International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)
Web Site: www.ijettcs.org Email: editor@ijettcs.org
Volume 5, Issue 4, July - August 2016 **ISSN 2278-6856**

HIGH PERFORMANCE SEQUENCE MINING OF BIG DATA USING HADOOP MAPREDUCE IN CLOUD

Dr. B.LAVANYA[1], Ms. G.LALITHA[2]

[1]Assistant Professor, Dept. of Computer Science, University of Madras, Chennai – 600 005.

[2]Research Scholar, Dept. of Computer Science, University of Madras,

Abstract
Text mining can handle unstructured data. The proposed work extracts text from a PDF document is converted to plain text format, then document is tokenized and serialized. Document clustering and categorization is done by finding similarities between documents stored in cloud. Similar documents are identified using Singular Value Decomposition (SVD) method in Latent Semantic Indexing (LSI). Then similar documents are grouped together as a cluster. A comparative study is done between LFS (Local File System) and HDFS (HADOOP DISTRIBUTED FILE SYSTEM) with respect to rapidity and dimensionality. The System has been evaluated on real-world documents and the results are tabulated.

Keywords: Big data, MAPREDUCE, SVD, LSI.

1. INTRODUCTION
Text's data are of two form, structured data and unstructured data. Structured data is in the first normal form stored in the relational databases, whereas the unstructured or semi-structured data are stored in the form of articles or documents.Text mining handles unstructured or semi-structured documents. Mining text data involve certain set of preprocessing steps. In documents two types of words are present, synonyms and homonyms. Different kinds of words that share the same meaning are called synonyms and words having same spelling with different meaning are called homonyms. In this paper we propose a system which does text compression, text categorization finally text clustering. The textual, unstructured document makes the above mentioned tasks complicated.

2. LITERATURE REVIEW
Ronen Feldman et al., [1] proposed a Knowledge Discovery in Databases called as Data mining. They proposed a tool for effectively mining interesting patterns from the large amount of data, which is available in unstructured format and proposed a taxonomy filtering approach using taxonomy creation tool and stated that text mining serves as a powerful technique to manage knowledge encapsulated in large document collections.

ShivakumarVaithiyanathan et al., [2] proposed a method to describe keywords of documents. A document is represented in a matrix form by applying dimensionality reduction; initial matrix is reduced to resultant matrix. The related resultant vectors are then clustered. For each cluster, the term having greatest impact in the document, is identified. Those terms form a cluster summary indicative, for the documents in the cluster.
Joel LaroccaNeto et al., [3] proposed a text mining tool performing two tasks, namely document clustering and text summarization. In this document clustering is performed by using the AUTOCLASS data mining algorithm; and Text summarization algorithm is based on computing the value of a TF-ISF (term frequency – inverse *sentence* frequency) measure for each word, which is an adaptation of the conventional TF-IDF (term frequency – inverse *document* frequency). Sentences with high values of TF-ISF are selected to produce a summary of the source text.

ManishaSahane et al., [4] the research objective is to study the HADOOP and its associated technologies with glance focus on MAPREDUCE and analysis of university research data set to know the focused area of research in Zoology and Botany department. Yen-hui Liang et al., [5] proposed frequent item set mining (FIM) to mine human behavior. Proposed a new distributed FIM algorithm called Sequence-Growth, and implemented on MAPREDUCE Framework, applied in an algorithm called lexicographical order to construct a tree called "lexicographical sequence tree" which allows finding all frequent item sets without exhaustive search over the transaction databases. They concluded that Sequence-Growth produced good efficiency and scalability with big data and long item sets.

Jingjing Wang et al., [6] used Locality Sensitive Hashing (LSH) technique for similarity joins for high dimensional data. The efficiency and approximation rate of LSH depend on number of false positive instances and false negative instances. So they proposed a technique called

Personalized Locality Sensitive Hashing (PLSH), where a new banding scheme is embedded to tailor the number of false positives, false negatives, and the sum of both. PLSH is implemented in parallel using MAPREDUCE framework to deal with similarity joins on large scale data.

