ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 11, November 2016

Characterization of User Inclinations for Service Recommender System in Big Data Applications

S. Kalaimani[1], Dr. R. Mala[2]

Research Scholar, Department of Computer Science, Marudupandiyar College, Thanjavur, Tamil Nadu, India[1]

Assistant Professor, Department of Computer Science, Marudupandiyar College, Thanjavur, Tamil Nadu, India[2]

ABSTRACT:Recommender system give proper suggestion to clients when client utilize online service on the web. Conventional strategy is inefficient when handling or examining huge volume of information. Existing recommender system consider client inclination and it is prescribed to the client utilizing relational database and linear processing. In this paper we classify the client inclination as positive and negative for proposals to client uses NoSQL(Not Only SQL) database and parallel handling for recommender system. Machine learning technologies are applied to the group the client inclination as a result of their capacity to gain from the training dataset to foresee or bolster basic leadership with generally high exactness. The client inclination are different sorts, for example, unstructured, semi-organized and it is mind boggling to deal with the relational databases so we go for NoSQL database to process this kind of dataset. NoSQL database give Non-relational and pattern less information model, Low latency and elite, highly scalable used to deal with the client inclination for grouping reason and proposal to client. As the inclination are sorted it enhance the precision of administration recommender systems utilizing NoSQL databases and Parallel Processing.

KEYWORDS:Big data, NoSQL, Polarity Classification, Parallel Processing, Recommender system.

I. INTRODUCTION

In recent years Products recommendations have become increasingly popular in e-commerce services such as in Amazon Instant Video service and Netflix service. The main goal of a recommender system is to suggest unknown items (movies or other entertainment media) by considering the information exchanged by users when interacting with the system [1]. Until recently, users commonly asked for a recommendation from their own circle of known friends or family. However, recommendations demand a certain level of trustworthy knowledge and not everyone is eligible to provide a skilled recommendation. Thus, a recommender system that observes the viewpoints of diverse users and movies may offer a more reliable and insightful recommendation than average user which is limited to the movies own awareness in a vast number of different movies.

Recommender Systems apply machine learning and data mining techniques for filtering unseen information and can predict whether a user would like a given resource [2]. Data on the web has been explosively increasing in the past few decades. The ability to automatically mine useful information from massive data has been a common concern for organizations who own large datasets. The Map Reduce framework [5] is commonly used to analyze extremely large datasets such as tweets collections, online documents or large-scale graphs [6] [7] [8]. The framework provides a simple and powerful interface for programmers to solve large-scale problems using a cluster of commodity computers.

A typical method to obtain valuable information is to extract the Polarity from a message. Polarity classification is useful for the business consumer industry or online recommendation systems. For example, online product reviews are usually analyzed by manufacturers to decide what products they will produce in the future to reduce risk. Machine learning technologies, such as Naive Bayes and support vector machine, are widely used in sentiment classification [9] [10] [11] [12] [13] [14] because of their ability to learn from the training dataset to make decisions with on-line data, to predict salient features, and to provide real-time analysis with relatively high accuracy.

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 11, November 2016

When the datasets are large, some information fusion algorithms might not scale up well. For example, if an algorithm needs to load data into memory constantly, the program may run out of memory for large datasets. One promising approach is to utilize and adapt Map Reduce for some machine learning technologies to resolve these large-scale problems. Apache Mahout is a machine learning library for clustering, classification and filtering, implemented on top of Hadoop, the open source version of Map Reduce. Although there are some machine learning algorithms implemented in Mahout, it is still helpful to study how to convert a machine learning algorithm to a Hadoop program and to optimize the algorithm scalability in large datasets [2].
In this paper, we aim to categorize the user preference as positive and negative using Naive Bayes classifier (NBC) in large-scale datasets. We can use suitable Recommender System to improve the results when the dataset size increases.
The rest of this paper is organized as follows. Section II introduces the background of this study. Section III illustrates the system description. The system is built on top of Hadoop and MongoDB basic components. The details of implementing Naive Bayes classifier addressed in section IV. Section V. Finally, we conclude.

