Volume 5, Issue 4, April 2015 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Analytics of Noisy Data in Web Documents Using a Dom Tree

S. Mythili, T. Vetriselvi
Computer Science and Engineering
K. Ramakrishnan College of Technology, India

Abstract* — In many web mining applications, side –information is available along with web documents. Such side information provides both relevant content as well as irrelevant content. The irrelevant contents are called as noisy blocks. Noisy blocks can be of advertisement, navigation panels, copyright and privacy notices etc .In general web mining is the p Incorporating these noisy blocks in to mining process can degrade the quality of web data mining. Generally, Web mining is the process of determining useful information from the web. A tremendous amount of information is available in web documents obtaining a pure web content is quite difficult. Therefore, a principle way to perform the mining process is needed to maximize the advantages of side information. To address this issue, Document object Model tree are constructed for a web document and fitness values are computed using a Genetic algorithm (GA) to determine the optimal number of clusters in a dataset. This work is extended to perform both clustering and classification task using (Web Content and auxiliary attribute based Text Clustering) WCOATES and (Web Content and auxiliary attribute based Text Classification) WCOLT algorithm respectively. By incorporating these techniques in to web mining process noisy blocks are pruned out and there by improve the results of web data mining significantly.

***Keywords* – Side Information; DOM Tree; Genetic Algorithm; WCOATES; WCOLT.**

I. INTRODUCTION

Text mining is a burgeoning field that attempts to glean meaningful information from natural language text. Usually text is unstructured, amorphous, and difficult to obtain an exact pattern. The phrase “text mining” is generally used to denote any system that analyzes large quantities of natural language text and detects lexical or linguistic usage patterns in an attempt to extract useful information. Text mining attempts to derive high quality information from a large corpus even though, issue arises in various domains where a tremendous amount of side information is available along with the documents. The text documents typically occur in the context of a variety of applications in which there may be a large amount of other kinds of database attributes or meta information which may be useful to the clustering process[7],[12]. Example of such side attributes may be user-access behaviour in the form of web logs, text documents that contains links among them or meta data in a web documents etc. While such side-information can sometimes be useful in improving the quality of clustering process, it can also be a risky approach when the side information is noisy. In such cases, it can actually worsen the quality of mining process.

In addition to Text mining, Web mining plays a major role in recent years. Web mining is the process of discovering useful information from web documents. In the contemporary world, Internet has made the WWW a popular place for collecting and sharing information. Mining on the Web has becomes an important task for extracting useful information from the Web. However, useful information on the Web is often accompanied by a large amount of noise such as banner advertisements, navigation bars, copyright notices, etc. The information contained in these noisy blocks can seriously harm Web data mining. Thus eliminating these noises is of great importance.An innovative idea is to use a DOM tree structure to abstract out the noises in web pages to improve the performance of mining. The aim of this work is to analyze and eliminate the noisy blocks in web documents using a DOM TREE structure. DOM Tree is constructed to determine the logical structure of a web document. This work is further extended to perform both clustering and classification for a web document using WCOATES and WCOLT algorithm respectively [7], [12]. Genetic Algorithm (GA) is used to find the optimal number of clusters in a dataset [11] and correctly assign each data point to a cluster without any prior knowledge about the data. The method provides good results, and requires a small number of iterations to converge. It is important to use these algorithms together to prune out the noises in web pages there by improve the performance of mining task significantly. This paper is organized as follows. Section II presents the related work and its issues on the topic. Section III deals with the methodology and algorithms to detect the noisy blocks in web documents. Section IV presents an algorithm to analyse the noisy blocks in web documents. Next section present conclusion and future work.

II. RELATED WORK

The problem of text clustering has been studied quite extensively in the context of text data [1], [2], [3], [12], [13] presented a method for text clustering with the use of side-information. Many forms of text databases contain a large amount of side-information or Meta information, which may be used in order to improve the clustering process. In order to design the clustering method, we combined an iterative partitioning technique with a probability estimation process which computes the importance of different kinds of side-information. We present results on real data sets illustrating the effectiveness of our approach.

