RESEARCH ARTICLE OPEN ACCESS

An Enhanced Novel Approach For Distributed Document Clustering

J.E. Johny Livinston[1], S. Siva Pradeepa[2]
1P.G. Scholar, Dept. of Software Engineering, Vins Christian College of Engineering, Nagercoil.
Email:- johnylivinston@gmail.com
2Asst. Professor, Dept. of Software Engineering, VCCE.

Abstract-
In an era of scientific progression, challenges in the field of Information Retrieval (IR) are wide spread and is so tedious to fathom, due to the increased usage of mass volumes of data. Hence, scalability and efficiency are the two main constraints when it pertains to IR from large complex distributed databases. Document clustering (text clustering) is one of the prime concerns for improvising the quality of IR, for both centralized and decentralized environments. So as to eradicate the aforementioned fallacies, this paper emphasizes on hybrid algorithms, one of which is the Particle Swarm Optimization (PSO) algorithm for the rational purpose of optimizing and subsequently followed with K-Means clustering algorithm. A framework is also being proposed for managing large volumes of data into chunks. This novel approach reaffirms the scalability and efficiency of document clustering on decentralized environments. Extensive evaluations based on simulation are carried out with the given datasets to demonstrate the effectiveness of the algorithm. Some of the applications include, the Sentimental Analysis on social networks, content search on Library of Congress, Boeing Aircraft, etc.

***Keywords** - Information Retrieval (IR), Decentralized,Document Clustering, Particle Swarm Optimization, K-Means Clustering*

I. INTRODUCTION

In a world where technology reigns, everything is networked, ranging from desktops to hand held Smartphone. One of the common purposes that put these devices to use is the social networking. Social networking is a massive trend where people communicate each other, either through voice or text or video, from one end of the globe to the other. Obviously one can visualize how massive and onerous this network can be. So the outcome of these is a huge enormous amount of data, which we call the Big Data. This is the realm where the choice of data mining is taken into account.

Data mining [1] is the concept of mining or extracting information from large datasets. In other words, it's the knowledge discovery or information retrieval. As aforementioned, the prime objective of data mining is to retrieve data from various sources. These sources can either be centralized or decentralized [1]. In centralized, all the data is bought to a unique site and then mining operation is performed on it. But in the case of decentralized, mining [3] is performed on distributed sites itself without bringing them to a central site. So in this case of decentralized mining, efficiency of algorithms takes its toll since scalability and performance of the algorithm plays a vital role in clustering and retrieval of data.

When it comes to mining from massive database [3], for instance let's take Twitter database from where we mine textual data; its number of users is more than billions and are scattered all over the world. So mining a desired textual data from this Big data is cumbersome. In the case of existing system, it emphasizes only on extracting data from small scale databases. Hence when it comes to Big data, scalability and performance of information retrieval takes its toll.

Hence the prime objective is to approach mining by the use of hybrid algorithms which significantly improvises the scalability and performance [7] which emphasizes on the clustering and retrieval of data, no matter what the size of the database and no matter the distribution of database.

Also, a framework is introduced unlike the existing system, for the sole intention of organizing the decentralized databases [1] which accelerates the mining process with efficient clustering through optimization. Hence, one of the other objectives is to introduce the aforementioned MapReduce framework for deliberately improvising the scalability, thus giving a hike to performance.

Information retrieval is an indispensable concept which is common in any domain [1]. Say for instance, Google, which is a search engine used for retrieving desired data from numerous distributed databases with an arbitrary query as an input.

These queries can be of textual form and is used to match the query with the N number of databases.

Sources holding document that holds the given text query is drawn out and is indexed on a table from where it is summarized and the desired data is produced to the end user.

Since users seek abrupt retrieval of data in this modern scientific era, existing system fails to satisfy the common valuable needs. Hence the problems to be eradicated are;

○ Scalability
constraints ○
Performance issues
○ Time complexity

One of the other problems that prevailed was that, the existing system didn't opt for the use of any frameworks. Basically, it performed operations on databases which are in the byte scales of Gigabytes and to an extent, Terabytes. Moreover, the number of nodes to be processed is also within the limits.

For the aforementioned process, lots of constraints are involved. Algorithms can't perform well on large scale databases [7], but even if it does perform, scalability issue is taken into account. More the number of databases, higher the time required to process them. Higher the processing time, lesser is the performance.

Hence such a scenario was acceptable to the traditional system, but not to the current system, as it engrosses the exploitation of internet by multiple nodes, which are emerging every now and then at a constant pace.

III. SYSTEM MODEL

Information retrieval is possible after performing chronological operations with the help of hybrid algorithms on three different modules, which of those are;

A. Preprocessing
B. Optimization
C. Clustering

Initially, the user gives the desired query to be processed. These queries are in the form of textual data. The subsequent modules elaborate the processes in detail.

