Middle-East Journal of Scientific Research 23 (7): 1521-1531, 2015
ISSN 1990-9233

DOI: 10.5829/idosi.mejsr.2015.23.07.22382

An Enhanced Topic Driven Clustering for Document Based Datasets

Veeramalai Sankaradass, P. Meenakshi, K. Karthick and R. Danu

Department of Computer Science and Engineering,
Vel Tech High Tech Dr. Rangarajan Dr. Sakunthala Engineering College, Avadi, Chennai, India

Abstract: In the web pages along with the text documents side information are also available. Such side-information may be of different kinds such as links, user access behavior and non textual attributes. Such attributes may contain a tremendous amount of information for clustering purposes. In the existing system approach it removes all the side information. But sometimes some side information can be useful and relevant for the given topic. So we need a principled way to perform the mining process, so as to maximize the advantages from using this side information. To overcome this drawback, in the proposed system we employ COATES(Content and Auxiliary attribute based Text Clustering) algorithm. We present this method for mining text data with the use of side information. We can retain the relevant and useful side information and remove the unwanted irrelevant noises efficiently. This helps in magnifying the clustering effects. In the modification part in addition to the COATES algorithm OSKM approach is used to make browser updation.

Key words: Clustering · Text mining · Dataset · COATES · OSKM

INTRODUCTION

In text clustering, a text or document is always represented as a bag of words. This representation raises one severe problem: the high dimensionality of the feature space. Obviously, a single document has a sparse vector over the set of all terms. The performance of clustering algorithms will decline dramatically due to the problems of high dimensionality and data sparseness [1]. Therefore it is highly desirable to reduce the feature space dimensionality. There are two commonly used techniques to deal with this problem: feature extraction and feature selection. In many real data mining applications, data comes in as a continuous stream and presents several challenges to traditional static data mining algorithms [2-4]. Application examples include topic detection from a news stream, intrusion detection from continuous network traffic, object recognition from video sequences [5], 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 [6, 7, 8] need to be adaptive since data patterns change over time. In many application domains, a tremendous amount of side-information is also associated along with the documents [9, 10]. This is because 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. Sometimes this side information can be useful. But there is no efficient method to remove only the unwanted and retain the useful information. To overcome this we can employ COATES approach. This helps to retain only the useful side information and removes the noisy information. This helps in magnifying the clustering effects. In addition to that OSKM is also used to make browser updating.

Existing System: While such side-information can sometimes be useful in improving the quality of the clustering process, it can be a risky approach when the side-information is noisy [3, 4, 11]. In such cases, it can actually worsen the quality of the clustering. Therefore, we will use an approach which carefully ascertains the coherence of the clustering characteristics of the side information with that of the text content. This helps in magnifying the clustering effects of both kinds of data.

Corresponding Author: Veeramalai Sankaradass, Department of Computer Science and Engineering, Vel. Tech. High Tech. Dr. Rangarajan Dr. Sakunthala Engineering College, Avadi, Chennai, India.

The core of the approach is to determine a clustering in which the text attributes and side-information provide similar hints about the nature of the underlying clusters and at the same time ignore those aspects in which conflicting hints are provided.

Disadvantages:

- Removes all the side information
- Relative importance's of side information is difficult to cluster when information is complex

Proposed System: In this section, we will describe our algorithm for text clustering with side-information. We refer to this algorithm as COATES throughout the paper, which corresponds to the fact that it is Content and Auxiliary attribute based Text clustering algorithm. We assume that an input to the algorithm is the number of clusters *k*. As in the case of all text-clustering algorithms, it is assumed that stop-words have been removed and stemming has been performed in order to improve the discriminatory power of the attributes. The algorithm requires two phases: 1.Initialization 2. Main Phase. In addition to this OSKM is used to perform browser updation automatically.

Advantages of proposed system:

- Irrelevant noises are removed efficiently.
- Improves efficiency of clustering process.
- Side information is utilized efficiently.

The complete system is described in different process based modules. They are 1. Text mining 2. Combining auxiliary attributes 3. Attribute based clustering 4. Checking cluster purity. All different modules are technically defined and described in the following sections with necessary diagrams

Literature Survey: We will examine the problem of clustering massive domain data streams. Massive-domain data streams are those in which the number of possible domain values for each attribute are very large and cannot be easily tracked for clustering purposes. Some examples of such streams include IP-address streams, credit-card transaction streams, or streams of sales data over large numbers of items. In such cases, it is well known that even simple stream operations such as counting can be extremely difficult because of the difficulty in maintaining summary information over the different discrete values. The task of clustering is significantly more challenging in such cases, since the intermediate statistics for the different clusters cannot be maintained efficiently.