C.C Agarwal et al are primarily focused on providing readers with an introduction to the area of social networks. The broad area is so vast, that it is probably not possible to cover it comprehensively in a single book [2]. The problem of social network data analytics is still in its infancy; much of the past research in this area has been based on structural analysis of social networks. Such analysis primarily uses linkage structure only in order to infer interesting characteristics of the underlying network. Most of the research in social networks is based on static networks. In adversarial networks, it is desirable to determine the analytical structure of a network in which the actors in the network are adversaries, and the relationships among the different adversaries may not be fully known.

S. Zhong et al (2005) have discussed that, in many real data mining applications, data comes as a continuous stream and presents several challenges to traditional static data mining algorithms [14]. Application examples include topic detection from a news stream, intrusion detection from continuous network traffic, object recognition from video sequences, etc. Challenges lie in several aspects: high algorithm efficiency is required in real time; huge data volume that cannot be kept in memory all at once; multiple scans from secondary storage is not desirable since it causes intolerable delays; and mining algorithms need to be adaptive since data patterns change over time. M. Asfia et al describes Visual Clustering Extractor (VCE) which gets DOM tree of input web page as its input and returns the informative content block as its output. But VCE algorithm precise gradually [15] T. Htwe et al build a DOM tree for each page and then merge it into the style tree in a top-down fashion. At a particular element node *E* in the style tree, which has the corresponding tag node *T* in the DOM tree, we check whether the sequence of child tag nodes of *T* in the DOM tree is the same as the sequence of element nodes in a style node *S* below *E* (in the style tree) [16]

III. CLUSTERING THE WEB DOCUMENT

A. Document Object Model

The Document Object Model (DOM) is an application programming interface (*API*) for valid *HTML* and well-formed *XML* documents. It defines the logical structure of documents and the way a document is accessed and manipulated. 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 as given in Fig 1 .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. For some of these tags, the closing bracket is inferred by the following tag, for example <LI> tag is closed by the following </LI> tag.

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. Fig 1, shows the example of a DOM tree of HTML web page. After creating the DOM tree, the system split it into multiple sub-trees according to threshold level. Different Web Sites have different layout and presentation style, therefore the depth of the tree of the Web page is varied according to their presentation style. 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. Regression analysis is also used to understand which among the independent variables are related to the dependent variable, and to explore the forms of these relationships

Once we obtained the threshold level, the system determine some nodes of DOM less than the threshold level as noise and remove them before classification process start.After splitting sub-trees, we transform them into numeric representation for input patterns of neural network classification model using eq.1.

Xi = Sn/Tn eq.1

Where, Sn is the number of occurrence of same leaf nodes in sub-tree, Tn be the total number of leaf nodes in sub-tree.By parsing a webpage into a DOM tree, more control can be achieved for the proposed system. Lastly, we remove the noise class in Web page and show extracted main content data in HTML page

B. The Steps of Content Extraction

Step 1 Standardizing the web page tags

a. Symbols, ”<” and ”>”, should only contain html tags.
When used in other place, they should be replaced by ”&lt” and ”&gt” respectively.
b. All tags must be matched, i.e. every starting tag has a
corresponding ending tag.
c. Attributes of all tags must be encircled by quotation
marks.
d. All tags must be nested correctly. For example, <a>. .

.<b>. . .</b>. . .</a> is a correct nest, while <a>. . .<b>. . .</a>. . .</b> is incorrect.

Step 2 Preprocessing the web page tags.
All tags on the page form a tree structure. Those nodes that do not contain any text should be removed, as well as invalid tags such as<script> <style> <form> <marquee> <meta> etc, which are unrelated to the content. Then the structure tree is built.

```
            <BODY bgcolor=WHITE
<IMG src=”picture.gif’ height=200>
<TABLE width=600 height=200>
…………
</TABLE>
</TABLE>
```

Step 3 Judging the location of content
The aim of this process is to select the optimum node containing content. If a node is not satisfied with this condition, the text under this node is not identified. As the news web page is a tree structure, the content must be under a general node

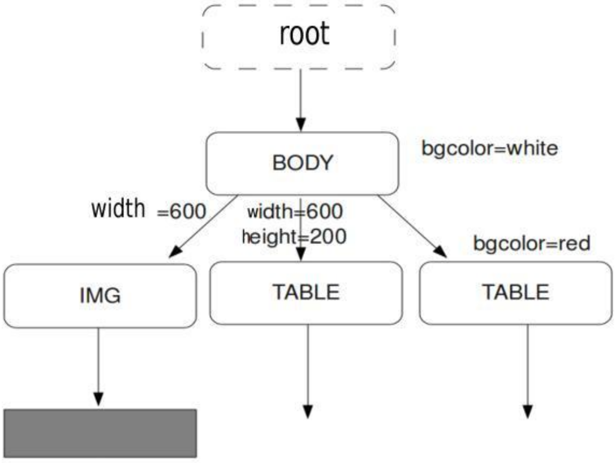

Fig 1: An example of DOM Tree of HTML web page**.**

.