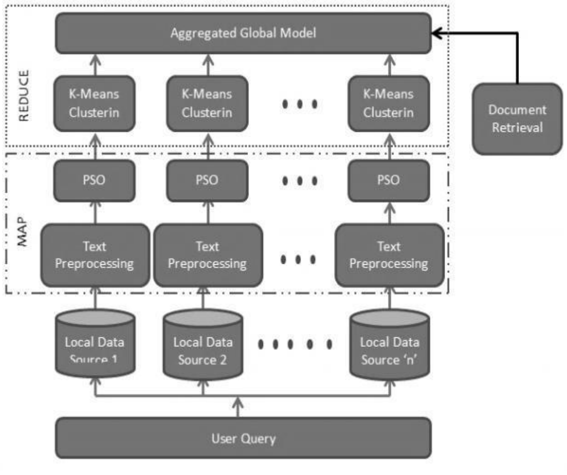

Fig. 1: System Model

A. PREPROCESSING

Pre-processing is the process of screening irrelevant, redundant, noisy and unreliable data for achieving better quality of data. Data pre-processing is an important step in the data mining process. The phrase "garbage in, garbage out" is particularly applicable to data mining and machine learning projects. Data-gathering methods are often loosely controlled, resulting in out-of-range values (e.g., Income: −100), impossible data combinations (e.g., Sex: Male, Pregnant: Yes), missing values, etc. Analysing data that has not been carefully screened for such problems can produce misleading results. Thus, the depiction and quality of data is first and foremost cleansed before running an analysis.

If there is much irrelevant and redundant information present or noisy and unreliable data, then knowledge discovery during the training phase is more difficult. Data preparation and filtering steps can take considerable amount of processing time.

Data preprocessing, includes cleaning, normalization, transformation, feature extraction and selection, etc. The outcome of preprocessing is the final training set. Prime preprocessing techniques which are applied are as follows.

i. Stop-word Removal

One of the simplest possible methods for feature selection in text clustering is that of the use of word frequency to filter out irrelevant features. While the use of inverse textual frequencies reduces the importance of such words, this may not alone be satisfactory to eradicate the snarky effects of very frequent words. In other words, words which are too frequent in the corpus can be detached because they are typically common words such as “a”, “an”, “the”, or “of” which are aloof from a clustering perspective. Such words are also referred to as stop words.

$$tf_{ij} = \frac{n_{i,j}}{\sum_k n_{k,j}} \qquad (1)$$

Where n_{ij} is the number of occurrences of the considered term in document d_j, and the denominator is the sum of number of occurrences of all terms in document d_j.

ii. Stemming

In the next step, the text documents have to be processed using the Porter stemmer. This concept involves the concept of removing suffixes which are particularly useful in the field of text mining and information retrieval. Instead of using the original terms in the documents, stemmed terms are used to construct a vector representation for each text document. The length of the consequential vectors after stemming is given by the number of different stemmed terms in the text corpus. Text is a collection of documents and each document is composed of a set of words or terms. In utmost scenarios, terms with a mutual stem would commonly have similar meaning. These words “test, testing, tester, tests” have same stem entitled as test.

Algorithm:-

1. Gets rid of plurals and -ed or -ing suffixes.
2. Turns terminal y to i when there is another vowel in the stem.
3. Maps double suffixes to single ones: -ization, -ational, etc.
4. Deals with suffixes, -full, -ness etc.
5. Takes off -ant, -ence, etc.
6. Removes a final -e.

iii. Pruning

For the purpose of experimental evaluations, all infrequent terms have to be left over. Let’s take a pre-defined threshold for instance, where a term is discarded from the representation (i.e., from the set), the basis behind pruning is with the aim of infrequent terms does nothing useful in identifying fitting clusters. While stemming, pruning and term weighting was performed, they have always performed them in the order in which it have been scheduled here. Some of the infrequent words are antidisestablishmentarianism, pneumonoultramicroscopicsilicovolcanokoniosis

$$if \sum_{d \in D} tf\,(d,t) \le \delta \qquad (2)$$

For a pre-defined threshold , a term *t* is discarded from the representation (i.e., from the set *T*). It's is illustrated on Fig 2. The basis behind pruning is with the aim of infrequent terms doing not be useful in identifying appropriate clusters.

B. OPTIMIZATION

Chosen a corpus of textual documents, apply Particle Swarm Optimization to gain most significant data. Particle swarm optimization (PSO) is a computational method that optimizes a delinquent dataset by iteratively trying to improve a candidate solution with regard to a given measure of quality. PSO optimizes a problem by having a population of N candidate solutions consisting of particles.