Here, Charu C. Aggarwal proposes a method for clustering massive-domain data streams with the use of sketches. Charu C. Aggarwal proves probabilistic results which show that a sketch-based clustering method [1] can provide similar results to an infinite space clustering algorithm with high probability. The problem of massive-domain clustering naturally occurs in the space of discrete attributes, whereas most of the known data stream clustering methods is designed on the space of continuous attributes. We will propose a sketch-based approach in order to keep track of the intermediate statistics of the underlying clusters. These statistics are used in order to make approximate determinations of the assignment of data points to clusters [12]. We provide probabilistic results which indicate that these approximations are sufficiently accurate to provide similar results to an infinite-space clustering algorithm with high probability. We also present experimental results which illustrate the high accuracy of the approximate assignments. In the next section, we will propose a technique for massive-domain clustering of data streams. We provide a probabilistic analysis which shows that our sketch-based stream clustering method provides similar results to an infinite space clustering algorithm with high probability.

In many real data mining applications, data comes in as a continuous stream and presents several challenges to traditional static data mining algorithms [6, 8]. 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 [3] 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.

Shi Zhong combines an efficient online spherical k-means (OSKM) algorithm with an existing scalable clustering strategy to achieve fast and adaptive clustering of text streams [15,7]. The OSKM algorithm modifies the spherical k-means (SPKM) algorithm, using online update (for cluster centroids) based on the well-known Winner-Take-All competitive learning. It has been shown to be as efficient as SPKM, but much superior in clustering quality. The scalable clustering strategy was previously developed to deal with very large data bases

that cannot fit into a limited memory and that are too expensive to read/scan multiple times. Using the strategy, one keeps only sufficient statistics for history data to retain (part of) the contribution of history data and to accommodate the limited memory. To make the proposed clustering algorithm [13, 14] adaptive to data streams, we introduce a forgetting factor that applies exponential decay to the importance of history data. The older a set of text documents, the less weight they carry. Our experimental results demonstrate the efficiency of the proposed algorithm and reveal an intuitive and an interesting fact for clustering text streams—one needs to forget to be adaptive.

Feature selection is a process that chooses a subset from the original feature set according to some criterions. The selected feature retains original physical meaning and provides a better understanding for the data and learning process [15]. Depending on the class label information feature selection can be either unsupervised or supervised.

System Architecture: The proposed system for is designed and focused here. The objective of this proposed system is that.

- Efficient use of side information.
- Quality of clustering process is improved.
- Irrelevant noises, advertisement links are removed.
- Browser updation can be made.

In the proposed work our objective will be firstly to collect information i.e. retrieving the different kinds of attributes for text clustering (side information of a particular page). After this Text based clustering will be performed. In the text based clustering we will cluster all the retrieved information using the COATES algorithm. Then the Text classification is carried out, means classifying the clustered text for generating the optimized result according to User Behavior (localization, personalization). Then the output will be shown in the form of graph. Graphical representation will show the relevant data mined from the particular page by removing the irrelevant information. Also the analytical mined reports will be generated. These reports will depend on the previous searching method and the method used in our proposed system

- Collection of Information In collection of information we will retrieve the different kinds of attributes for text clustering. These different attributes are the side information of a particular page.
- Text based clustering; we will cluster all the retrieved information using the COATES algorithm
- Text classification classifying the clustered text for generating the optimized result according to the user behavior (localization, personalization). User behavior can be predicted by using the cookies stored in the browser and the user's login details i.e. the location, interests, etc.
- Outcome, we will show the graph that will show the relevant data mined from the particular page by removing the irrelevant information.

In our proposed system first of all we will take any document as input, also we will extract the side information from the document. After this we will apply COATES algorithm on it which is iterative clustering process. After the COATES algorithm is applied we will get the output in the form of clusters. Once the clusters are formed we can do searching on the mined data with the help of user behavior (localization, personalization). By user behavior we mean that what the user wants to search exactly. User behavior can be predicted by using the cookies stored in the browser and the user's login details i.e. the location, interests, etc. Figure shows the system flow of the proposed system.

Text Mining: We use the Content and Auxiliary attribute based Text clustering (COATES) algorithm in text clustering. This algorithm has two phases. Now, this text-mining is the initialization phase of the algorithm. We use a lightweight initialization phase in which a standard text clustering approach is used in weblog without any side-information. The centroids and the partitioning [2] created 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 use the auxiliary information. 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.

Text Mining Process: The complete process involved in text mining is given as flow process in the following diagram which describes the each and module practically involved in the process. In all the way, we need to have complete knowledge about the whole process in depth. Most of the time, we are confusing in the process flow. To make it clear, we have designed and given below the flow process.