Step 4 Extracting the content
The content is extracted by tools such as html parser. If the node is not satisfied with the conditions, return the step 3 in order to find the optimal nodes of the next level nodes (the child nodes of the node).

Step 5 Adjusting the extraction results from step 4
In step 3, only the node that most likely contains the content is selected. But if the structure of a web page is relatively decentralized, it is very prone to extract a section or a paragraph of the whole content. As the adjacent nodes on the same level are free of judge, in this step, we must adjust the above result. The text also should be extracted from the adjacent nodes that meet the conditions of the precise content extraction. So all text will be extracted from the qualified nodes on the same level

C. Genetic Algorithm

Genetic algorithm **(**GA) based model is proposed for frequent pattern mining in web content database using various genetic operators such as Reproduction , Crossover, mutation. This algorithm uses minimum threshold fitness as measure for selecting contents (items) as frequent. In this work it makes use of fitness function to determine the useless individuals or candidates are pruned out thereby reducing the number of candidates for next test [11]. Also this approach requires only one scan of database. Thus this model is able to address the issues of large number of candidate generation and number of database scans.

D. Fitness Function

Fitness function is a mathematical function to calculate the survivability of chromosome. Calculate percent support probability ratio of each chromosome-C or individual. Here items are treated as chromosome in the form of binary code.[11]
Where
1= indicates item sold in a particular session.
0= indicates item not sold in a particular session.

The Fitness function support is given by

$$F(s) = \frac{\text{Total number of true bit in chromosome-C}}{\text{Total length of chromosome -C}}$$

$$F(s) = TDAc / TLc$$

Where

TDAc= Total dominants alleles in chromTLc= Total length of chromosome-C

Percent support f(Sp) = f(s)*100= TDAc/TLc*100.

Relative percentage of each Individual

F(Rsp)=f(Sp)

E. WCOATES Clustering With Side Information

In this section describe algorithm for web clustering with side-information. We submit to this algorithm as WCOATES right through the paper, which communicate to the information that it is Web Content and Auxiliary attribute based Text clustering algorithm [7]. We suppose that a contribution to the algorithm is the number of clusters k. As in the case of all web-clustering algorithms, it is specified that stop-words have been eliminated, and stemming has been performed in order to improve the discriminatory power of the attributes The algorithm requires two phases as given in Fig 3

i) Initialization: We use a lightweight initialization phase in which a standard web clustering approach is used without any side-information. For this purpose, we use the algorithm described in[11] The reason for this algorithm is utilized, because it is a simple algorithm which can quickly and efficiently provide a reasonable initial opening point. The centroids and the partition produced by the clusters formed in the first phase provide an initial starting point for the second phase. We note that the first phase is based on text only, and does not utilize the supplementary data. Thus Fig 2 depicts the overall system architecture that explains each and every stage of the process. Architectural design is the conceptual model that defines the structure, behaviour and more views of a system. Intially upload the web document and perform preprocessing inorder to remove discriminatory power of the attributes. Next DOM tree structure is obtained for a web document .Use Genetic algorithm to determine the optimal number of cluster for a dataset . Then Clustering is done using WCOATES and Classification is done using WCOLT algorithm to obtain a pure web content