Nagwani et al., [7] proposed a technique for faster understanding of text documents. In this a novel framework called MAPREDUCE is used for summarizing large text collection. Proposed a method called Latent Dirichlet Allocation (LDA) for summarizing the large text collection over MAPREDUCE framework. The summarization task is performed in four stages. The presented technique is evaluated in terms of compression ratio, retention ratio, ROUGE and pyramid score. MAPREDUCE is used for faster implementation of summarizing large text collections and is a powerful tool in Big Text Data analysis.

Negrevergne et al., [8] proposed a technique to find sequence of symbols that are included in a large number of input sequences that satisfy some user specified conditions. They proposed a constraint based framework for finding sequence of symbols. Feinerer et al., [9] proposed a method to import data, corpus handling, preprocessing, Meta data management and a creation of term-document matrices.

El-Halees et al., [10] Large amount of information available in text document only, it is important to use text mining to discover knowledge from unstructured data, this paper deal about Arabic text documents. Arabic language is inflectional and derivational language which makes text mining complicated. They used technique such as preprocessing, tokenizing, steaming and part-of-speech, and then they used maximum entropy method to classify Arabic documents.

3. PROBLEM DEFINITION

E-book lodge massive storage in cloud, our objective is to save storage space, by compressing actual content before stowing it on cloud. Text documents were scattered in cloud by categorizing and clustering related documents together that aids e-documents to be accessed in efficient way.

3.1 Phases of the Proposed Work

In Step1 Text Compression is done. Text extracted from PDF and saved in plain text format. Then input document is tokenized and serialized. Next, the document is compressed, decompressed and formalized using effective methodology.
In Step 2 Text Categorization is done. Unique words from a document is tokenized and serialized, and then stop words are pruned from the document, top ten highest ranked terms are selected as keywords, using APRIORI algorithm.
In Step 3 Document Ranking is done using SVD, which ranks the documents based on user queries.
In Step 4 Document is executed in HDFS. Document is tokenized and serialized in HDFS for effectively handling Big Data in cloud at minimal duration.

4. C2RH METHODOLOGY

The Proposed system comprises of the following preprocessing steps

4.1 Text Compression

Step 4.1.1 Data Collection

Input documents are chosen from Google books. (e. g. Data Mining and its applications, Data Preprocessing in Data Mining, Data Mining and Machine Learning in Cyber Security and so on) .

Step 4.1.2 Data Extraction

The Input document in PDF format is converted to plain text and the text extracted is used for further processing.

Step 4.1.3 Document Tokenization and Serialization Unique words from the document are tokenized and serialized along with their frequencies, as shown in Fig 1.

E.g. Serial Number: Term: Frequency

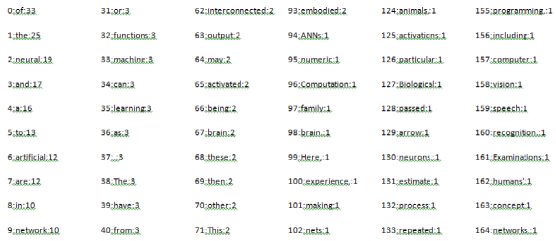


Fig 1 Dataset after step 4.1.3

Step 4.1.4 Document Compression

As shown in Fig 1, using the look-up table and original document as input, encode the original document with their corresponding serial number.

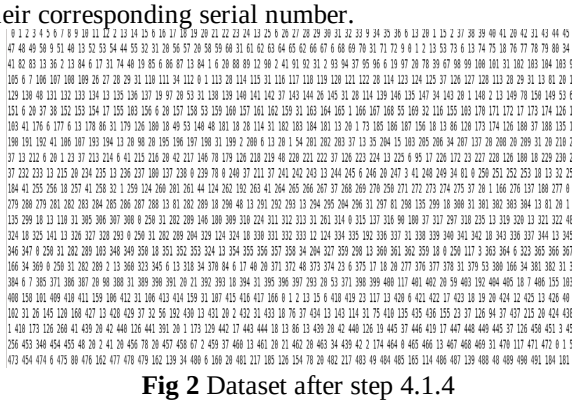


Fig 2 Dataset after step 4.1.4

The original document will be fully encoded with numeric values (corresponding serial number will be replaced to the original term) as shown in Fig 2.