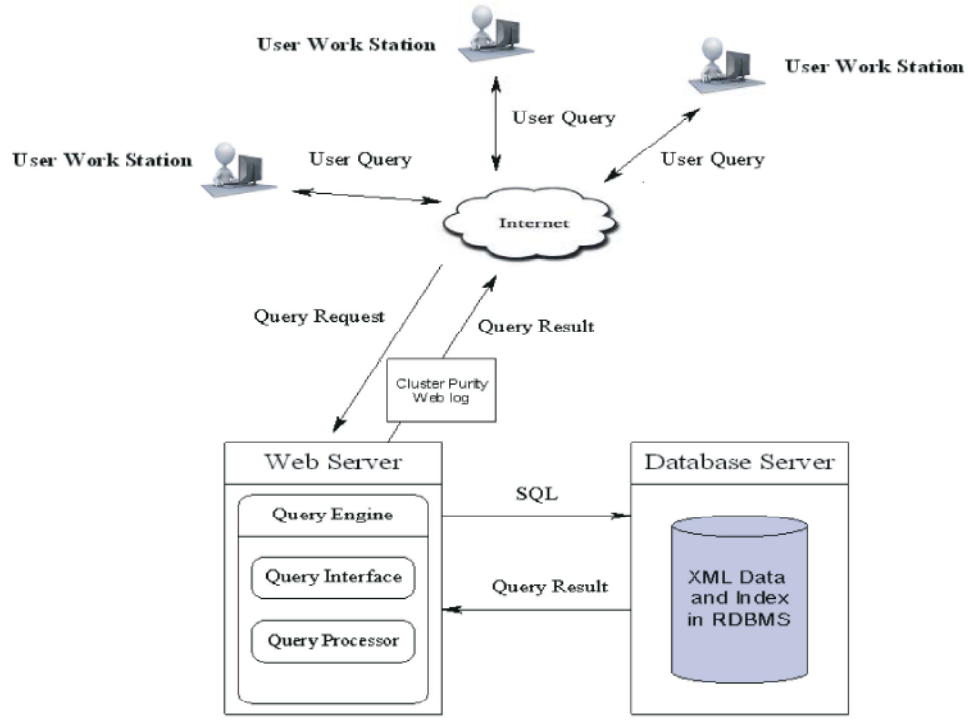

Fig. 1: System Architecture

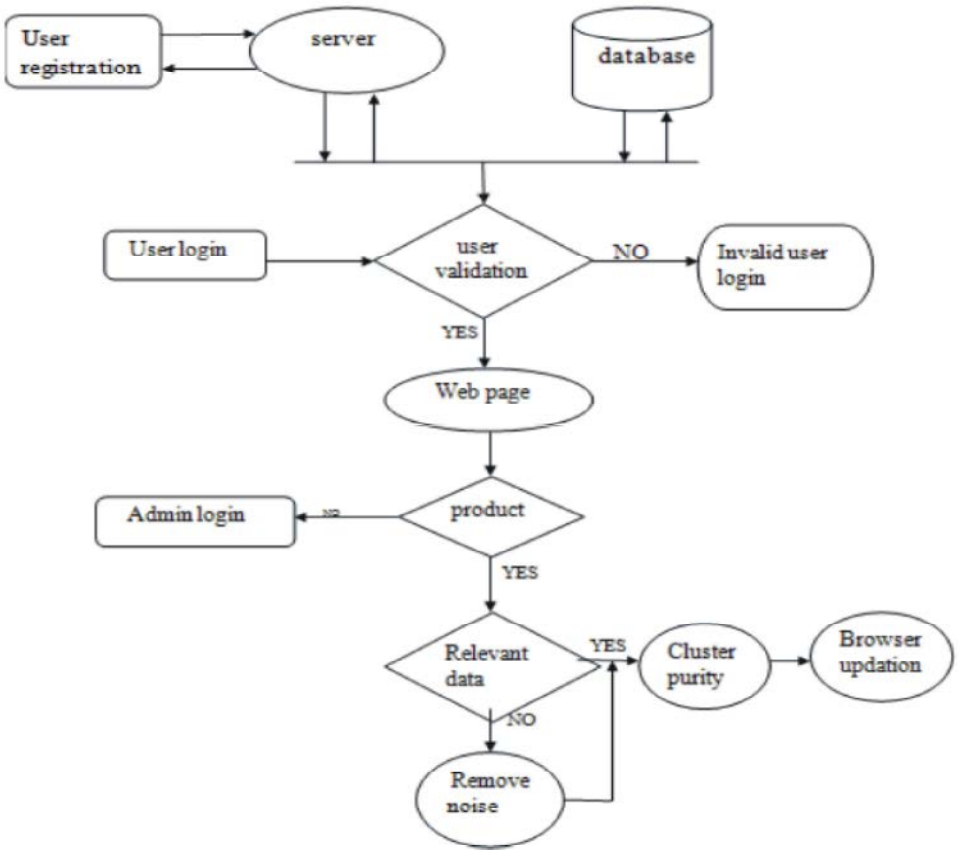

Fig. 2: Data flow diagram of proposed System

Fig. 3: Block Diagram of Text Mining Process

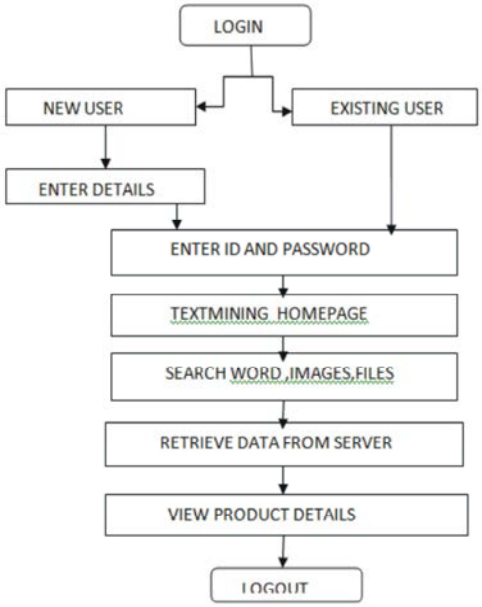

Fig. 4: Flow Process of Text Mining Process

Combining Auxillary Attribute: The main phase of the algorithm is executed after the first phase. The main phase is the Combining auxiliary attributes in the text clustering. This phase starts off with these initial groups and iteratively reconstructs these clusters with the use of both the text content and the auxiliary information. This phase performs alternating iterations which use the text content and auxiliary attribute information in order to improve the quality of the clustering. We call these iterations as content iterations and auxiliary iterations respectively. The combination of the two iterations is referred to as a major iteration. Each major iteration thus contains two minor iterations, corresponding to the auxiliary and text-based methods respectively.

Attribute Based Text Clustering: The overall approach uses alternating minor iterations of content-based and auxiliary attribute-based clustering. The algorithm maintains a set of seed centroids, which are subsequently refined in the different iterations. In each content-based phase, we assign a document to its closest seed centroid based on a text similarity function. In each auxiliary phase, we create a probabilistic model, which relates the attribute 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 text clustering with the side-information attributes. In order to construct a probabilistic model of membership of the data points to clusters, we assume that each auxiliary iteration has a prior probability of assignment of documents to clusters (based on the execution of the algorithm so far) and a posterior probability of assignment of documents to clusters with the use of auxiliary variables in that iteration. 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 auxiliaryattributes is quite large.