II. BACKGROUND

A. NoSQL

The Map Reduce [3] model and NoSQL data stores have evolved due to large amounts of Internet and log data. The Map Reduce programming model emerged for processing large data sets on large commodity clusters. The model allows users to write map and reduce functions to operate on the data and the system provides scalability and fault-tolerance. Traditionally, Map Reduce jobs have relied on specialized file systems such as Google File System [5]. Apache Hadoop [15] is an open-source Map Reduce implementation and in the last few years has gained significant traction. NoSQL [3] databases, does not use SQL or the relational model. NoSQL data stores achieve scalability through a distributed architecture and by trading consistency for availability.
Recently, a number of NoSQL stores have provided extensions that allow users to use Hadoop/Map Reduce to operate on the data stored in these NoSQL data stores. An integrated environment that allowed one to capture semi structured data using a NoSQL store such as MongoDB, yet use the Map Reduce model for analysis, would address the data requirements from these scientific communities. However, there is a limited understanding of the performance, scalability and fault-tolerance trade-offs in this integrated environment.

B. MAP REDUCE

Hadoop is an open-source platform that allows for the distributed processing of large datasets across clusters of computers using simple programming model. It is designed to scale up from single servers to thousands of machines, each offering local computation and storage. Hadoop comprises two important modules: HDFS and Map Reduce. HDFS is a distributed file system that provides high-throughput access to application data and Map Reduce is a programming model for parallel processing of large datasets [16]. Therefore, we choose Hadoop as the develop platform to study the scalability of Naive Bayes classifier.
The Map Reduce framework has been used to process large datasets since the original paper [5] was published. Googles clusters process more than 20 Petabytes of data every day by running one hundred thousand Map Reduce jobs on average [15]. Using this framework, programmers only need to focus on problem solving versus implementation. The Map Reduce runtime system will take care of the underlining parallelization, fault tolerance, data distribution and load balance. Google file system (GFS) is a distributed file system that Map Reduce uses for the storage of large amount of data across inexpensive hard drives. The availability and reliability of underlining unreliable hardware are provided by replicating file blocks and distributing them across different nodes.

C. NAIVE BAYES CLASSIFICATION

The Naive Bayes classifier is based on the Bayes theorem with strong (Naive) independence assumption, and is suitable for the cases having high input dimensions. The Naive assumption is that features are conditionality independent, for instance in a document the occurrence of words (features) do not depend upon each other [4].
Naive Bayes has proven to be a simple and effective machine learning method in previous text classification studies. It is even optimal in some cases [16] [17].

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 11, November 2016

Suppose there are m possible classes $C = \{c_1,c_2,...,c_m\}$ for a domain of documents $D = \{d_1,d_2,...,d_n\}$. Let $W = \{w_1,w_2,...,w_s\}$ be the set of unique words, each of which appears at least once in one of the documents in D. The probability of a document d being in class c can be computed using Bayes rule [2]:

(1)

$$P(c|d) = \frac{P(c)P(d|c)}{P(d)}$$

Since $P(d)$ is a cor... data set size, the denominator of (1) is typically not calculated for maximum a posteriori (MAP), ...ric statistical problems. A Naive Bayes model assumes that each term or word, w_i, in a document occurs independently in the document given the class c. Therefore, equation (1) becomes:

$$P(c|d) \propto P(c)\prod_{k=1}^{n_d}[P(w_k|c)]^{t_k}$$

Where n_d is the number of unique words in document d and t_k is the frequency of each word w_k. To avoid floating point underflow, we use the equivalent equation:

(2)

$$\log P(c|d) \propto \log P(c) + \sum_{k=1}^{n_d}[t_k \log P(w_k|c)]$$

The class of d is decided as the class, c^*, which maximizes log $P(c|d)$ in equation (3).

(3) $$c^* = argmax_{c\in C}\{\log P(c) + \sum_{k=1}^{n_d}[t_k \log P(w_k|c)]\}$$

When applying Naive Bayes classifier (NBC), we can estimate $P(c)$ and $P(w_k|c)$ as:

$$P(c) = \frac{N_c}{N} \quad and \quad P(w_k|c) = \frac{N_{w_k}}{\sum_{w_i \in W} N_{w_i}}$$

Where N is the total number of documents, N_c is the number of documents in class c and N_{wi} is the frequency of a word w_i in class c. With these estimations, the calculation of the right hand side of equation (3) is essentially a counting problem. This makes Map Reduce a suitable framework for the implementation of NBC in large-scale datasets.

