International Journal of Engineering & Technology, 7 (4.10) (2018) 19-23

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper



Elite Sequence Mining of Big Data using Hadoop Mapreduce

P Amarendra Reddy[1], O Ramesh[2]

[1,2] *Department of Information Technology, MLR Institute of Technology, Dundigal, Hyderabad - 500043, India.*
**Corresponding author E-mail: amarpanyala88@gmail.com*

Abstract

Text mining can deal with unstructured information. The proposed work extricates content from a PDF report is changed over to plain content configuration; at that point record is tokenized and serialized. Record grouping and classification is finished by discovering similarities between reports put away in cloud. Comparable archives are distinguished utilizing Singular Value Decomposition (SVD) strategy in Latent Semantic Indexing (LSI). At that point comparative records are assembled together as a group. A similar report is done between LFS (Local File System) and HDFS (HADOOP DISTRIBUTED FILE SYSTEM) as for rate and dimensionality. The System has been assessed on genuine records and the outcomes are classified.

Keywords: *Big data; MAPREDUCE; SVD; LSI.*

1. Introduction

Content's information are of two frame, organized information and unstructured information. Organized information is in the principal typical shape put away in the social databases, while the unstructured or semi-organized information are put away as articles or documents. Text mining handles unstructured or semi-organized reports. Mining content information include certain arrangement of pre-processing steps. In records two sorts of words are available, equivalent words and homonyms. Various types of words that offer a similar significance are called equivalent words and words having same spelling with various importance are called homonyms. In this paper we propose a framework which does content pressure, content classification at long last content bunching. The literary, unstructured archive makes the previously mentioned assignments confounded.

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.

Shivakumar Vaithiyanathan 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

Copyright © 2018 Authors. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

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.

3. Problem Definition

Digital book stop gigantic capacity in cloud, our goal is to spare storage room, by compacting real substance before putting away it on cloud. Content records were scattered in cloud by ordering and grouping related archives together that guides e-reports to be gotten to in proficient way.

3.1. Phases of the Proposed Work

In Step 1 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. TR-OAR Methodology

The Proposed system comprises of the following pre-processing steps

4.1 Text Compression

1. Data Collection: Input documents are chosen from Googel books. (e.g Data Mining and its applications, Data Preprocessing in Data Mining and so on).
2. Data Extraction: The Input document in PDF format is converted to plain text and the text extracted is used for further processing.
3. Document Tokenization and Serialization Unique words from the document are tokenized and serialized along with their frequencies, as shown in Fig 1.

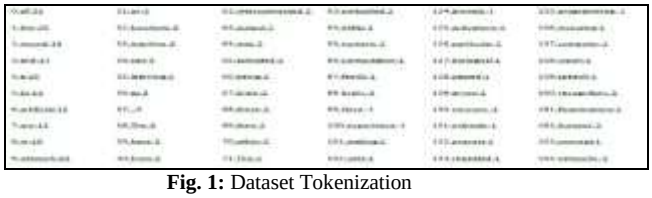


Fig. 1: Dataset Tokenization

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 Document Compression

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.
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. Dataset after decompression. The decoded output will be in unaligned format, then align is done which results as original document.As shown in Fig.4



Fig. 3: Dataset after Document Decompression

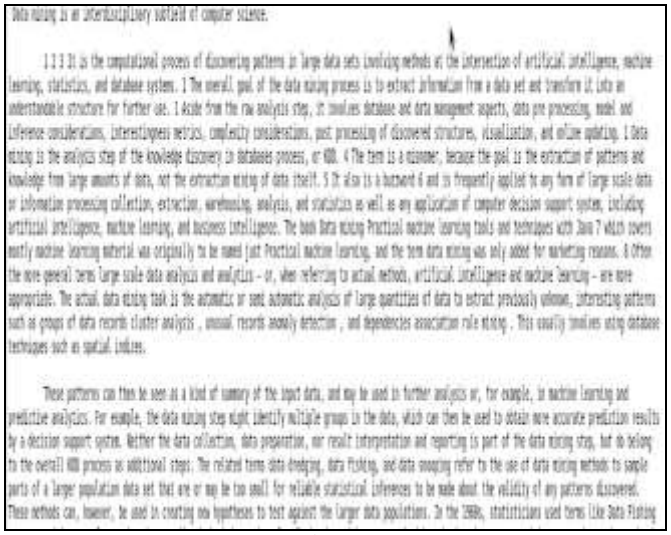


Fig. 4: Dataset after alignment

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

Ranking Document: 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.