Step 4.1.5 Document Decompression

Decode the encoded document into Original document using Encoded Document and look-up table (corresponding term will be replaced to the serial number). As shown in the Fig 3.

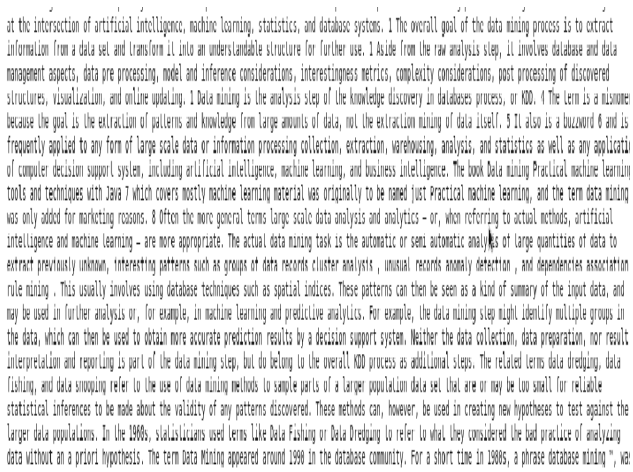


Fig 3 Dataset after step 4.1.5

The decoded output will be in unaligned format, then align is done which results as original document.

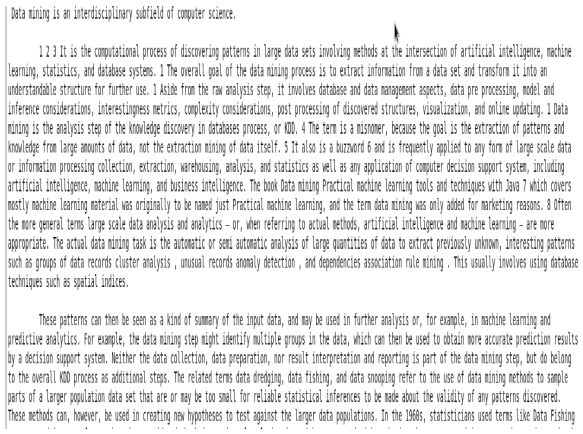


Fig 4 Dataset after Alignment

Step 4.2 Text Categorization

Tokenization and serialization of the unique terms present in training set is carried out, so as to create one look-up table for the entire training dataset.

E.g. Serial Number: Term: Frequency

Remove stop words from the document. Filter top ten high frequency terms from the training data set. These keywords are used to categorize the e-book.

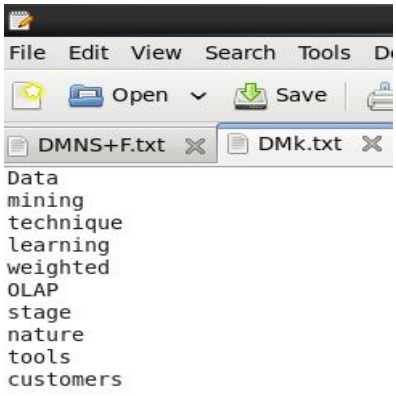


Fig 5 Dataset for keywords

Step 4.2.1 Ranking Documents

Latent Semantic Indexing (LSI); is used, which indexes and is a retrieval method that uses mathematical technique called SVD (Singular Value Decomposition).