III. SYSTEM DESCRIPTION

On top of Hadoop Map Reduce and HDFS, NoSQL Database MongoDB we designed a Big Data analyzing system to evaluate whether Service Recommender System can scale up to Recommendation millions of movie reviews. This section explains the three modules of the system and important steps of the work flow.

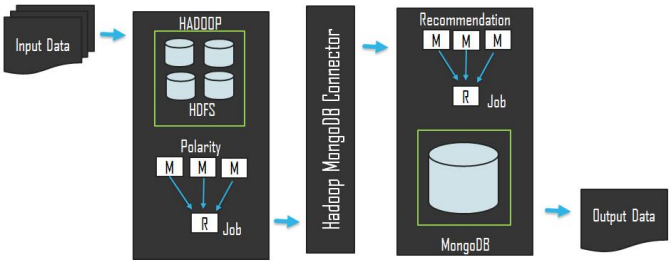

Fig. 1: A Simple System Architecture

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 11, November 2016

A. SYSTEM ARCHITECTURE

As shown in Fig. 1, the system describe the process of polarity classification and Recommendation using software components such as Hadoop and MongoDB. We design this system based on our need to generate different sizes of datasets and test the Hadoop program on them respectively.
Our raw data comes from large sets of movie reviews collected by research communities. The data preprocessing is responsible to produce the desired data format to assist the program to efficiently process each review. The user submits jobs through the user terminal. Experiment results are also accessible through the user terminal after the job completion of MongoDB Map Reduce job in Database.

B. DATASET

In our experiments, we use three datasets: the Cornell University movie review dataset [19] and Stanford SNAP Amazon movie review dataset [20].The Movie Lens data set contains 10000054 ratings and 95580 tags applied to 10681 movies by 71567 users of the online movie recommender service Movie Lens [21]. The Cornell dataset has 1000 positive and 1000 negative reviews, which makes P (c) in equation (3) 0.5 for both class. A reviews class is then decided by the frequency of each word that appears in the model obtained from training dataset. Table I lists the five Dataset and its details.

TABLE I: Datasets Description

Datasets	User Count	Movie Count	Review Count
Polarity	—	—	2000
Reviews	889,176	253,059	7,911,684
Movie	71567	10681	—
Rating	71567	10681	10000054
Tags	71567	10681	95580

The Amazon movie review dataset is organized into eight lines for each review, with additional information such product identification (ID), user ID, profile Name, score, summary etc. Since it is a 5 points rating system, we simply divide all reviews into positive and negative subset using 3.0 as a threshold. We consider unigrams only for the Naive Bayes model.
Users were selected at random for inclusion. All users selected had rated at least 20 movies. Unlike previous MovieLens data sets, no demographic information is included. Each user is represented by an id, and no other information isProvided. The data are contained in three files, movies.dat, ratings.dat and tags.dat. Also included are scripts for generating subsets of the data to support five-fold cross-validation of rating predictions. More details about the contents and use of all these files follows. This and other GroupLens data sets are publicly available for download at GroupLens Data Sets.

C. OVERALL WORK FLOW

Pre-Processing Raw Dataset:The data parser first preprocesses all reviews into a common format. After the processing, each review is one line in the dataset, with document ID and sentiment (positive or negative) prefixed. This is useful because by default Map Reduce splits the input files by line and passes each line to a mapper. To pre-process a raw review from the dataset, we simply delete unwanted context such as punctuation, special symbols and numbers. We did not introduce a lexicon or vocabulary to filter out meaningless words.

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 11, November 2016

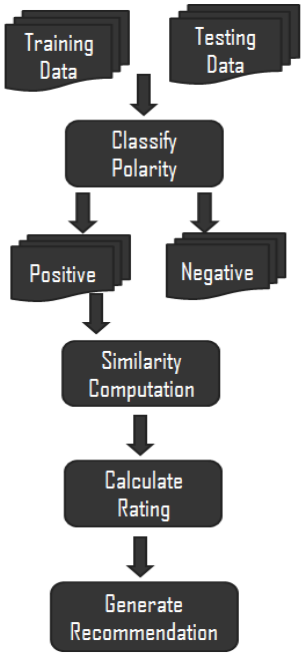

Fig. 2: Service Recommender System Work Flow

The reviews after pre-processing, each review is one line in the dataset file. We use two prefixed tags: sentiment of either positive or negative and document ID, which is a number. The two tags are both surrounded by colons to help the program differentiate them from review text. Note that there are some isolated single letters resulting from the remove of punctuation.