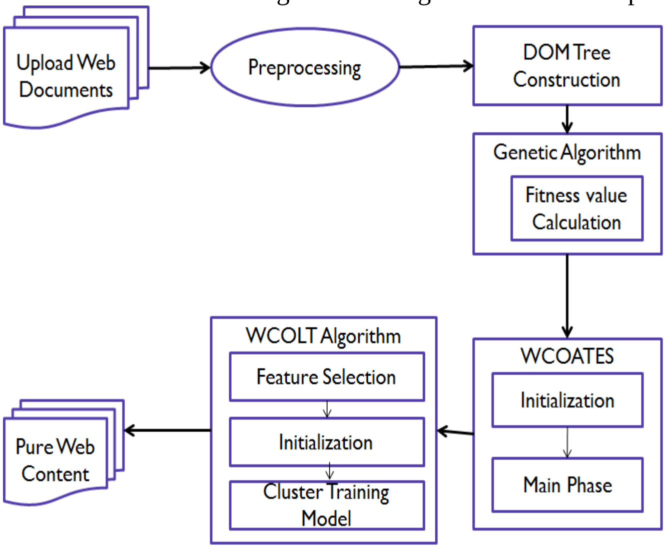

Fig 2 System Architecture

ii) Main Phase**:** The main phase of the algorithm is executed after the initial phase. This phase starts off among with these first groups, and iteratively reconstructs these clusters with the use of both the text content and the supplementary information. This phase execute alternative iterations which use the text content and auxiliary attribute information in order to improve the quality of the clustering. We call these iterations as web content iterations and web auxiliary iterations correspondingly.

The arrangement of the two iterations is referred to as a major iteration. Each major iteration thus contains two small iterations, equivalent to the subsidiary and text-based technique respectively. The focus of the first phase is simply to construct an initialization, which provides a good starting point for the clustering process based on text content. Since the key techniques for content and auxiliary information integration are in the second phase, we will focus most of our subsequent discussion on the second phase of the algorithm. The first phase is simply a direct application of the web clustering algorithm proposed. The overall approach uses alternating minor iterations of content-based and auxiliary attribute-based clustering. These phases are referring to as web content-based and web supplementary attribute- based iterations respectively

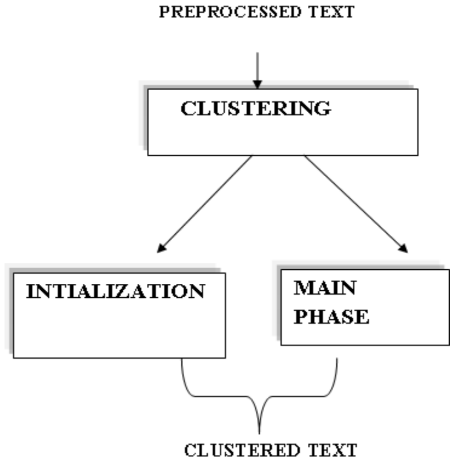

Fig 3: Clustering using WCOATES

The algorithm maintains a set of starting centroids, which are later refined in dissimilar iterations. In each web content-based phase, assign a web document to its closest seed centroid based on a text similarity function. The centroids for the k clusters created during this phase are denoted by L1 . . . Lk. Specifically; the cosine similarity function is used for consignment purpose. In every web auxiliary phase, we make a probabilistic model, which communicate the characteristic probabilities to the cluster-membership probabilities, based on the clusters which have already been created in the most recent text-based phase. The goal of this modeling is to examine the coherence of the web clustering with the side-information attributes. Before discussing the auxiliary iteration in detail we will first initiate a number of notations and definitions which help in explaining the clustering model for combining auxiliary and text variables.

We assume that the k clusters associated with the data are denoted by C1 . . . Ck. In order to construct a probabilistic model of membership of the data points to clusters, we suppose that every supplementary iteration has a prior probability of assignment of documents to clusters and a posterior probability of assignment of documents to clusters with the use of auxiliary variables in that iteration. We indicate the previous prospect that the document Ti belongs to the cluster C_j by P (T_i€C_j). Once the pure-text clustering phase has been executed, the a-priori cluster membership probabilities of the auxiliary attributes are generated with the use of the last content-based iteration from this phase. The apriori value of P (T_i€C_j) is simply the fraction of documents which have been assigned to the cluster C_j. In order to compute the posterior probabilities P(T_i€ $C_j|\overline{Xi}$) of membership of a record at the end of the auxiliary iteration, we use the auxiliary attributes $\overline{Xi}$which are related with Ti.

