www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 4 Issue 3 March 2015, Page No. 10647-10651

Implementation of Data Clustering With Meta Information Using Improved K-Means Algorithm Based On COATES Approach

Mr. Patankar Nikhil S., Prof.P.P.Rokade
S. N. D. College of Engineering & Research Center,
Yeola – 423401, (M.S.), India
e-mail:nikhil.patankar1991@gmail.com
S. N. D. College of Engineering & Research Center,
Yeola – 423401, (M.S.), India
e-mail:prakashrokade2005@gmail.com

Abstract— **In many text mining applications, such as Scientific Research Publication data, Internet Movie Database, etc. as meta-information or side-information is linked with the text documents collection. It is observed that, such attributes may contain a tremendous amount of information for clustering purposes. However, the relative importance of this side-information may be difficult to estimate, especially when some of the information is noisy. Additionally, it can be risky to incorporate side- information into the mining process, because it can either improve the quality of the representation for the mining process, or can add noise to the process. Therefore, this paper explores way to perform the mining process, so as to maximize the advantages from using this side information in text mining applications with the use of COntent and Auxiliary attribute based TExt clustering Algorithm (COATES) approach which combines classical partitioning algorithms with probabilistic models in order to create an effective clustering approach along with its extension to the classification problem**

Keywords—Data Mining; Clustering; Classification; Text Mining.

I. INTRODUCTION

In numerous content mining applications, for example, Scientific Research Publication information, Internet Movie Database, and so on as meta- data or side-data is connected with the content records gathering. This side-data may be accessible in different structures, for example, archive provenance data, the connections in the record to perform route, client access rights and conduct from web logs, and other non-text based characteristics which are inserted into the content report which are distributed with reports. It is watched that, such properties may contain a colossal measure of data for grouping purposes. Then again, the relative essentialness of this side-data may be hard to gauge, particularly when a portion of the data is loud. Furthermore, it can be hazardous to join side-data into the mining procedure, on the grounds that it can either enhance the nature of the representation for the mining process, or can add commotion to the methodology. Thusly, we proposed a methodical approach to perform the mining process, in order to augment the points of interest from utilizing this side data as a part of content mining applications. In this proposed venture, there is plan to utilize COATES (Content and Auxiliary trait based Text grouping Algorithm) approach which consolidates traditional apportioning calculations with probabilistic models keeping in mind the end goal to make a powerful bunching approach alongside its expansion to the order issue. The current COATES methodology is focused around utilization of directed K-means Clustering Algorithm alongside Gini Index Computation and subsequently, in proposed task, we will use the Improved K-Means Clustering Algorithm to watch the execution of the proposed framework against standard dataset like Cora (Scientific Publication in Computer Science Domain), IMDB (Internet Movie Data Base), and so on.

II. RELATED WORK

The issue of content bunching has been considered broadly by the database group as depicted in [1]. The real center of this work was found on adaptable bunching of multidimensional information of diverse sorts. The issue of grouping has additionally been mulled over broadly in the connection of content information [2] [3]. An overview of content bunching is completed and discovered a standout amongst the most extraordinary procedures for content grouping is the disperse accumulate system, which utilizes a mix of agglomerative and divided grouping. Other related systems for content grouping which utilize comparative routines are distinguished [5] [6]. Co-bunching strategies for content information are proposed in existing examination field. Additionally, an Expectation Maximization (EM) system for content grouping has been proposed in prior exploration work. Grid factorization systems for content grouping are contemplated and this procedure chooses words from the archive focused around their significance to the bunching process, and uses an iterative EM technique to refine the bunches. A nearly related region is that of point displaying, occasion following, and content classification. In this setting, a technique for point driven grouping for content information has been proposed [4]. And in addition numerous systems for content grouping in the connection of catchphrase extraction are considered. Indeed,

quantities of useful apparatuses for content bunching are additionally accessible in late research work [7] [8]. The issue of content grouping has likewise been concentrated on in connection of versatility. Then again, these routines are intended for the instance of unadulterated content information, and don't work for cases in which the content information is joined with different manifestations of information. Some constrained work has been carried out on bunching content in the connection of system based linkage data; however this work is not relevant to the instance of general Meta data characteristics[9] [10].