Preparing Input Datasets:The Data Preprocessing to prepare input data sets for all test trials. The data Preprocessing thread extracts from the repository the desired number of positive reviews, as well as the same amount of negative reviews. The result is an input dataset of two equal-size classes of movie reviews. After a dataset is generated, the desired input is moved to HDFS. After the job completion of the polarity classification the data is written in MongoDB for Recommendation Process this time the data in the form of JSON format.

Polarity Classification Using Hadoop:The Polarity classification is the key step in the work flow. Fig. 2 shows the job sequence of this step. Once the training data and test data are ready in HDFS, the WFC starts the training job to build a model. The combining job then combines test data with the model, resulting an intermediate table. Finally, the classify job simultaneously computes the probabilities of each review in the two classes respectively and makes a decision about the Polarity of this review.

Reading Data from HDFS and Store in MongoDB:In this section reading the data from the HDFS and store in mongoDB, During This process for each file parse each record and produce a function that performs tokenization and parsing. The data set is a comma-separated values (CSV) and this is converted in to JSON form this is native data structure form of MongoDB. After this process three tables created name as following Movies, Tags, and Ratings. After this preprocessing the dataset is ready for following processes.

Recommendation using MongoDB:After the Preprocessing the dataset are applied for following sequence for parallel processing using MongoDB Map Reduce framework shown in Fig. 2. Similarity Computation this is perform two process Finding Users Similar to a Particular User and Finding Number of Movies that Belong to each Genre. Calculate Rating such as Finding Average Rating of Movies. Generate Recommendation.

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 11, November 2016

IV. IMPLENMENTAION ON MAP REDUCE

NBC on Map Reduce

Defining the Problem:We first define our task as to classify the Polarity of a movie review, positive or negative. Hence, we only have two classes of documents. To simplify the problem, we choose the same number of positive and negative reviews, which makes the $P(c)$ in equation (3) a constant. The estimation of $P(w_k|c)$ is computed by the relative frequency of wk in all documents in c. The classification problem is then converted to a counting problem on the training and test datasets.

Algorithms:After the simplification of the problem, the task can be divided into three sequential jobs as follows.

i) Training job all training reviews are fed into this job to produce a model for all unique words with their frequency in positive and negative review documents respectively.

ii) Combining job in this job, the model and the test reviews are combined to an intermediate table with all necessary information for the final classification.

iii) Classify job this job classifies all reviews simultaneously and writes the classification results to HDFS.

These jobs are executed in sequence because the dependencies between them, as shown in Fig.2. The training job produces a model that is used to compute the probability of each word in the two classes. The combining job then associates the test data with the model, excluding the words that appear in test data but not in the model. The intermediate table produced by the combining job is then fed to the classify job. By the end of classify job, all reviews are classified into positive or negative classes.

V. EXPERIMENTAL STUDY

A. COMMODITY CLUSTER INFRASTRUCTURE

Physical Hadoop cluster created using four computers and its configurations are Intel 2.3GHz cores, 2G RAM Primary, 320G DDR3 secondary memory.

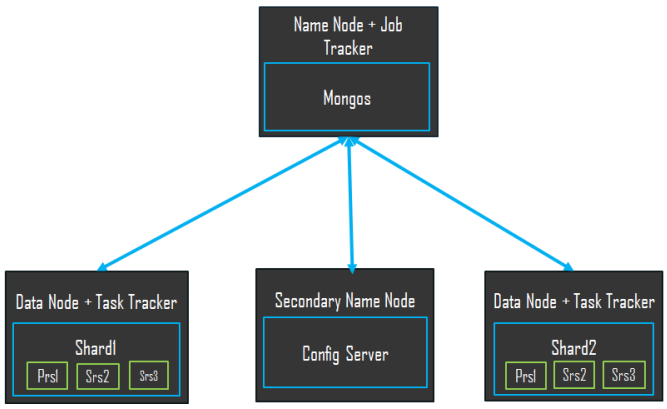

Fig. 3: Commodity Hadoop and MongoDB Cluster Infrastructure Setup Using Four Computers

Fig 3 shows there is one Name node, and then one Secondary Name Node, The rest of two Data nodes that we use to manage the Hadoop Map Reduce jobs in the cluster. The Backend NoSQL Database MongoDB cluster is created all four computer nodes Fig 4 shows there is one or three mongos, config server, sharding (shard1, shard2…ShardN) and replicasets such as Primary replica set (Prs1), Secondary replicaset (Srs2), Secondary replicaset (Srs3) in this clusters.