$$X = U \sum V^T \quad (1)$$

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 are shown in figure 6.

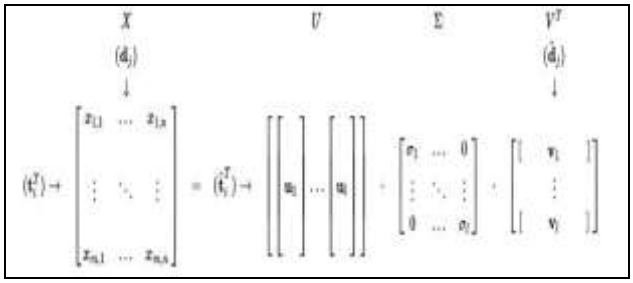


Fig. 6: Text Categorization

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 7.
Query: Data Mining tools and techniques

4.4 Local File System

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.

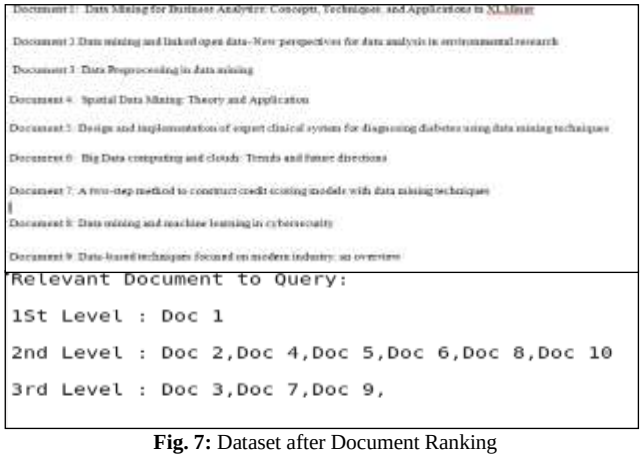


Fig. 7: Dataset after Document Ranking

4.5 HADOOP Distributed File System (HDFS)

HADOOP is a structure for running applications on extensive bunch worked off ware equipment. The Hadoop system straightforwardly gives applications both unwavering quality and information movement. Hadoop actualizes a computational worldview named Map/Reduce, where the application is isolated into numerous little parts of work, each of which might be executed or re-executed on any hub in the group. What's more, it gives an appropriated document framework (HDFS) that stores information on the register hubs, giving high total data transfer capacity over the bunch. Both Map Reduce and the Hadoop Distributed File System are composed with the goal that hub disappointments are consequently dealt with by the structure.
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 shown in Figure 8.