Consequently, we would similar to calculate the conditional probability P (T_i€ $C_j|\overline{Xi}$). We will create the estimate of allowing for only those auxiliary attributes, which take on the value of 1. Since we are focussing on sparse binary data, the value of 1 for an attribute is a much more informative event than the value of 0. Therefore, it is enough to state only on the case of attribute values taking on the value of 1. For instance, let us think an application in which the auxiliary information corresponds to users which are browsing specific web pages. In such a case, the clustering behavior is influenced much more significantly by the case when a user does browse a particular page, rather than one in which the user does not browse an exacting page, since the majority pages will usually not be browsed by a particular user. This is generally the case across many sparse data domains such as attributes corresponding to links, discredited numeric data, or categorical data which is quite often of very high cardinality. Furthermore, in order to ensure the robustness of the approach, we need to eliminate the noisy attributes.

This is especially important, when the number of auxiliary attributes is fairly huge. Therefore, at the start of every auxiliary iteration, we compute the gini-index of each attribute based on the clusters created by the last content based iteration. This gini-index provides a quantification of the discriminatory power of each attribute with respect to the clustering process. The gini-index is computed as follows. Let frj be the fraction of the records in the cluster C_j, for which the feature get on the value of 1.

```
Algorithm WCOATES
Input: (Num Clusters: k, Corpus: $T_1…T_N$, Auxiliary Attribute: $X_1…X_{N});$
Begin
Use Genetic Algorithm to create initial set of    Clusters $C_1….C_K$;
Let Centroids of $C_1….C_K$ be denoted by $L_1….L_k$;
t=1;
while not (termination criterion) do
Begin
{First minor iteration}
Use Cosine –Similarity of each document $T_i$ to Centroids $L_1….L_k$ in order to determine the closest cluster to $T_i$ and update the cluster assignments $C_1….C_K$;
```

```
Denote assigned cluster index for document $T_i$ by $q_c$ (i, t);
Update cluster centroid $L_1 \dots L_k$ to the centroids of updated clusters $C_1 \dots C_k$;
{Second minor iteration}
Compute gini-index of $G_r$ for each auxiliary attribute r with respect to current clusters $C_1 \dots C_k$;
Mark attributes with gini-index which is $\delta$ standard deviation below the mean as non-discriminatory;
{for document $T_i$ let $R_i$ be the set of attributes which take on the value of 1 and for which gini-index is discriminatory ;}
For each document $T_i$ determine the Posterior Probability $P^n(T_i$ € $C_j \mid R_i)$; Denote $q_a(i, t)$ as the cluster-index with highest Posterior Probability of assignment for document $T_i$
Update cluster centroids $L_1 \dots L_k$ with the use of Posterior Probability
t=t+1;
end
end
```

Algorithm1: WCOATES

Then, we calculate the comparative occurrence prj of the attribute r in cluster j as follows:

$$\mathbf{P_{rj}} = \frac{frj}{\sum_{m=1}^{k} frm} \quad \text{eq.2}$$

The values of p_{rj} are defined, so that they sum to 1 over a particular attribute r and dissimilar clusters j. We note that when all values of prj take on a like value of 1/k, then the attribute values are evenly distributed across the different clusters. Such an attribute is not very discriminative with respect to the clustering procedure, and it should not be utilized for clustering. While the auxiliary attributes may have a different clustering behavior than the textual attributes, it is also predictable that useful auxiliary attributes are at least somewhat related to the clustering behavior of the textual attributes. This is usually true of numerous applications such as those in which auxiliary attributes are defined either by linkage-based patterns or by user performance. On the other hand, totally noisy characteristic are unlikely to have any relationship to the text content, and will not be very effective for mining purposes. Therefore, we would like the values of prj to vary across the different clusters. We refer to this difference as skew. The stage of skew can be quantify with the use of the gini-index. The gini-index of feature is indicate by Gr, and is definite as follow:

$$G_r = \sum_{j=1}^{k} Prj \quad \text{eq.3}$$

The value of Gr lies between 1/k and 1. The additional discriminative the feature, the advanced the value of Gr.In everyiteration, we use only the auxiliary attributes for which the gini-index is above a particular threshold γ. The value of γ is picked to be 1.5 standard deviations below the mean value of the gini-index in that particular iteration. We note that since the clusters may change from one iteration to the next, and the gini-index is defined with respect to the present clusters, the values of the gini-index will also alter over the dissimilar iterations. Therefore, dissimilar auxiliary attributes may be utilize over different iterations in the clustering procedure, as the excellence of the clusters develop into additional refined, and the corresponding discriminative power of auxiliary attributes can also be computed more effectively as described in Algorithm 1