Here documents related to the query will be ranked and displayed using a technique called SVD in LSI.

$$\mathbf{X = U \sum V^T}$$

Where X is the Original Matrix,

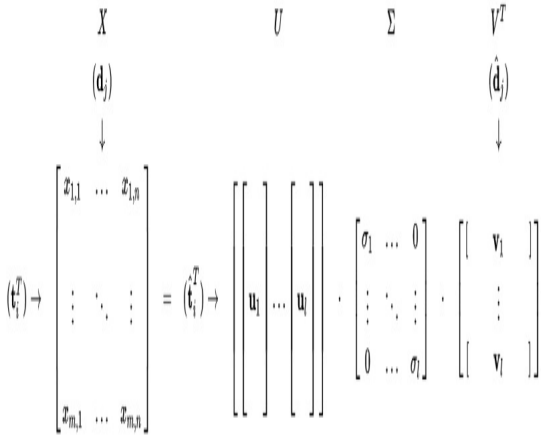
U and V are Orthogonal Matrix

U must contain Eigen vectors of XX^T

V must be the Eigen Vectors of X^TX

$\sum$ - Diagonal Matrix.

The matrix products giving us the term and document correlations then become



Step 4.3 Document Ranking

With respect to user query, documents will be ranked and displayed. This is done with Latent Semantic Indexing using SVD.

Here documents in the cluster will be ranked and displayed based on the user search query as shown in Fig 6.

Query: Data Mining tools and techniques

Document 1: Data Mining for Business Analytics: Concepts, Techniques, and Applications in XLMiner

Document 2 Data mining and linked open data--New perspectives for data analysis in environmental research

Document 3: Data Preprocessing in data mining

Document 4: Spatial Data Mining: Theory and Application

Document 5: Design and implementation of expert clinical system for diagnosing diabetes using data mining techniques

Document 6: Big Data computing and clouds: Trends and future directions

Document 7: A two-step method to construct credit scoring models with data mining techniques

Document 8: Data mining and machine learning in cybersecurity

Document 9: Data-based techniques focused on modern industry: an overview

```
Relevant Document to Query:
1St Level : Doc 1
2nd Level : Doc 2,Doc 4,Doc 5,Doc 6,Doc 8,Doc 10
3rd Level : Doc 3,Doc 7,Doc 9,
```

Fig 6 Dataset after Document Ranking

Step 4.4 Local File System (LFS) Method

For Local File System (LFS) application was developed using java; java provides a system for developing application software and deploying it in cross platform computing environment and allows parallel processing. In LFS sequence of process called text compression, text categorization and text clustering is done using java.

Step 4.5 HADOOP Distributed File System (HDFS) Method

HADOOP is an open source that allows us to store and process data sets in distributed environments across clusters of computers using simple programming models. It is designed to scale up from single servers to thousands of machines which offer locale computation and storage. **MAPREDUCE** is a Programming model which process large datasets in both parallel and distributed environment. This is done using MPI Message Passage Interface.

Map () –Performs Filtering and Sorting.
Reduce () – Performs Summary Operation.

MAPREDUCE does its job in 5 different steps, they are as follows

It Prepare the Map Input, then run the user provided Map code, shuffle the Map output to the Reduce processors, run the user provided reduce code, produce the final output as shown in Fig 5.1.

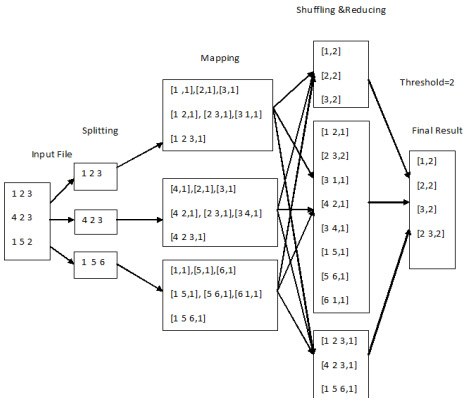


Fig 7 MapReduce Output in HDFS

Step 4.5.1 Running in HDFS

Here in HADOOP big data set were loaded and processed, it uses python programming language.

Input document is feed into HADOOP directory, then MAPPER function is called MAPPER function will map each and every word in the document as shown in Fig 8.