Checking Cluster Purity: For each class, we computed the cluster purity, which is defined as the fraction of documents in the clusters which correspond to its dominant class. The average cluster purity over all clusters (weighted by cluster size) was reported as a surrogate for the quality of the clustering process. Let the number of data points in the k clusters be denoted by n1 . . . nk. We denote the dominant input cluster label in the k clusters by l1 . . . lk. Let the number of data points with input cluster label li be denoted by ci. Then, the overall cluster purity P is defined by the fraction of data points in the clustering which occur as a dominant input cluster label. Therefore, we have: P=Ci/Ni. The cluster purity always lies between 0 and 1. Clearly, a perfect clustering will provide a cluster purity of almost 1, whereas a poor clustering will provide very low values of the cluster purity.

Proposed Approaches: In this work, we proposed COATES(Content and Auxiliary attribute based Text Clustering) algorithm. We present this method for mining text data with the use of side information. We can retain the relevant and useful side information and the unwanted irrelevant noises are removed efficiently. This helps in magnifying the clustering effects. In the modification part in addition to the COATES algorithm OSKM approach is used to make browser updation.

Coates: We will use the COATES for doing the text clustering with the help of side information. COATES is the abbreviation of COntent and Auxiliary attribute based Text clustering algorithm. The input to this algorithm will

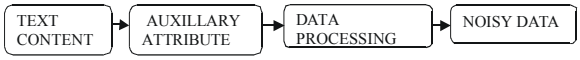

Fig. 5: Block Diagram for Combining Auxillary Attribute

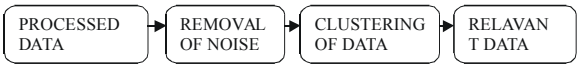

Fig. 6: Block Diagram for Attribute Based Clustering

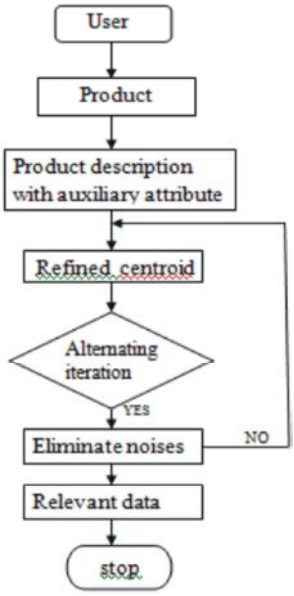

Fig. 7: Flow Process for Attribute Based Clustering

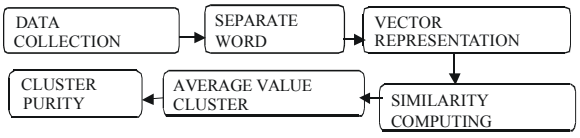

Fig. 8: Block Diagram for Checking Cluster Purity

be the numbers of clusters say k. For applying the COATES algorithm it is necessary that the stop words are removed and stemming has been performed. The algorithm works in two phases:

Initialization: This is the first phase of the COATES algorithm. In the first phase pure text clustering is performed without using any kind of side information or the auxiliary information.

Main Phase: This phase is executed after the completion of the first phase. The work of the main phase is to perform the alternating iterations with the help of the text content and the auxiliary attribute information. Thus this will improve the quality of clustering.