III. OBJECTIVE SET

General Objective of this system is

- To Implement Data Clustering with Meta Information based on COATES Approach.

Further, this main objective can be further classified into sub-objectives like

- To optimize side information use in text mining applications.
- To combine classical partitioning algorithms with probabilistic models in order to create an effective clustering approach.
- To utilize the Enhanced K-Means Clustering Algorithm to observe the performance of the proposed system.

IV. SYSTEM MODEL

The overall proposed system architecture, breakdown structure and mathematical modeling details are covered in this section.

A. System Architecure

The following Fig.1 shows the system architecture for the proposed dissertation work based on introduced dissertation idea in introduction Section. The breakdown structure mainly focuses on following modules and their details are explained in subsequent section.

B. Break Down Structure

1. Module 1: Data Collection
2. Module 2: Data Preprocessing
3. Module 3: Application of COATES Approach
4. Module 4: Extension to Classification
5. Module 5: Performance Evaluation

The details of each module are introduced in the following section.

Module 1: Data Collection

It will be used to collect the input data required for the further processing. It will be possible to generate either Synthetic or to collect standard data set for the proposed system.

Module 2: Data Pre-processing

It will be applied to get preprocessed data for further system execution based on Data Preprocessing Techniques available in Data Mining. Common Techniques used in data preprocessing are Data Cleaning, Data Integration, Data Transformation, Data Reduction and Data Discretization which are explained in short in below section.

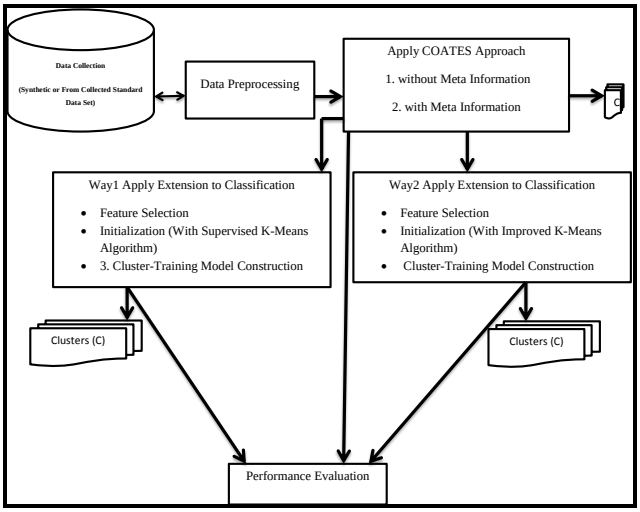

Fig.1.System Architecture

Data cleaning

It is done with the help of fill in missing values, smooth noisy data, identify or remove outliers, and resolve inconsistencies in the input data.

Data integration

It is performed using multiple databases, data cubes, or files.

Data transformation

It does normalization and aggregation of the input data.

Data reduction

It is used for reducing the volume but producing the same or similar analytical results.

Data discretization

It is part of data reduction and replaces numerical attributes with nominal ones.

Module 3: Application of COATES Approach

It will give the clusters for the input data with the possible sub modules application like Without Meta Information and With Meta Information for the Provided Input Data Document Collection.

Module 4: Extension to Classification

It contains two ways for refinement of Clusters i.e. use of Supervised K-Means Clustering Algorithm in initial case and with use of Improved K-Means Clustering Algorithm in later Case.

Module 5: Performance Evaluation

This module will help to evaluate performance of proposed system including use of COATES Approach, Extension to Classification with Supervised K-Means Clustering Algorithm and with Improved K-Means Clustering Algorithm based on evaluation parameters like Cluster Purity, Effectiveness, Efficiency, etc.