E.g.: (Serial Number, Term, 1)

Data	1
mining	1
is	1
an	1
interdisciplinary	1
subfield	1
of	1
computer	1
science.[1][2][3]	1
It	1
is	1
the	1
computational	1
process	1
of	1
discovering	1
patterns	1
in	1
large	1
data	1
sets	1
involving	1

Fig 8 MAPPER's Output in HDFS

Then Reducer function will be called reducer function will reduce the value mapped by MAPPER class by summing up the count value of similar terms found in the document and store the output in HDFS user directory as shown in Fig 9.

E. g: (Serial Number, Term, Frequency)

part-00000(1)

1	"'[i]n	1
2	"A	3
3	"An	1
4	"Automatic	1
5	"Bad."	1
6	"Big	1
7	"Data	14
8	"Don't	1
9	"Encyclopædia	1
10	"Figure	1
11	"First	1
12	"From	1
13	"Good"	1
14	"Google	1
15	"How	1
16	"Is	2
17	"Judge	1
18	"Knowledge	1
19	"Lesson:	1
20	"Licences	1
21	"Magic	1
22	"Microsoft	1
23	"Number	1
24	"Practical	1
25	"Predictive	1
26	"SIGKDD	1

Fig 9 MAPREDUCED Output in HDFS

5. C2RH ALGORITHM'S

Input: - PDF Document.
Output: - Text Document, Tokenized and Serialized Document, Encoded Document, Decoded Document, Document Ranking, MAPREDUCED WORDCOUNTED Document.

Step 1 Procedure_Convert PDF to Text Document
Step 2 Procedure_To Tokenization and Serialization (WORDCOUNT)

- Read the Text Document.
- Identify the Unique terms present in the Document.
- Calculate the Frequency of the unique terms.
- Serialize the Unique terms along with their frequencies.

Step 3Procedure_To Encode an Document

- Read the Text Document and Identify the Unique Terms.
- Read the Unique Terms present in the WORDCOUNTED Document.
- Replace the Terms present in the Text Document to its corresponding Serial Number present in the WORDCOUNTED Document.
- Now the Text Document is fully encoded with Serial Number.

Step 4 Procedure_To Decode an Document

- Read the Encoded Document and Identify the Unique Numbers.
- Read the Unique Serial Numbers present in the WORDCOUNTED Document.
- Replace the Numbers present in the Encoded Document to its corresponding Terms present in the WORDCOUNTED Document.
- Now the Encoded Document is decoded to Original Text Document.

Step 5 Procedure_To Prune Stop Words
Step 6 Procedure_To Document Ranking

1. Get the Number of terms present in the Query i.e. m=10.
2. Parse the terms in a Query to an array i.e. (Q).
3. Read the number of Text Documents.
4. Parse contents from documents (D).
5. Compare Q and D to X.
6. Find Eigen Value and Eigen Vector for B matrix.
7. CONCAT the Eigen vectors of B to produce U matrix.
8. Find the Diagonal matrix for the square root of Eigen values of B matrix.
9. Repeat Steps 10 and 11 to produce V matrix from C matrix.
10. Find Transpose of U and V matrix.
11. Calculate SVD.
    $X=UDV^T$ so that original matrix can be obtained.

Step 7 Procedure_To implement in HDFS

- Call MAPPER Function, this function will individually map each term's present in the input document.
- Call REDUCER Function, this function will sum the mapped frequency of the same term's.
- Serialize it.

