_{supp}: minimum number of instances required.
Output: Set of n valid rules $\mathcal{VR} = \{r_1, \ldots, r_n\}$

```
1: Create table CT (canonical, url)
2: Create table RT (context, transformation, domains, support)
3: for all candidate rules r in CR do
4:     Nsupp = 0; Nfpp = 0; S_support = ∅;
5:     U = ∪_{d ∈ r.Domains} URLs from domain d in VS.
6:     for all urls u in U do
7:         if r.context matches with u then
8:             canonical = NormalizeURL (u, r)
9:             add (canonical, u) to CT
10:        end if
11:    end for
12:    group tuples in CT into buckets by (canonical)
13:    for all buckets B do
14:        if (|B| > 1) then
15:            for all pairs of distinct tuple t1,t2 ∈ B do
16:                Nsupp = Nsupp + 1
17:                S_support = S_support ∪ {(t1.url, t2.url)}
18:                if (t1.url and t2.url are not DUST) then
19:                    Nfpp = Nfpp + 1
20:                end if
21:            end for
22:        end if
23:    end for
24:    if (Nsupp >= min_supp) then
25:        fpr = Nfpp/Nsupp
26:        if (fpr <= fpr_max) then
27:            add (r.context, r.transformation, r.domains, S_support) to
               RT
28:        end if
29:    end if
30:    Clear table CT
31: end for
32: return a set of all rules in RT
```

Evaluation Criteria:

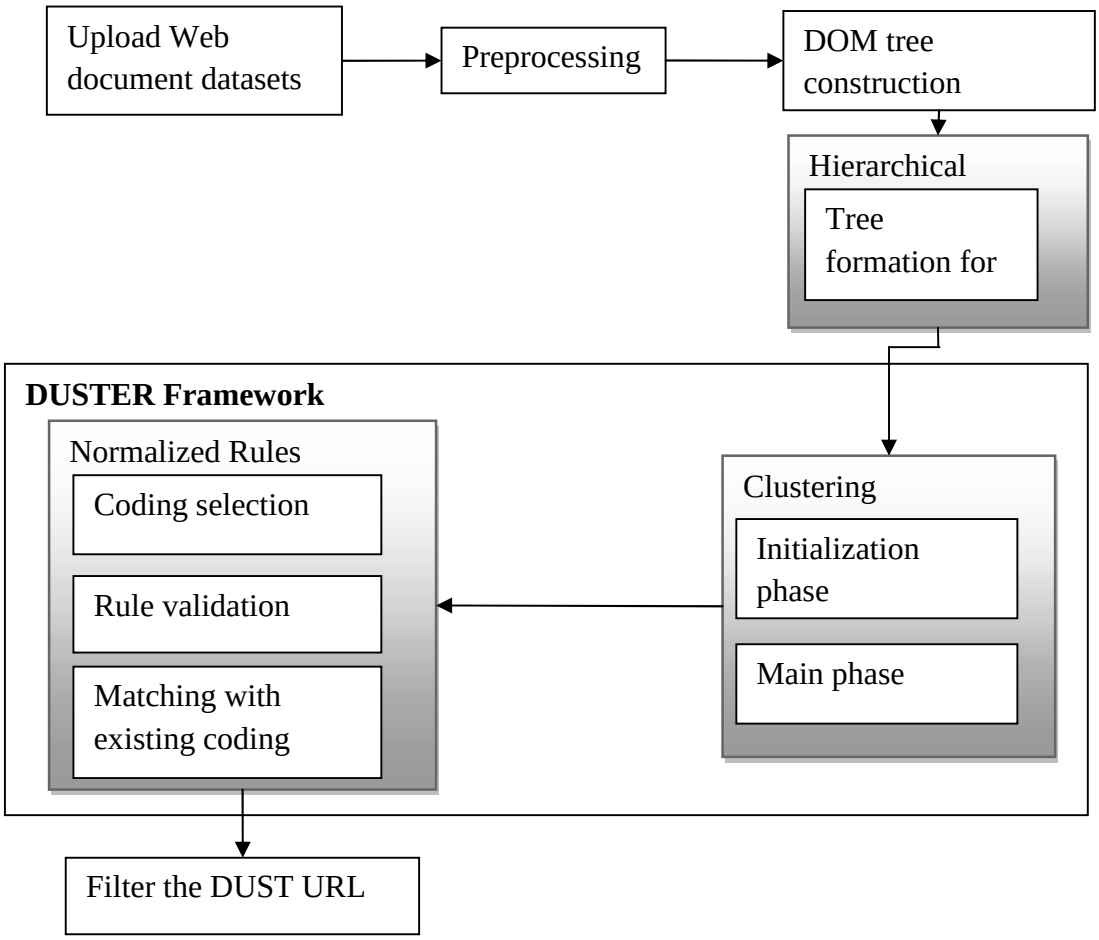
Finally we analyzed side information to eliminate noises and improve quality of text documents. And to provide improved results in time complexity problems. In this module provide mobile intimation system to intimate the admin at the time of unwanted content.