Oskm: It is a vector space model. Here clustering is carried out on the basis of cosine dissimilarity. It represents documents (queries) by vectors of term weights.

It is of more use in case of microarray data.K means algorithm with cosine similarity is a popular method for clustering high dimensional text data. In this each document as well as each cluster mean is represented as a high dimensional unit-length vector. This is mainly used in batch mode.(ie) each cluster mean vector is updated, only after all document vectors being assigned, as the (normalized) average of all document vectors to that cluster.

Each cluster centroids is incrementally updated given in a document. It can achieve significantly better clustering results than the batch version, especially when an annealing-type learning rate schedule is used.

Proposed Algorithm:

Algorithm 1: Query Construction

Data: Q = {Q1,Q2, . . . ,} is the set of previous queries executed on Fi.

Result: Qoneis the query of One-Query
begin
σone←- 0
forQ ← Q do
σone<←- σone←-σQ
Aone←- AFi
?Ar(Fi)
Qone←- GenerateQuery(Aone,σone)

Q is the set of queries in which the condition extractor extracts the specified the condition and then the query processor process the data.

The result is the report of the specific condition

Steps Involved in the Proposed Approach: The function Generate Query is to generate the database query based on the given set of projection attributes A one with selection expression σ one.

The basic idea of this algorithm is based on a simple property.

The attribute as with a data instance d, given two conditions:

s1: As = a1,
s2: As = a2 and a1 = a2, if s1 is satisfied, then s2 must be satisfied.

When the system receives the result of the query Qone from the database engine, it calls the second algorithm of One-Query to find the best query condition

RESULTS AND DISCUSSIONS

Our new novel proposed work is clearly analyzed, designed and implemented. We have have taken more care in the performance of the system in mining side information where we incorporated text mining concept, noise removing, clustering the relevant side text information and analyzing the purity of the cluster and also the browser updation. To mine the text data, the proposed COATES algorithm giving max of above 90.7% relevancy in the retrieved side information. in addition to the COATES algorithm OSKM approach is used to make smooth and complete browser updation.

In the above Figure 9 and Figure 10, we made very clear that how the more relevant text data are mined with max relevancy and in which how the noisy information are indentified for the further process. The practically implemented screen shot is given for clear understanding.