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 11, November 2016

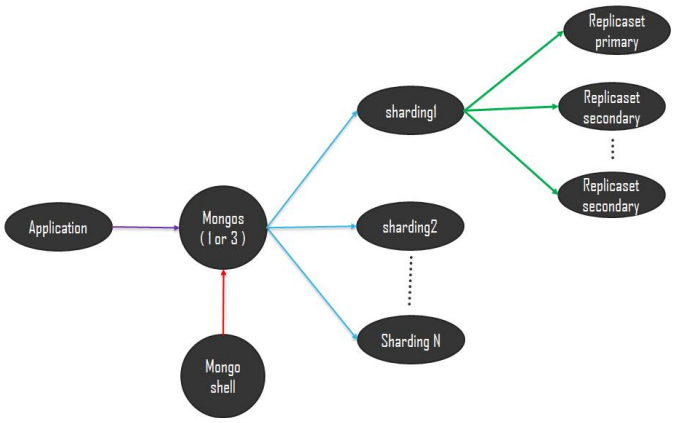

Fig. 4: Replicaset and sharding setup

B. EXPERIMENTAL SETUP

First we tested our code on Cornell dataset without changing the Hadoop code, our program could classify different Subsets of Amazon movie review dataset with comparable accuracy. To test it improve the throughput of service Recommender system of the size of dataset in our experiment varies from one thousand to one million reviews in each class.

C. AN IMPROVED SWITCHING HYBRID RECOMMENDER SYSTEM USING NAIVE BAYES CLASSIFIER AND COLLABORATIVE FILTERING

They consider the recommender systems apply machine learning and data mining techniques for filtering unseen information and can predict whether a user would like a given resource. To date several recommendation algorithms have been proposed, where collaborative filtering and content-based filtering are the two most famous and adopted recommendation techniques. Collaborative filtering recommender systems recommend items by identifying other users with similar taste and use their opinions for recommendation; whereas content-based recommender systems recommend items based on the content information of the items. These systems suffer from scalability, data sparsity, over specialization, and cold-start problems resulting in poor quality recommendations and reduced coverage. Hybrid recommender systems combine individual systems to avoid certain mentioned limitations of these systems. In this paper, we proposed a unique switching hybrid recommendation approach by combining a Naive Bayes classification approach with the collaborative filtering. Experimental results on two different data sets, show that the proposed algorithm is scalable and provide better performance–in terms of accuracy and coverage–than other algorithms while at the same time eliminates some recorded problems with the recommender systems.

VI. CONCLUSION AND FUTURE WORK

In this paper, we displayed a straightforward system for Service Recommender System on large datasets utilizing a Naive Bayes classifier with the Hadoop framework, MongoDB NoSQL Database. We executed the NBC on top of Hadoop framework and store in MongoDB NoSQL Database. Our outcomes demonstrate that Polarity characterization and in addition Recommendation this can scale up effectively, utilizing NoSQL database. A keen channel may be useful to build the throughput. Future work will be identifying suitable recommendation technique to improve the efficiency of the results.

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 11, November 2016

REFERENCES

[1] Mustansar Ali Ghazanfar and Adam Pr gel-Bennett, "An Improved Switching Hybrid Recommender System Using Naive Bayes Classifier and Collaborative Filtering", in Proceedings of the International MultiConference of Engineers and Computer Scientists Vol-I, Hong Kong, IMECS March 17–19, 2010.

[2] Bingwei Liu, Erik Blasch, Yu Chen, Dan Shen and Genshe Chen, "Scalable Sentiment Classification for Big Data Analysis Using Naive Bayes Classifier", Big Data, 2013 IEEE International Conference on IEEE, Oct, 2013, pp. 99–104.

[3] Vijaykumar S, and Saravanakumar S.G, Future Robotics Database Management System Along With Cloud TPS, International Journal on Cloud Computing: Services and Architecture, 1(3), 103-114, 2011.