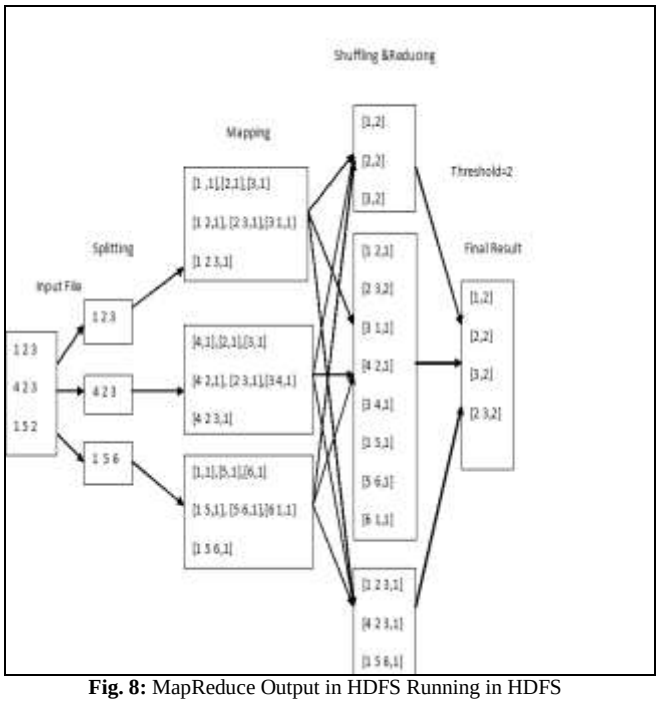


Fig. 8: MapReduce Output in HDFS Running in HDFS

Here in HADOOP big data set were loaded and processed, it uses java programming language.
Input document is feed into HADOOP directory, then Mapper task processes each input record and it generates a new <key,value> pairs. The <key, value> pairs can be completely different from the input pair. In mapper task, the output is the full collection of all these <key, value> pairs as shown in Fig 9.

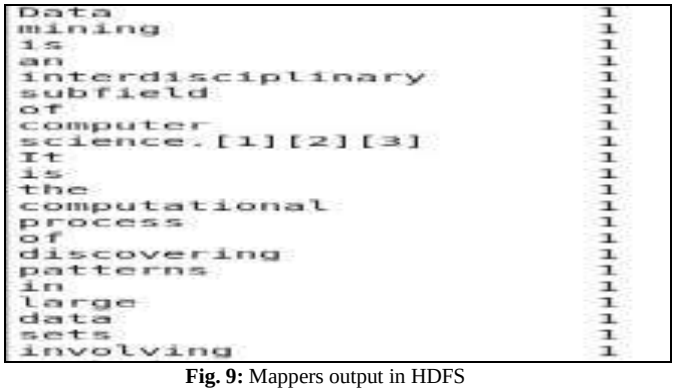
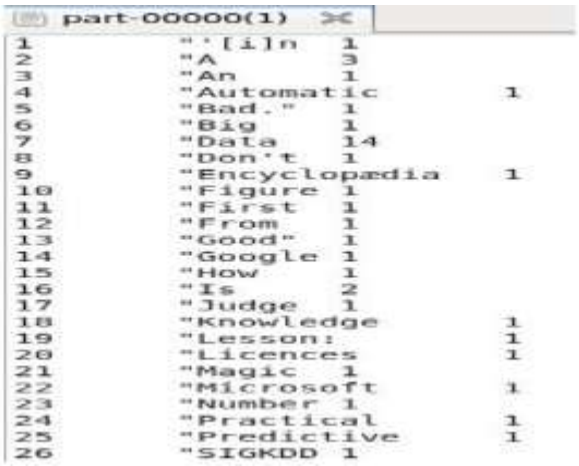


Fig. 9: Mappers output in HDFS

Then Reducer reduces a set of intermediate values which share a key to a smaller set of values. Reducer has 3 primary phases: shuffle, sort and reduce.
Shuffle
Contribution to the Reducer is the arranged yield of the mappers. In this stage the system gets the pertinent segment of the yield of the considerable number of mappers, by means of HTTP.

Sort
The system bunches Reducer contributions by keys (since various mappers may have yield a similar key) in this stage.
The rearrange and sort stages happen all the while; while outline are being brought they are blended.
Reduce
In this stage the lessen technique is required each <key, (rundown of values)> combine in the assembled inputs. The yield of the diminish errand is regularly composed to the FileSystem client catalog as shown in Fig 10.