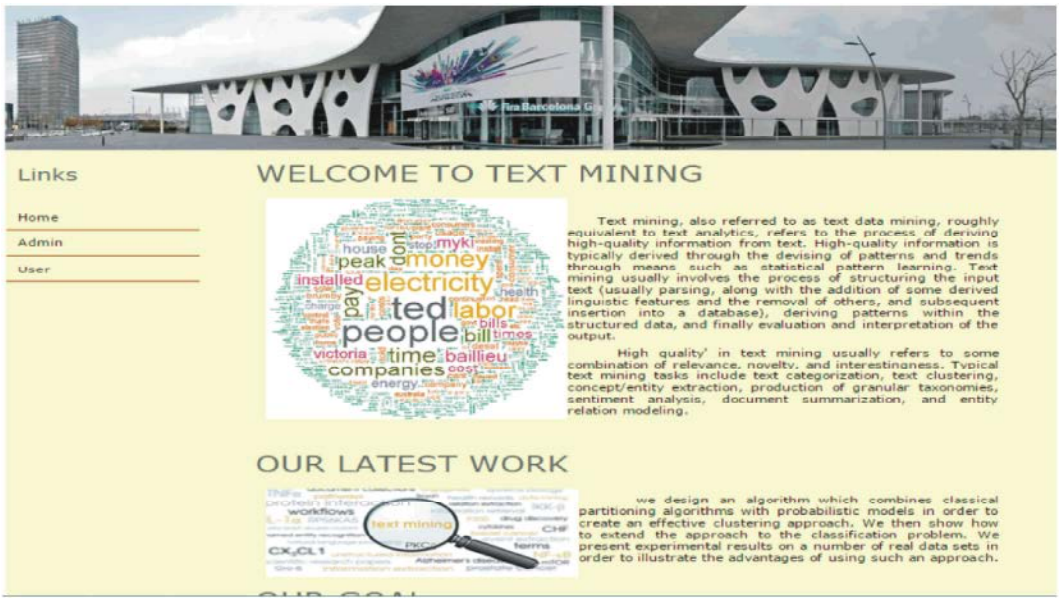

Fig. 9: Textmining

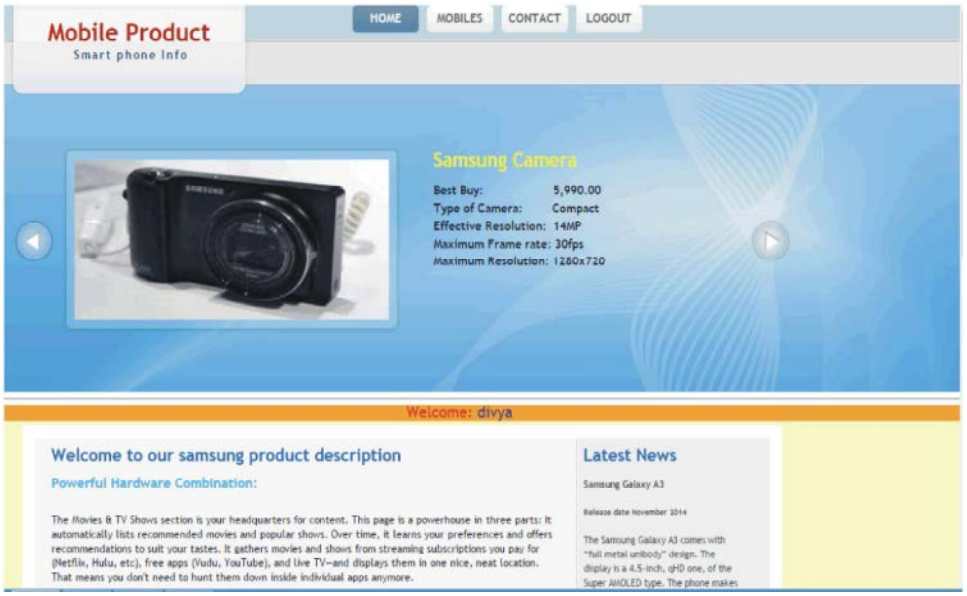

Fig. 10: Noisy Information

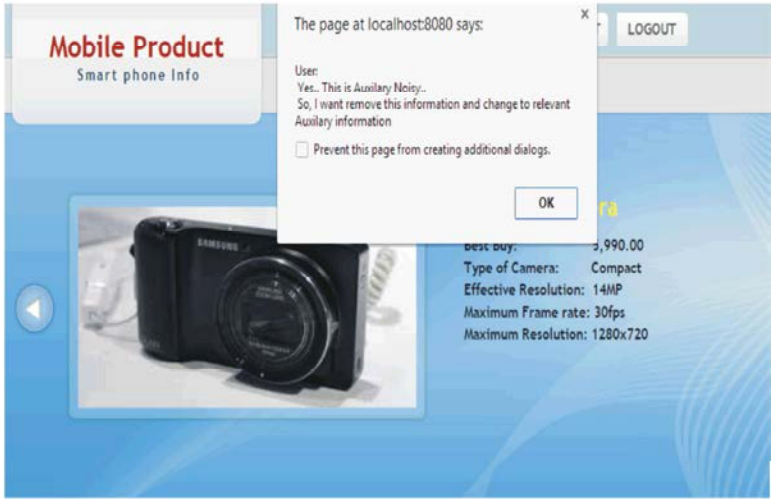

Fig. 11: Removal of Noisy Information

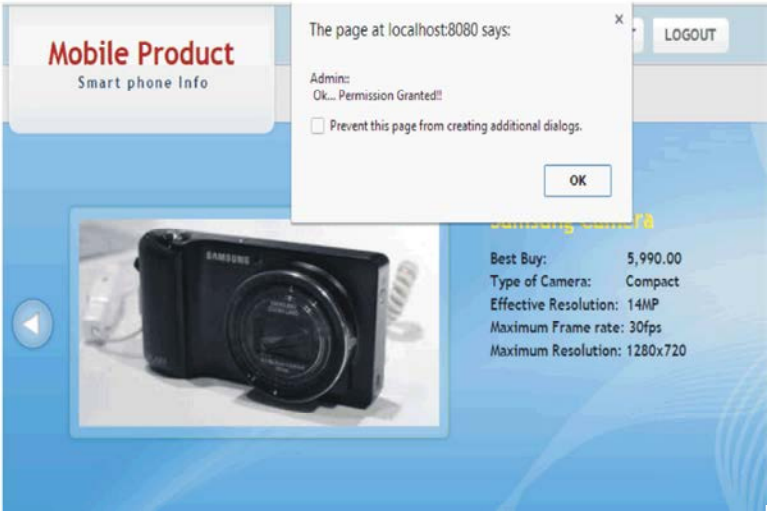

Fig. 12: Admin Permission Granted

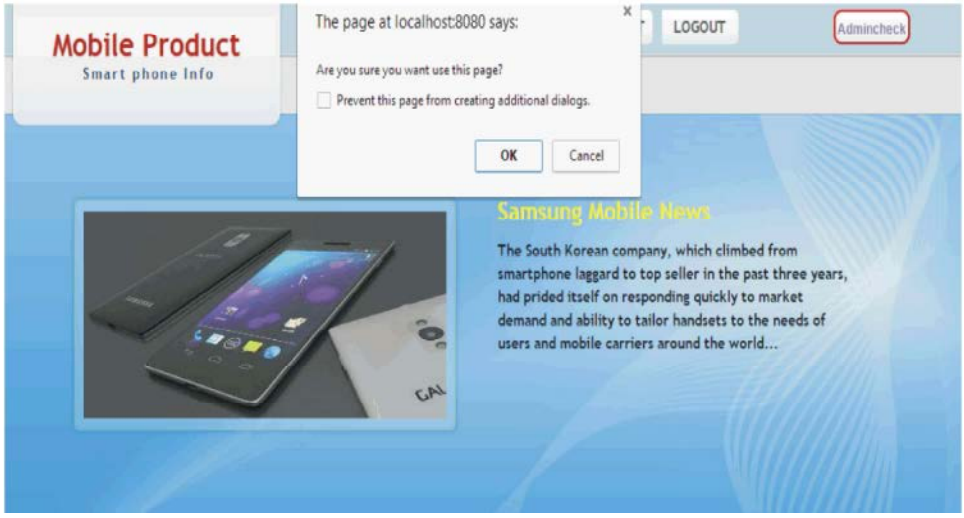

Fig. 13: Relevant Data Extraction

Fig. 14: Checking Cluster Purity

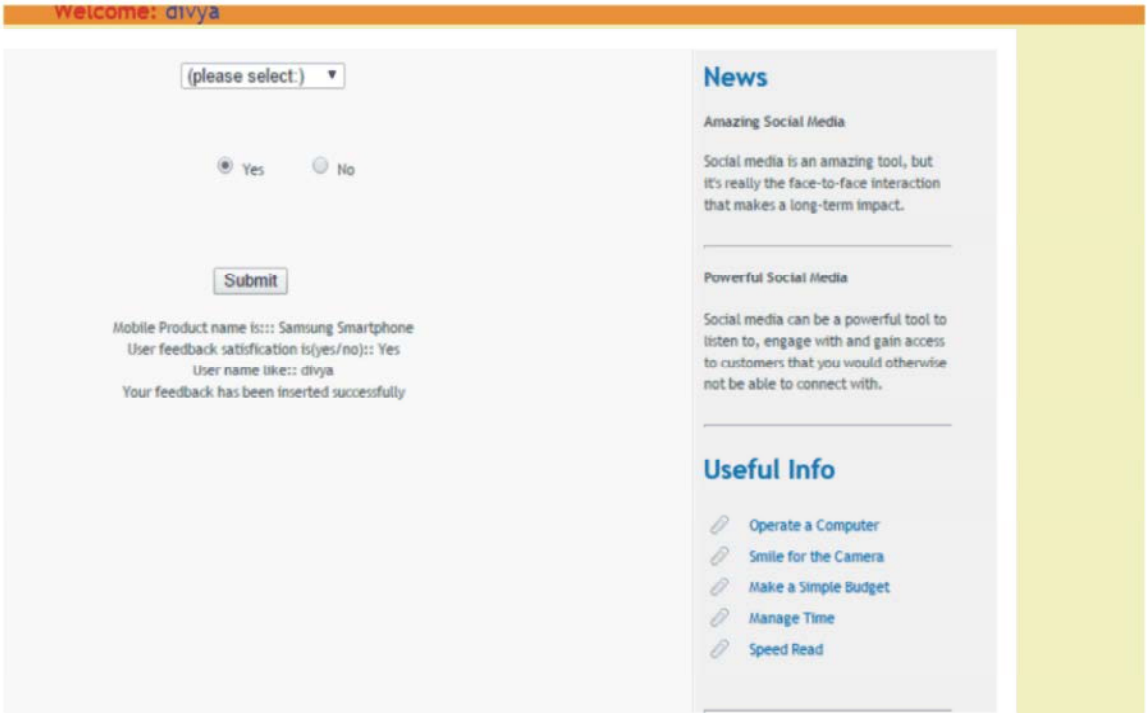

Fig. 15: Browser Updation

As a project, only authorized peoples only will be allowed to access the features. The above Figure 11 and Figure 12 shows the noisy removal techniques admin permission granted details. Even the internal text information features also defined properly to proceed further.

In the data mining and text mining, the relevancy of the same is very important measure. In the above screenshot Figure 13 and Figure 14 gives sufficient detailed information to measure the relevancy of the retrieved data and the practical guidance to check the cluster purity.