IV. WEB CLASSIFICATION

A.WCOLT Algorithm

We refer to our algorithm as the W*COLT* algorithm throughout the paper, which refers to the fact that it is a web content and auxiliary attribute-based text classification algorithm. The algorithm uses a supervised clustering approach in order to partition the data into *k* different clusters. This partitioning is then used for the purposes of classification. T he steps used in the training algorithm are as follows as in Fig 4

- Feature Selection**:** In the first step, we use feature selection to remove those attributes, which are not related to the class label [13]. This is performed both for the text attributes and the auxiliary attributes.
- Initialization**:** In this step, we use a *supervised k*means approach in order to perform the initialization, with the use of purely text content. The main difference between a supervised *k*-means initialization, and an unsupervised initialization is that the class memberships of the records in each cluster are pure for the case of supervised initialization. Thus, the *k*-means clustering algorithm is modified, so that each cluster only contains records of a particular class.

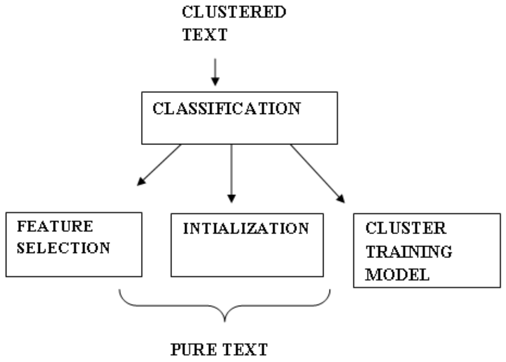

Fig 4: Classification using WCOATES

• Cluster-Training Model Construction**:** In this phase, a combination of the text and side-information isused for the purposes of creating a cluster-based model. As in the case of initialization, the purity of the clusters in maintained during this phase. The first step in the supervised clustering process is to perform the feature selection, in which only the discriminative attributes are retained. In this feature selection process, we compute the gini-index for each attribute in the data with respect to the class label.

If the gini index is γ standard deviations (or more) below the average gini index of all attributes, then these attributes are pruned globally, and are never used further in the clustering process. With some abuse of notation, we can assume that the documents *Ti* and auxiliary attributes *Xi* refer to these pruned representations. We note that this gini index computation is different from the gini-index computation with respect to the auxiliary attributes. The latter is performed during the mainphase of the algorithm. Once the initialization has been performed, the main process of creating supervised clusters with the use of a combination of content and auxiliary attributes is started.

As in the previous case, we use two minor iterations within a major iteration. One minor iteration corresponds to content-based assignment, whereas another minor iteration corresponds to an auxiliary attribute-based assignment. The main difference is that class-based supervision is used in the assignment process. For the case of content-based assignment, we only assign a document to the closest cluster centroid, which belongs to the same label.

```
Algorithm WCOLT
Input: (Num Clusters: k, Corpus: $T_1 \ldots T_N$, Auxiliary Attribute: $X_1 \ldots X_N$ , Labels $l_1 \ldots l_N$);
Begin
  Perform feature selection on text and auxiliary attributes with the use of class labels and gini index
  Use Genetic Algorithm to create initial set of C Clusters $C_1 \ldots C_K$ so that each cluster $C_i$ contains only records of a particular class
  Let Centroids of $C_1 \ldots C_K$ be denoted by $L_1 \ldots L_k$;
  t=1;
  while not(termination criterion) do
  Begin
    {First minor iteration}
    Use Cosine –Similarity of each document $T_i$ to Centroids $L_1 \ldots L_k$ in order to determine the closest cluster to $T_i$ and update the cluster assignments $C_1 \ldots C_K$;
    Denote assigned cluster index for document $T_i$ by $q_c(i,t)$;
    Update cluster centroid $L_1 \ldots L_k$ to the centroids of updated clusters $C_1 \ldots C_k$;
    {Second minor iteration}
    Compute gini-index of $G_r$ for each auxiliary attribute r with respect to current clusters $C_1 \ldots C_k$;
    Mark attributes with gini-index which is $\delta$ standard deviation below the mean as non-discriminatory.
    {for document $T_i$ let $R_i$ be the set of attributes which take on the value of 1 and for which gini-index is discriminatory;}
    For each document $T_i$ determine the Posterior Probability $P^n(T_i € C_j | R_i)$;
    Denote $q_a(i,t)$ as the cluster-index with highest Posterior Probability of assignment for document $T_i$ which also belongs to the same class;
    Update cluster centroids $L_1 \ldots L_k$ with the use of Posterior Probability
    t=t+1;
  end
end
```