C. Mathematical Modeling

When solving problems we have to decide the difficulty level of our problem. There are three types of classes provided for that. These are as follows:

1) P Class

2) NP-hard Class
3) NP-Complete Class

P Class Problems

Informally, the class P is the class of decision problems solvable by some algorithm within a number of steps bounded by some fixed polynomial in the length of the input. Turing was not concerned with the efficiency of his machines, but rather his concern was whether they can simulate arbitrary algorithms given sufficient time. However it turns out Turing machines can generally simulate more efficient computer models (for example machines equipped with many tapes or an unbounded random access memory) by at most squaring or cubing the computation time. Thus P is a robust class and has equivalent definitions over a large class of computer models. Here we follow standard practice and define the class P in terms of Turing machines.

NP-hard Problems

A problem is NP-hard if solving it in polynomial time would make it possible to solve all problems in class NP in polynomial time. Some NP-hard problems are also in NP (these are called "NP-complete"), some are not. If you could reduce an NP problem to an NP-hard problem and then solve it in polynomial time, you could solve all NP problems. Also, there are decision problems in NP-hard but are not NP-complete, such as the infamous halting problem

NP-complete Problems

A decision problem L is NP-complete if it is in the set of NP problems so that any given solution to the decision problem can be verified in polynomial time, and also in the set of NP-hard problems so that any NP problem can be converted into L by a transformation of the inputs in polynomial time.

The complexity class NP-complete is the set of problems that are the hardest problems in NP, in the sense that they are the ones most likely not to be in P. If you can find a way to solve an NP-complete problem quickly, then you can use that algorithm to solve all NP problems quickly.

Summary

After doing the study and analysis of various types of problems, It is found that topic entitled as "Implementation of Data Clustering with Meta Information based on COATES Approach" is of P-Class because:

1. Problem can be solved in polynomial time.
2. It takes fixed input and produces fixed output.

Let S be the system, then it can be presented as

S = {I, P, R, O} ---------- (1)

Where,

I represent the input dataset which gives the details about inputs to the proposed system,

P represents the set of functions in the form of processes that are applied on Input dataset,

R represent the constraints or rules applied during any process computation, and

O represents the set of final outputs of the proposed system.

Now, input for this system is considered as document collection input along with or without Meta Information. Therefore, the input set I represented as

I= {I1, I2} ----------(2)

Where, I1 is Document Collection with Meta Information; I2 is Document Collection without Meta Information

Processes

These are the functions of the system to represent the complete flow of execution from initial process to final step of execution. Each process does certain task as part of system computation. For this system, the main processes are described below:

P1 is Data Collection

P2 is Data Preprocessing

P3 is Application of COATES Approach

P4 is Extension to Classification

P5 is Performance Evaluation. Therefore, the set P is represented as:

P= {P1,P2,P3,P4,P5}----------(5)

Output

Outputs are the final outcomes of the system. Generally, this set represents only the final expected outcomes and does not provide the intermediate outcome details. The output set O can be written as

O= {O1, O2, O3} ---------- (4)

Where, O1 represents Clusters with COATES Approach; O2 represents Clusters from Supervised K-Means Clustering as Extension to Classification; O3 represents Clusters from Enhanced K-Means Clustering as Extension to Classification

The mapping of proposed system input, process and output set can be represented with the help of Venn diagram as shown in Fig.2.For the proposed system, to bring out completeness in the model presentation; the process state diagram is shown in Fig.3.