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. 10: Map reduced output in HDFS

5. TR-OAR (Term Recurrence – Opposite Archive Recurrence) Methodology

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 WORD-COUNTED 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 WORD-COUNTED 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
Get the Number of terms present in the Query i.e. m=10.
Parse the terms in a Query to an array i.e. (Q).
Read the number of Text Documents.
Parse contents from documents (D).
Compare Q and D to X.
Find Eigen Value and Eigen Vector for B matrix.
CONCAT the Eigen vectors of B to produce U matrix.
Find the Diagonal matrix for the square root of Eigen values of B matrix.
Repeat Steps 10 and 11 to produce V matrix from C matrix.
Find Transpose of U and V matrix.
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

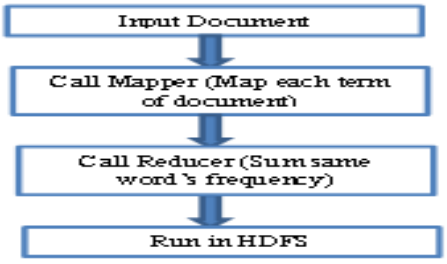


Fig. 11: Flow Chart for HDFS

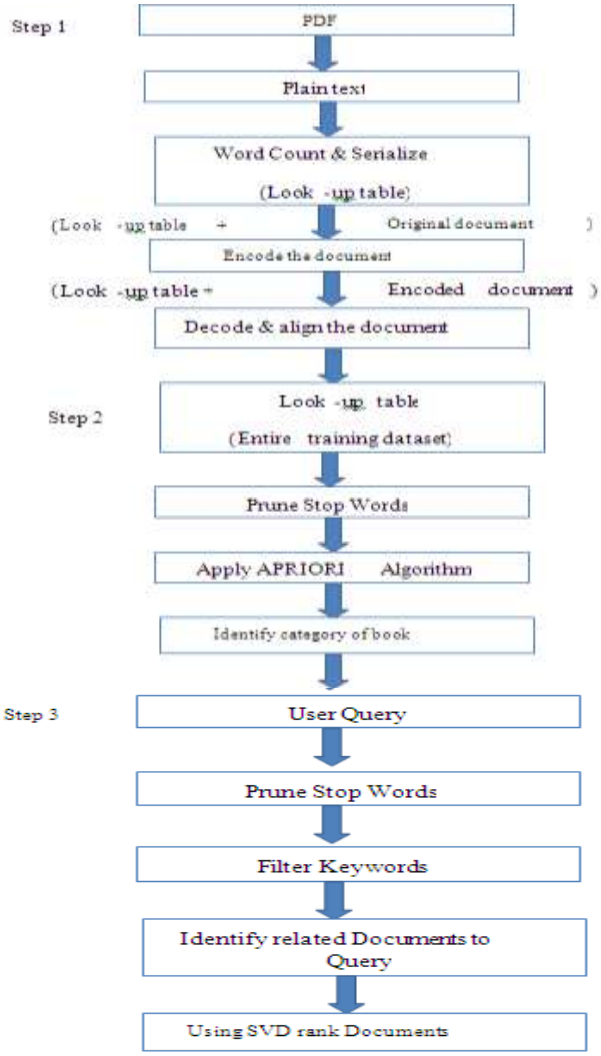


Fig. 12: Flow Chart for LFS

7. Applications

Packing content record before stowing it on cloud reduces memory deficiency emergencies, bunching of related reports helps to order archives .Using SVD helps to rank reports in light of client question.
In LFS client can't process mass archives, though in HDFS mass reports oversaw admirably. MAPREDUCE executes PETABYTE of information in couple of hours.

8. Conclusion

Here E-book held in capable strategy. Record Compressed, Categorized, positioned utilizing different Text Mining systems. Record Ranking is done through SVD, utilizing which united archives positioned and exhibited. In HADOOP MAPREDUCE report was hampered which handles enormous information proficiently.

9. Future Work

In proposed work, Isolating a class of specialized book is done; But separating a classification of non-specialized book is fragmentary, specialized books sorted in view of catchphrases show in book's, the place as non-specialized books can't be arranged in light of terms introduce in book's; it requirements surplus information to classify non-specialized books.

Acknowledgment

The authors would like to acknowledge Science and Engineering Research Board, India for financial support.

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/Mineracao/RDataMining/tm, pdf (2015).