Each particle's movement is influenced by its local best known position and is inclined to one another, but, is also guided toward the best known positions in the search-space, which are rationalized as better positions are found by other particles. This is expected to move the swarm toward the best solutions.

Algorithm:-

1. Randomly choose k number of document vectors from the document collection as the initial cluster centroid vectors.
2. For each particle:
   - Assign each document vector in the document set to the closest centroid vector.
   - Calculate the fitness value based on the average distance between cluster centroid and a document.

$$f = \frac{\sum_{i=1}^{N_c} \{ \frac{\sum_{j=1}^{p_i} d(o_i, m_{ij})}{p_i} \}}{N_c} \qquad (3)$$

Where, d (o_i, m_{ij}) → distance between document $m_{ij}$and the cluster centroid o_i

P_i→ document number.
N_c→ cluster number.

- Determine the particle changes and its location.

3. Repeat step (2) until the stopping criterion is satisfied. i.e. No documents change clusters any more.

C. CLUSTERING

K-means [1] is one of the meekest unsupervised learning algorithms that help to eradicate clustering fallacies. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (take up k clusters) fixed apriori. The main notion is to outline k centres, one for each cluster. These centres should be sited in a shrewd way because of different location causes altered outcomes.

Hence, the better choice is to place them as much as conceivable far away from each other. The consecutive step is to take each point belonging to a given data set and subordinate it to the nearest centre [10]. Meanwhile, when no point is pending, the first step is completed and an early clustering is done.

Given a set of observations (x_1, x_2, …, x_n), where each observation is a d-dimensional real vector, k-means clustering targets to partition the n observations into k sets ($k \leq n$) S = {S_1, S_2, …, S_k} so as to minimize the within-cluster sum of squares:

$$arg_s min \sum_{i=1}^{k} \sum_{x_j \epsilon s_i} \| x_j - \mu_i \|^2 \qquad (4)$$

Where, μ_i is the mean of points in S_i.

Algorithm:-

1. Opt k random starting points as primary centroids for the k clusters.
2. Assign each document to the cluster with the nearest centroid.
3. Re-compute the centroid of each cluster as the mean of all cluster documents.

Repeat steps 2-3 until a stopping criterion is met. i.e., no documents change clusters anymore.

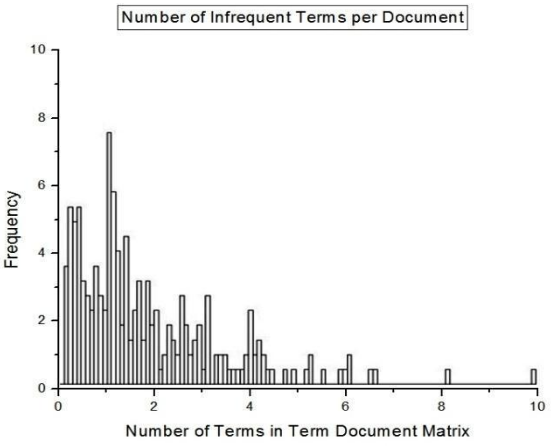

Fig. 2: Frequency of terms

IV. EXPERIMENTAL RESULTS

The above operations are implemented on pseudo distributed databases with an input query. Thousands of documents are loaded into the Revolution analytics toolset. For precise clustering to take place, as an initial leap, noisy data are removed. This redundancy removal step is the preprocessing phase of the system. We get perfect data that's of less ambiguity.

Added to it, a hybrid algorithm is implemented which will perform the optimization and clustering on the preprocessed data with more scalability. The extensive evaluation results are tabulated below.

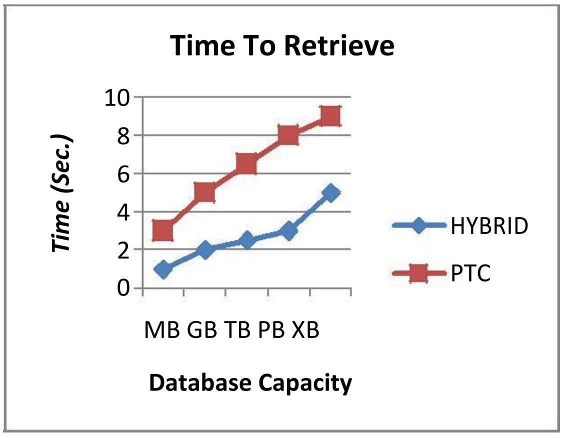

Fig. 3: Performance Comparison

The above illustrated figure 3, represents the performance measure of the existing and proposed algorithms in terms of database capacity and time in seconds. PTC fails in performance as the size of database increases. But, the proposed hybrid algorithm excels in Information Retrieval on huge databases.