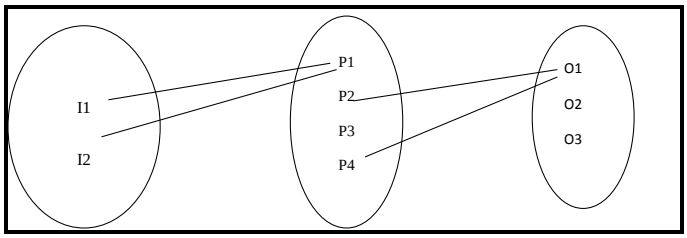

Fig.2.Input, Process and Output Mapping

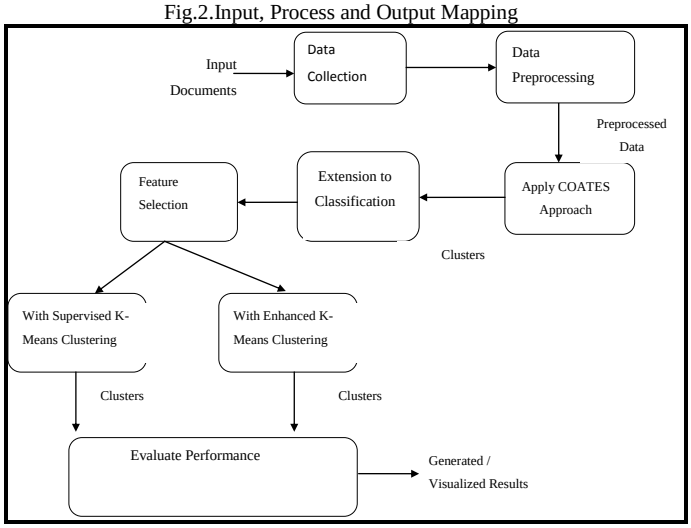

Fig.3. Process State Diagram

Time Complexity

The entire system time complexity includes the time computation required for each module of the system. Therefore, the time complexity of this proposed system is sum of each module time complexity.

V. RESULT ANALYSIS

This section highlights initial module development of the proposed system where developed GUI is explained in next section and further actual result analysis for this module are outlined in later part.

The development environment is selected for the proposed system development is Operating System: Windows XP and Above with Front-End: C#.NET and Back- End: MY SQL SERVER 2005 on single computer system with minimum 1GB RAM and enough storage space.

A. Provided User Inerface

Fig.4. shows the initial window developed to input the data contents for the proposed system. As stated earlier, with the help of provided provision as shown in Fig.4, user can upload standard dataset contents as well as synthetic data contents by browsing the desired storage location on the terminal. For this proposed system, the loaded data is considered as standard dataset and it is loaded into the system as shown in Fig.5.
Once, the original data contents are loaded, it cannot be used as it is in further computation and hence, required data preprocessing, so the system provides preprocessing facility as shown in Fig.6, where, by clicking on preprocess data tab, it is possible to get compatible form of data for next computation. The details of this step execution are given in Fig.7.