6. C2RH FLOW CHART

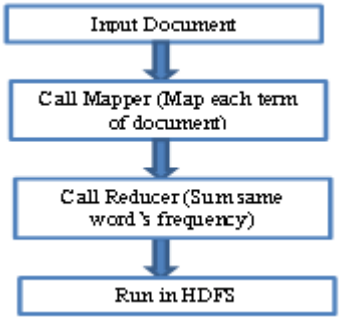


Fig 10 Flow Chart for HDFS

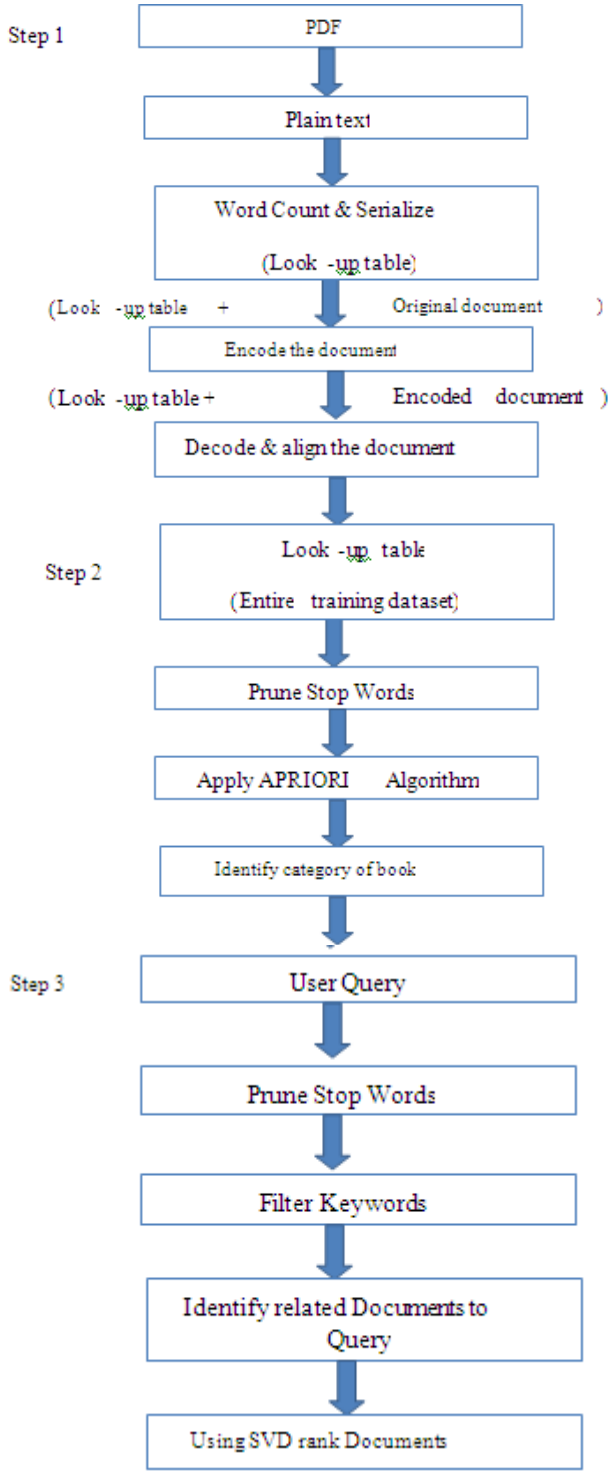


Fig 11 Flow Chart for LFS

7. DATASET USED

E-books are downloaded from Google books https://books.google.co.in/from four category as training and test data they are

- Data Mining
- Computer Networks
- Computer Graphics
- Story Books

Ten Books from each category were downloaded and used for categorization and clustering.

8. RESULTS

Fig 12 A Data Mining E-BOOK of size 174KB taken 22SEC to get serialized and tokenized in HADOOP Distributed File System (HDFS).

```
[hadoop@localhost main]$ time java pg2>oc.txt
ni.txt

real    0m4.709s
user    0m1.870s
sys     0m0.107s
[hadoop@localhost main]$
```

Fig13 A Data Mining E-BOOK of size 174KB taken 04m.709s to get serialized and tokenized in Local File System (LFS).

Table 1: File Size Comparison

Original File Size	Encoded File Size
174k	107.4k
99.7 k	60.4k
70.6k	41.9k

Table 2: HADOOP File System (HDFS) VS Local File System (LFS)

	HDFS	**LFS**
Elapsed Time	22 sec	0m1.870s
Application Type	Map Reduce	STRING TOKENIZER
Final Status	Finished & Success	Finished & Success
User	HADOOP	user

Data Load	Giga Byte (1 GB)	Mega Byte (10 MB)