NO	PARTICULARS	EXISTING SYSTEM	PROPOSED SYSTEM
1	TYPE	CENTRALIZED ALGORITHM	DISTRIBUTED ALGORITHM
2	ALGORITHM	PTC	PSO & K-MEANS
3	FRAMEWORK	-	MAPREDUCE
4	INFORMATION RETRIEVAL TIME	MORE	LESS
5	SCALABILITY	NO	YES
6	PERFORMANCE	LOW	EXCELLENT

Table 1: Evaluation

V. CONCLUSION AND FUTURE WORK

The enhanced novel distributed document clustering approach was developed using K-Means and Particle Swarm Optimization algorithms. The datasets are taken and processed on the basis of pseudo distributed environment. The entire data set is analyzed for input keyword given and every document which contains the keyword is retrieved back. A framework has been constructed for the process to take place, which is the MapReduce framework to improvise efficiency of IR.

The performance of the proposed algorithm can still be improved by blending it with any standard optimization technique and also different methods can be employed to improve the search quality and search performance as the size of data increases. The design can be more closely studied and implemented so that more comprehensive results can be obtained.

REFERENCES

[1] .Odysseas Papapetrou, Wolf Siberski, and Norbert Fuhr, "*Decentralized Probabilistic Text Clustering*", IEEE Transactions On Knowledge And Data Engineering, Vol. 24, No. 10, October 2012.

[2] Odysseas Papapetrou, Wolf Siberski, Wolfgang Nejdl, "*PCIR: Combining DHTs and Peer Clusters for Efficient Full-text P2P Indexing*", Computer Networks, vol. 54, no. 12, pp. 2019-2040, 2010.

[3] K.M. Hammouda and M.S. Kamel, "*Hierarchically Distributed Peer-to-Peer Document Clustering and Cluster Summarization*", IEEE Trans. Knowledge and Data Eng., vol. 21, no. 5, pp. 681-698, May 2009.

[4] Philippe Cudré-Mauroux, Suchit Agarwal, Karl Aberer, *GridVine: An Infrastructure for Peer Information Management*, Published by the IEEE Computer Society 1089-7801/07 © 2007.

[5] Jie Lu, Jamie Callan, *Content-Based Retrieval in Hybrid Peer-to-Peer Networks*, *ACM, CIKM'03*, November 3-8, 2003, New Orleans, Louisiana, USA. Copyright 2003.

[6] Martin Eisenhardt, Wolfgang Muller, Andreas Henrich, *Classifying Documents by Distributed P2P Clustering*, Informatik 2003 - Innovative Informatikanwendungen, October 2003.

[7] Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek, Hari Balakrishnan, *Chord: A Scalable Peer to peer Lookup Service for Internet Applications, SIGCOMM'01,* August 2731, 2001, San Diego, California, USA.

[8] Karl Aberer, Philippe Cudré-Mauroux, Anwitaman Datta, Zoran Despotovic, Manfred Hauswirth, Magdalena Punceva, Roman Schmidt, *P-Grid: A Self-organizing Structured P2P System,* (NCCR-MICS), a center supported by the Swiss National Science Foundation.

[9] Souptik datta, Chris Giannella, Hillol Kargupta, *K-Means Clustering Over a Large Dynamic Network*, Copyright by SIAM.

[10] Linh Thai Nguyen, Wai Gen Yee, Ophir Frieder, *Adaptive Distributed Indexing for Structured Peer-to-Peer Networks,ACM, CIKM'08*, October 26–30, 2008, Napa Valley California, USA. Copyright 2008.

[11] Ibrahim Aljarah and Simone A. Ludwig, *"Parallel Particle Swarm Optimization Clustering Algorithm based on MapReduce Methodology".*

[12] Yang Liu, Maozhen Li, SuhelHammoud, Nasullah Khalid Alham, Mahesh Ponraj, "*A MapReduce based Distributed LSI*".

[13] Y. Jahnavi, Y. Radhika , *"A Cogitate Study on Text Mining*", International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 – 8958, Volume-1, Issue-6, August 2012

Johny Livinston J.E.[1] received his Bachelor of Engineering (B.E.) degree in Computer Science and Engineering in 2012 from Anna University, Chennai. At extant he is pursuing Master of Engineering (M.E.) degree in Software Engineering under Anna University, Chennai. He

has presented more than 14 papers, both in Engineering and Business Management, on National and International Symposiums and Conferences. He has attended various Workshops and has Research interests on Data Mining, Big Data Analytics, Pervasive Computing, and Mobile Computing. Currently, he is carrying out a project on Data Mining and Big Data Analytics.