Algorithm Structure:

- ✓ Input: Multiple web Documents
- ✓ Method: DUSTER based DOM Method
- ✓ Output: Extraction of relevant web documents free of noise.
- ✓ Step 1: Access multiple web page
- ✓ Step 2: Read one by one page
- ✓ Step 3: Check Web HTML tag

- ✓ Step 4: Consider the document with various tags
- ✓ Step 5: create DOM Tree structure using HTML parser.
- ✓ Step 6: Train the dataset in database where all the information related to web pages is stored for efficient retrieval of pattern by using DUSTER.
- ✓ Step 7: Match the constructed DOM tree with the information in the database using DUSTER and retrieve similar kind of information or pattern which contains noisy data and eliminate it from each web page.
- ✓ Step 8: Finally receive web page without noisy data using normalized rules.

System Architecture:



Conclusion:

Organizing and removing noise from web pages will get better on correctness of search results as well as explore speed, and may advantage web page association purpose. For removal of noise Dom tree construction is always feasible as it converts the complex page into simplified form. DUSTER framework has many advantages which help to store and retrieve better results from the database. The proposed technique aims at helping document classification in web content mining based on a new tree structure, featured DOM tree, and DUSTER based crawling method for similarity verification. Instead of processing a set of web pages as such, we proposed a three stage algorithm which runs on a single web page and increases the mining result remarkably.

In this project, we focus an optimal feature subset selection method along with a similarity verification method for identifying noisy blocks of a page. We could detect and remove local noises with an increased relevancy. We evaluate the performance of our algorithm in terms of F score and accuracy of web page classification and we could

achieve an improved result with a large margin than before cleaning. Further research works can extend this to a more efficient method for directly finding main content blocks rather than identifying and pruning noisy blocks. It can be incorporated with search engines for better indexing and page ranking. Accuracy can be improved further by devising more efficient methods for optimal feature subset selection. Also this method can be easily associated with block classification of web pages directly with the help of featured DOM tree.

Future Enhancement:

Web Page Noise Cleaning is one of the new research areas of study for removing the noise patterns of web pages for effective web mining. The World Wide Web contains large amount of web pages which are accessible by users. With conventional data or text, Web pages generally contain a large amount of noise information that is not part of the main contents of the web pages,. In future we can extend our application to analyze the web noises to using various data mining algorithms to improve the accuracy rate with limited number of error rates.

References:

1. "Neural Networking using Multiple Web Page Noise Removing Method" P.Siva Kumar, Dr. R.M.S Parvathi IJCST Vol. 3, Issue 1, Jan. -March 2012
2. "Neural Networks In Data Mining", dr. Yashpal Singh, alok Singh Chauhan. Journal of Theoretical and Applied Information Technology.
3. "Elimination of Noisy Information from Web Pages", Alpa K. Oza, Shailendra Mishra. International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-2, Issue-1, March 2013.
4. L. Yi, B. Liu, X. Li., "Eliminating Noisy Information in Web Pages for Data Mining", in Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. Washington, DC, USA, 2003.
5. Y. Yang, H. J. Zhang, "HTML page analysis based on visual cues", In Proceedings of the Sixth International Conference on Document Analysis and Recognition, pp. 859– 864, Washington, DC, USA, 2001.
6. Jinbeom Kang, Joongmin Choi, "Block classification of a web page by using a combination of multiple classifiers", Fourth International Conference on Networked Computing and Advanced Information Management, pp 290 -295, September 2008.
7. Thanda Htwe, Khin Haymar Saw Hla, "Noise Removing from Web Pages Using Neural Network", The 2nd International Conference on Computer and Automation Engineering, Singapore, Volume 1, pp. 281 – 285, February 2010.
8. Ziv Bar-Yossef, Sridhar Rajagopalan, "Template Detection via Data Mining and its Applications", Proceedings of the 11th international conference on World Wide Web, pp 580-591, 2002.
9. Shian-Hua Lin, Jan-Ming Ho, "Discovering informative content blocks from Web documents", Proceedings of ACM SIGKDD'02, July 2002.
10. Jingqi Wang, Qingcai Chen, Xiaolong Wang, Hongzhi Guo, "Basic Semantic Units Based Web Page Content Extraction", International Conference on Systems, Man and Cybernetics, pp 1489 – 1494, 2008.