Fig.4. Window to Load Dataset

Fig.5. Selection of Dataset Contents

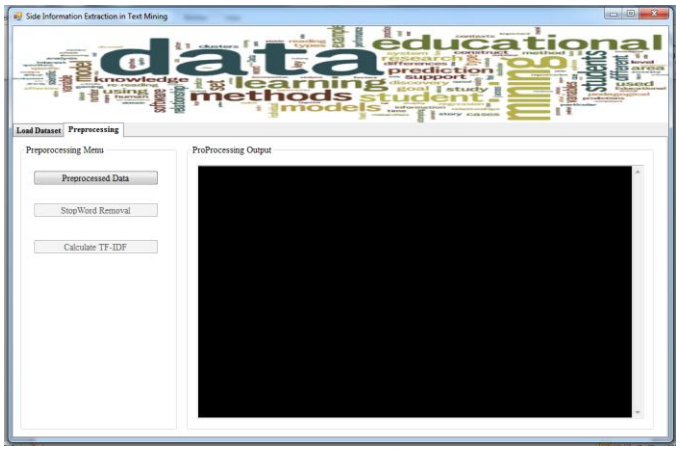

Fig. 6. Preprocessing Provision for Selected Dataset

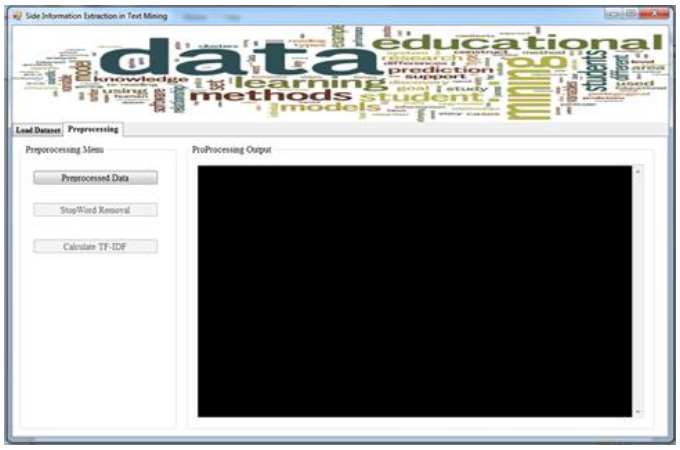

Fig.7. **Preprocessed Contents of Selected Dataset**

After getting compatible form of data, to form the clusters, it is necessary to calculate the TF-IDF values for each input data based on words in each input data. Fig.8, shows the provision provided in implemented module of the proposed system to do this operation

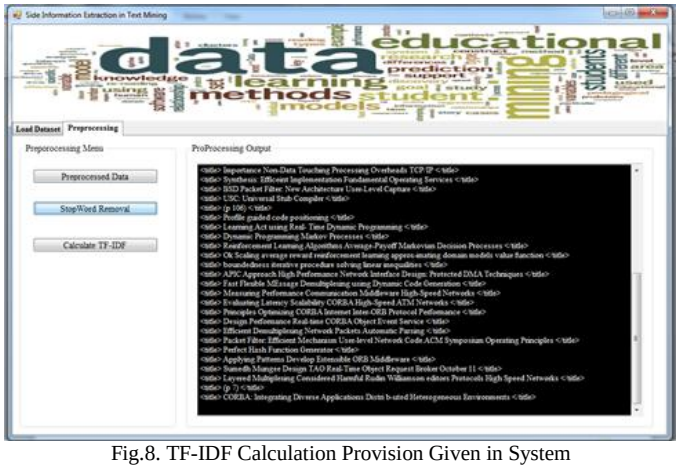

Fig.8. TF-IDF Calculation Provision Given in System

B. Experimental Result Analysis

Data Set Used

To test the effectiveness of developed module, Cora data set is used as standard dataset for system evaluation. The details of this dataset are given in following section.

Cora Data Set: The Cora data set1 contains 19,396 scientific publications in the computer science domain. Cora data set is classified into a topic hierarchy. On the leaf level, there are 73

classes in total. We used the second level labels in the topic hierarchy, and there are 10 class labels, which are Information Retrieval, Databases, Artificial Intelligence, Encryption and Compression, Operating Systems, Networking, Hardware and Architecture, Data Structures Algorithms and Theory, Programming and Human Computer Interaction. We further obtained two types of side information from the data set: citation and authorship. These were used as separate attributes in order to assist in the clustering process. There are 75,021 citation and 24,961 authors. One paper has 2.58 authors in average, and there are 50,080 paper-author pairs in total.

Results of Implemented Module

Following Fig.9 shows the original contents of Cora dataset before preprocessing module execution. It contains data contents with markup tags like </title>, </year>, etc. So, it is required to be cleaned and tag-free data contents to form hierarchy and clustering of original data contents. Therefore, to get compatible and cleaned data for processing, data preprocessing module is applied and result of implemented module execution is displayed in Fig.10 for Cora data set contents.