[4] Chaobo He, Yong Tang, Zhenxiong Yang, Kai Zheng and Guohua Chen, "SRSH: A Social Recommender System based on Hadoop", International Journal of Multimedia and Ubiquitous Engineering Vol.9, No.6 (2014), CA, pp. 141–152.

[5] J. Dean and S. Ghemawat, "Map Reduce: simplified data processing on large clusters", in Proceedings of the 6th conference on Symposium on Operating Systems Design & Implementation - Volume 6, ser. OSDI04. Berkeley, CA, USA: USENIX Association, 2004.

[6] U. Kang, D. H. Chau, and C. Faloutsos, "Pegasus: Mining billion-scale graphs in the cloud", in Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on pp. 5341–5344.

[7] S. Suri and S. Vassilvitskii, "Counting triangles and the curse of the last reducer", in Proceedings of the 20th international conference on World Wide Web. ACM, 2011, pp. 607–614.

[8] U. Kang, D. H. Chau, and C. Faloutsos, "Counting triangles and the curse of Mining large graphs: Algorithms, inference, and discoveries", in Data Engineering (ICDE), 2011 IEEE 27th International Conference on, pp. 243–254.

[9] B. Pang, L.Lee, and S. Vaithyanathan, "Thumbs up?: sentiment classification using machine learning techniques", in Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10, pp.79–86, 2010.

[10] B. Pang and L. Lee, "A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts ", in Proceedings of the 42nd annual meeting on Association for Computational Linguistics, pp. 271.

[11] P. Chesley, B. Vincent, L. Xu, and R. K. Srihari, "Using verbs and adjectives to automatically classify blog sentiment ", Training, vol. 580, no. 263, pp. 233

[12] M. Gamon, A. Aue, S. Corston-Oliver, and E. Ringger, "Pulse: Mining customer opinions from free text", in Advances in Intelligent Data Analysis VI. ringer, 2005 pp. 221–132.

[13] A. Kennedy and D. Inkpen, "Sentiment classification of movie reviews using contextual valence shifters", Computational Intelligence, vol. 22, no. 2, 2006 pp. 110–125.

[14] J. Dean and S. Ghemawat, "Map Reduce: simplified data processing on large clusters ", Communications of the ACM vol. 51, no. 1 pp. 107–113, 2008

[15] D. Lewis, "Naive (bayes) at forty: The independence assumption in information retrieval", in Proceedings of the 20th international conference on World Wide Web. Machine Learning: ECML-98 pp. 4–15, 1998.

[16] P. Domingos and M. Pazzani, "On the optimality of the simple bayesian classifier under zero-one loss", Machine learning, vol. 29 no. 2-3, pp. 103–130, 1997.

[17] H. Karloff, S. Suri, and S. Vassilvitskii, "A model of computation for Map Reduce", in Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms. Society for Industrial and Applied Mathematics, 2010, pp. 938–948.

[18] http://www.cs.cornell.edu/people/pabo/movie-review-data/

[19] http://snap.stanford.edu/data/web-Movies.html

[20] http://grouplens.org/datasets/movielens/

[21] http://en.wikipedia.org/wiki/MinHash/

[22] A. Nancy, Dr.M. Balamurugan, S. Vijaykumar, "A Comparative Analysis of Cognitive Architecture", International Journal of Advanced Research Trends in Engineering and Technology (IJARTET), Vol.3, Special Issue 20, April 2016, Pg. 152-155, ISSN (P): 2394-3777, ISSN (O): 2394-3785.

[24] M. Mayilvaganan, D. Kalpanadevi. Comparison of Classification Techniques for predicting the Cognitive Skill of Students in Education Environment. 2014 IEEE International Conference on Computational Intelligence and Computing ResearchIEEE Xplore. Advancing Technology for Humanity. IEEE Catalog Number: CFP1420J-PRT. ISBN: 978-1- 4799-3974-9. Page: 279-282. (Thomson Reuters and Scopus Indexed)

[25] Ghosh K, Indra N. Efficacy of Centella asiatica ethanolic extract on cadmium induced changes in blood haematology, hepatic and nephritic function markers in albino rats. Int J Curr Res 2015 07(07): 18283-18288. [ISSN 0975-833X]