9. C2RH TIME COMPLEXITY

Let m, n denotes Input Documents

Step 9.1 Text Compression

TC=TC(Document Conversion)+TC(DocumentTokenization andSerialization)+TC(DocumentCompression)
TC=TC(n)+ TC(mn) + TC(mn)
TC=Θ(mn)

Step 9.2 Text Categorization

TC=TC(Tokenization for Entire Training Set)+TC(Prune Stop Words)+TC(Keywords Training)
$TC=TC(n^2)+TC(n^2)+TC(n^2)$
$TC=\Theta(n^2)$

Step 9.3 Document Ranking

TC=TC (Document Ranking)
$TC=\Theta(n^2)$

Step 9.4 HDFS Method

TC=TC (Executing in HDFS)
$TC=\Theta(n)$

10. APPLICATIONS

Compressing text document before stowing it on cloud helps to curtail memory insufficiency crises, clustering of correlated documents aids to categorize documents .Using SVD aids to rank documents based on user query.
In LFS user cannot process bulk documents, whereas in HDFS bulk documents managed meritoriously. MAPREDUCE implements PETABYTE of data in few hours.

11. CONCLUSION

Here E-book held in competent tactic. Document Compressed, Categorized, ranked using various Text Mining techniques.
Document Ranking is done via SVD, using which allied documents ranked and exhibited.In HADOOP MAPREDUCE document was encumbered which handles massive data efficiently.

12. FUTURE WORK

In proposed work, Isolating an category of technical book is done; But isolating an category of non-technical book is fragmentary, technical books categorized based on keywords present in book's, where as non-technical books cannot be categorized based on terms present in book's; it prerequisites surplus data to categorize non-technical books.

REFERENCES

[1]. Feldman, Ronen, et al. "Knowledge Management: A Text Mining Approach."PAKM.Vol. 98. 1998.
[2]. Vaithyanathan, Shivakumar, Mark R. Adler, and Christopher G. Hill. "Computer method and apparatus for clustering documents and automatic generation of cluster keywords." U.S. Patent No. 5,857,179. 5 Jan. 1999.
[3]. Neto, Joel Larocca, et al. "Document clustering and text summarization." (2000).
[4]. Sahane, Manisha, Sanjay Sirsat, and Razaullah Khan. "Analysis of Research Data using MapReduce Word Count Algorithm." Internl.Journal of Advanced Research in Computer and Commn.Engg 4 (2015).
[5]. Liang, Yen-Hui, and Shiow-Yang Wu. "Sequence-Growth: A Scalable and Effective Frequent Itemset Mining Algorithm for Big Data Based on MapReduce Framework." Big Data (BigData Congress), 2015 IEEE International Congress on.IEEE, 2015.
[6]. Wang, Jingjing, and Chen Lin. "MapReduce based personalized locality sensitive hashing for similarity joins on large scale data." Computational intelligence and neuroscience 2015 (2015): 37.
[7]. Nagwani, N. K. "Summarizing large text collection using topic modeling and clustering based on MapReduce framework." Journal of Big Data 2.1 (2015): 1-18.
[8]. Negrevergne, Benjamin, and Tias Guns. "Constraint-based sequence mining using constraint programming." International Conference on AI and OR Techniques in Constriant Programming for Combinatorial Optimization Problems.Springer International Publishing, 2015.
[9]. Feinerer, Ingo. "Introduction to the tm Package Text Mining in R." 2013-12-01]. http://www, dainf, ct. utfpr, edu.br/-kaestner/Min-eracao/RDataMining/tm, pdf (2015).
[10]. El-Halees, Alaa M. "Arabic text classification using maximum entropy." IUG Journal of Natural Studies 15.1 (2015).

Authors

Dr. B. Lavanya, Assistant Professor in the Department of Computer Science, at the University of Madras, has over 16 years of experience in teaching. Has been awarded "Senior Women Educator and Scholar Award", "Outstanding Scientist Award – Data Mining". Has a life membership in Computer Society of India. Has given many inspiring lecture / seminar and invited talks. Also has published papers and research articles in various numbers of International journals.

G. Lalitha, is a research scholar at the Department of Computer Science, University of Madras.