nd C++ Conference, </booktitle> <address> San Fran-cisco, California, </address> <month> April </month> <year> 1990, </year> <booktitle> USENIX, </booktitle> <pages> pp. 87102. 24 </pages>
150 http:##www.cs.wustl.edu#~cleeland#papers#ORB-patterns#ORB-patterns.ps.gz [SC99] <author> Douglas C. Schmidt and Chris Cleeland. </author> <title> Applying Patterns to Develop Extensible ORB Middleware. </title> <journal> IEEE Communications Magazine, </journal> <note> 1999. (to appear), available at http://www.cs.wustl.edu/~schmidt/ACE-papers.html. </note>
150 http:##www.cs.wustl.edu#~schmidt#ORB-patterns.ps.gz [SC99] <author> Douglas C. Schmidt and Chris Cleeland. </author> <title> Applying Patterns to Develop Extensible ORB Middleware. </title> <journal> IEEE Communications Magazine, </journal> <note> 1999. (to appear), available at http://www.cs.wustl.edu/~schmidt/ACE-papers.html. </note>
151 http:##www.cs.wustl.edu#~schmidt#RT-ORB.ps.gz [SLM97] <author> Douglas C. Schmidt, David L. Levine, </author> <title> and Sumedh Mungee . The Design of the TAO Real-Time Object Request Broker . October 11, </title> <booktitle> 1997. </booktitle> <pages> 81 </pages>
151 http:##www.cs.wustl.edu#~schmidt#TAO.ps.gz [SLM97] <author> Douglas C. Schmidt, David L. Levine, </author> <title> and Sumedh Mungee . The Design of the TAO Real-Time Object Request Broker . October 11, </title> <booktitle> 1997. </booktitle> <pages> 81 </pages>
152 http:##www.cs.comell.edu#skeshav#papers#transport.ton.ps [Tennenhouse89] <author> D. L. Tennenhouse. </author> <title> Layered Multiplexing Considered Harmful. In Rudin and Williamson, editors, Protocols for High Speed Networks. </title> <publisher> Elsevier, </publisher> <year> 1989. </year> <title> (p 7) </title>
153 http:##www-csag.cs.uiuc.edu#achien#cs491-f97#papers#vinoski.ps.gz [Vin97] <author> Steve Vinoski. </author> <title> CORBA: Integrating Diverse Applications Within Distri b-uted Heterogeneous Environments. </title> <journal> IEEE Communications Magazine. </journal> <month> February, </month> <year> 1997. </year>
153 http:##www.cs.wustl.edu#~schmidt#vinoski.ps.gz [Vin97] <author> Steve Vinoski. </author> <title> CORBA: Integrating Diverse Applications Within Distri b-uted Heterogeneous Environments. </title> <journal> IEEE Communications Magazine. </journal> <month> February, </month> <year> 1997. </year>

Fig.7. Cora Dataset Contents before Preprocessing

To get exact clusters of input side information data contents, at initial level, TF-IDF calculation is required. So, based on provided provision in implemented module of TF-IDF computation, results are obtained as shown in Fig.9.


<title> Importance Non-Data Touching Processing Overheads TCP/IP </title>
<title> Synthesis: Efficient Implementation Fundamental Operating Services </title>
<title> BSD Packet Filter: New Architecture User-Level Capture </title>
<title> USC: Universal Stub Compiler </title>
<title> (p 106) </title>
<title> Profile guided code positioning </title>
<title> Learning Act using Real- Time Dynamic Programming </title>
<title> Dynamic Programming Markov Processes </title>
<title> Reinforcement Learning Algorithms Average-Payoff Markovian Decision Processes </title>
<title> Ok Scaling average reward reinforcement learning approx-imating domain models value function </title>
<title> boundedness iterative procedure solving linear inequalities </title>
<title> APIC Approach High Performance Network Interface Design: Protected DMA Techniques </title>
<title> Fast Flexible MEssage Demultiplexing using Dynamic Code Generation </title>
<title> Measuring Performance Communication Middleware High-Speed Networks </title>
<title> Evaluating Latency Scalability CORBA High-Speed ATM Networks </title>
<title> Principles Optimizing CORBA Internet Inter-ORB Protocol Performance </title>
<title> Design Performance Real-time CORBA Object Event Service </title>
<title> Efficient Demultiplexing Network Packets Automatic Parsing </title>
<title> Packet Filter: Efficient Mechanism User-level Network Code ACM Symposium Operating Principles </title>
<title> Perfect Hash Function Generator </title>
<title> Applying Patterns Develop Extensible ORB Middleware </title>
<title> Sumedh Mungee Design TAO Real-Time Object Request Broker October 11 </title>
<title> Layered Multiplexing Considered Harmful Rudin Williamson editors Protocols High Speed Networks </title>
<title> (p 7) </title>
<title> CORBA: Integrating Diverse Applications Distri b-uted Heterogeneous Environments </title>