Algorithm 2: WCOLT

For the case of the auxiliary minor iteration, we compute the prior probability *Pa(Ti ∈Cj)* and the posterior probability *Ps(Ti ∈Cj|Ri),*as in the previous case, except that this is done only for cluster indices which belong to the same class label. The document is assigned to one of the cluster indices with thelargest posterior probability. Thus, the assignment is alwaysperformed to a cluster with the same label, and each cluster maintains homogeneity of class distribution. As in the previouscase, this approach is applied to convergence. The overall training algorithm is illustrated in Algorithm 2.Once the supervised clusters have been created, they can be used for the purpose of classification. The supervised clusters provide an effective summary of the data which canbe used for classification purposes. Both the content- and auxiliary- information is used in this classification process

VI. CONCLUSION AND FUTUREWORK

World Wide Web is a source of information where large amount of data is stored. These web pages mainly consist of noisy data. Extracting useful information from these web pages is very complex task. Extra information from web pages like header footer, advertisements, Navigational Bars which are called noisy data. These noisy data are removed to extract the main content. A DOM Based Page Segmentation method is proposed for noise reduction and extraction of Web content from Web Pages. The navigational bar, Home page and short description noise is removed using Dom based page segmentation which convert the Web Pages into blocks and regions. Finally it removes the noise and extract the information based on regions and blocks.In future work, we can eliminate noisy web videos from web pages using the video processing techniques.

REFERENCES

[1] C. C. Aggarwal and H. Wang, Managing and Mining Graph Data. New York, NY, USA: Springer, 2010.

[2] C. C. Aggarwal, Social Network Data Analytics. New York, NY, USA: Springer, 2011.

[3] C. C. Aggarwal and C.-X.Zhai, Mining Text Data. New York, NY, USA: Springer, 2012.

[4] C. C. Aggarwal and C.-X.Zhai, "A survey of text classification algorithms," in Mining Text Data.New York, NY, USA: Springer, 2012.

[5] C. C. Aggarwal and P. S. Yu, "A framework for clustering massive text and categorical data streams," in Proc. SIAM Conf. Data Mining, 2006, pp. 477–481.

[6] C. C. Aggarwal, S. C. Gates, and P. S. Yu, "On using partial supervision for text categorization," IEEE Trans. Knowla. Data Eng., vol. 16, no. 2, pp. 245–255, Feb. 2004.

[7] C. C. Aggarwal and P. S. Yu, "On text clustering with side information," in Proc. IEEE ICDE Conf., Washington, DC, USA, 2012.

[8] R. Angelova and S. Siersdorfer, "A neighborhood-based approach for clustering of linked document collections," in Proc. CIKM Conf., New York, NY, USA, 2006, pp. 778–779.

[9] A. Banerjee and S. Basu, "Topic models over text streams: A study of batch and online unsupervised learning," in Proc. SDM Conf., 2007, pp. 437–442.

[10] J. Chang and D. Blei, "Relational topic models for document networks," in Proc. AISTASIS, Clearwater, FL, USA, 2009, pp. 81–88.

[11] Vikrant Sabnis, R. S. Thakur "GA Based Model for Web Content Mining,"in IJCSI International Journal of Computer Science Issues, Vol. 10, Issue 2, No 3, March 2013

[12] Charu C Aggarwal, Yuchen Zhao and Philip S.Yu "On the Use of side information For Mining Text Data" in IEEE Transactions on Knowledge and Data Engineering,VOL 26,NO 6 JUNE 2014.

[13] Tao Liu and Shengping "An Evaluation on Feature Selection For Text Clustering" in ICML International Conference on Machine Learning,Washington DC -2003

[14] S.Zhong Streaming text clustering",Neural Network vol 18 , no:5-6.pp.790-798,2005.

[15] M. Asfia, M. M. Pedram and A. M. Rahmani, Main Content Extraction from Detailed Web Pages. International Journal of Computer Applications- 2010

[16] T. Htwe, N. S. M. Khan, Extracting Data Region in Web page by Removing Noise using DOM and Neural Network. ICIFE- 2011.