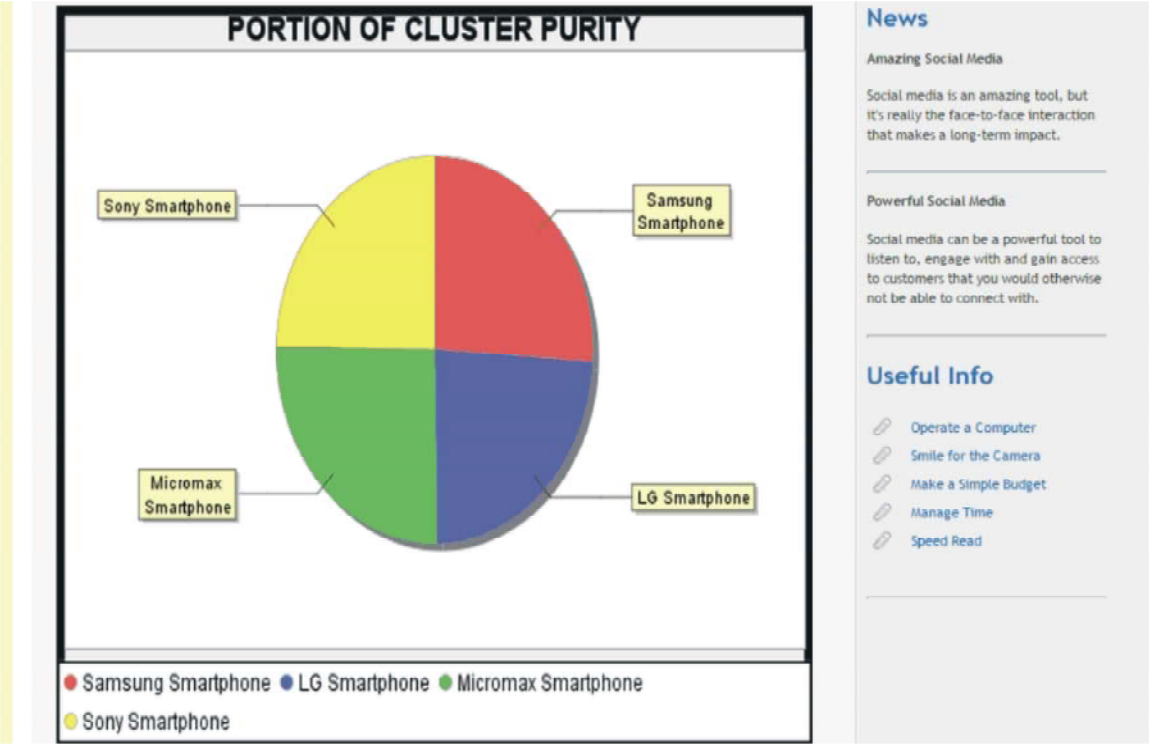

Fig. 16: Cluster Purity

Finally, the browser updation is implemented and analyzed the cluster purity. The above Figure 15 and Figure 16 shows clearly that how the browser updation will be done practically and analysis of the browser updation as well as cluster purity. It is also cleared that how it is sampled and purity of the cluster is analyzed.

CONCLUSION

In this paper, we presented methods for mining text data 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. This general approach is used in order to design both clustering and classification algorithms. We present results on real data sets illustrating the effectiveness of our approach. The results show that the use of side-information can greatly enhance the quality of text clustering and classification, while maintaining a high level of efficiency.

REFERENCES

1. Aggarwal, C.C. and P.S. Yu, 2006. "A framework for clustering massive text and categorical data streams," in Proc. SIAM Conf. Data Mining, pp: 477-481.
2. Banerjee, A. and S. Basu, 2007. "Topic models over text streams: A study of batch and online unsupervised learning," in Proc. SDM Conf., pp: 437-442.
3. Chang, J. and D. Blei, 2009. "Relational topic models for document networks," in Proc. AISTASIS, Clearwater, FL, USA, pp: 81-88.
4. Cutting, D., D. Karger, J. Pedersen and J. Tukey, 1992. "Scatter/Gather: A cluster-based approach to browsing large document collections,"in Proc. ACM SIGIR Conf., New York, NY, USA, pp: 318-329.
5. Zhong, S., 2005. " Efficient online spherical k-means clustering" Proceedings of the International Joint Conference on Neural Networks, 5: 3180-3185.
6. Aggarwal, C.C. and H. Wang, 2013. Managing and Mining Graph Data. New York, NY, USA: Springer, 2013.
7. Aggarwal, C.C., 2011. Social Network Data Analytics. New York, NY, USA: Springer, 2011.

8. Aggarwal, C.C. and C.X. Zhai, 2012. Mining Text Data. New York, NY, USA: Springer, 2012.
9. Aggarwal, C.C., S.C. Gates and P.S. Yu, 2014. "On The Use Of Side Information For Mining Text Data," IEEE Trans. Knowl. Data Eng., 16(2): 245-255.
10. Aggarwal, C.C. and P.S. Yu, 2012. "On text clustering with side information," in Proc. IEEE ICDE Conf., Washington, DC, USA.
11. Dhillon, I., 2001. "Co-clustering documents and words using bipartite spectral graph partitioning," in Proc. ACM KDD Conf., New York, NY, USA, pp: 269-274.
12. Domingos, P. and M.J. Pazzani, 1997. "On the optimality of the simple Bayesian classifier under zero-one loss," Mach. Learn., 29(2-3): 103-130.
13. Angelova, R. and S. Siersdorfer, 2006. "A neighborhood-based approach for clustering of linked document collections," in Proc. CIKMConf., New York, NY, USA, pp: 778-779.
14. Aggarwal. C.C. and C.X. Zhai, 2012. "A survey of text classification algorithms," in Mining Text Data. New York, NY, USA: Springer, 2012.
15. Tao Liu, Shengping Liu and Zheng Chen, 2003. "An Evaluation on Feature Selection for Text Clustering" Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003), Washington DC.