Fig.9. Cora Dataset contents After Preprocessing

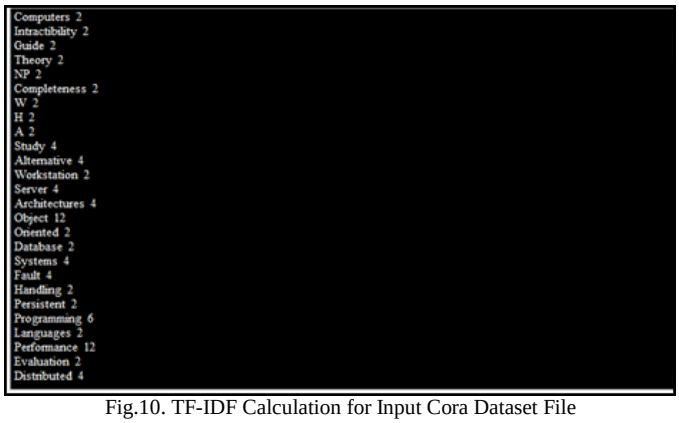
Computers 2
Intractibility 2
Guide 2
Theory 2
NP 2
Completeness 2
W 2
H 2
A 2
Study 4
Alternative 4
Workstation 2
Server 4
Architectures 4
Object 12
Oriented 2
Database 2
Systems 4
Fault 4
Handling 2
Persistent 2
Programming 6
Languages 2
Performance 12
Evaluation 2
Distributed 4

Fig.10. TF-IDF Calculation for Input Cora Dataset File

VI. CONCLUSION

The initial steps of the system have performed with successful results and with test cases. As with the standard dataset used to perform the data preprocessing steps have results and with removal of stops words and calculating the term frequency of the words. We proposed a systematic way to perform the mining process, so as to maximize the advantages from using this side information in text mining applications.

References

[1].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.

[2]. A. McCallum. (1996). Bow: *"A Toolkit for Statistical Language Modeling, Text Retrieval, Classification and Clustering"*, [Online] Available: http://www.cs.cmu.edu/ mccallum/bow

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

[4]. D. Cutting, D. Karger, J. Pedersen, and J. Tukey, *"Scatter/Gather: A cluster-based approach to browsing large document collections"*, in Proc. ACM SIGIR Conf., New York, NY, USA, 1992, pp. 318–329.

[5]. H. Schutze and C. Silverstein, *"Projections for efficient document clustering"*, in Proc. ACM SIGIR Conf., New York, NY, USA, 1997, pp. 74–81.

[6]. C. Silverstein and J. Pedersen, *"Almost-constant time clustering of arbitrary corpus set,"*, in Proc. ACM SIGIR Conf., New York, NY, USA, 1997, pp. 60–66.

[7] I. Dhillon, *"Co-clustering documents and words using bipartite spectral graph partitioning"*, in Proc. ACM KDD Conf., New York,NY, USA, 2001, pp. 269–274.

[8] I. Dhillon, S. Mallela, and D. Modha, *"Information-theoretic coclustering"*, in Proc. ACM KDD Conf., New York, NY, USA, 2003,pp. 89–98.

[9] T. Liu, S. Liu, Z. Chen, and W.-Y. Ma, *"An evaluation of feature selection for text clustering,"*, in Proc. ICML Conf., Washington, DC, USA, 2003, pp. 488–495.

[10] W. Xu, X. Liu, and Y. Gong, *"Document clustering based on nonnegative matrix factorization"*, in Proc. ACM SIGIR Conf., New York, NY, USA, 2003, pp. 267–273.