International Journal of Pure and Applied Mathematics
Volume 118 No. 7 2018, 191-198
ISSN: 1311-8080 (printed version); ISSN: 1314-3395 (on-line version)
url: http://www.ijpam.eu
Special Issue
ijpam.eu

Text Mining With Lucene And Hadoop: Document Clustering With Updated Rules Of NMF Non-Negative Matrix Factorization

E. Laxmi Lydia,
Associate Professor
Department of Computer Science Engineering
Vignan's Institute of Information Technology
Andhra Pradesh, India.

D.Ramya,
Junior Research Fellow
Department of Computer Science Engineering
Vignan's Institute of Information Technology
Andhra Pradesh, India.

***Abstract*— The enormous amount of Big Data brought into new era with innovative retrieval of information through analysis. Problem Defined: Massive amounts of data are being collected, stored and Analyzed. mostly the data that exists in documents are unstructed data like text information, log survey results, emails so on. all the data that is dumped on the platform needs systematic arrangement or ordering for robust retrieval and analysis of information. Problem Statement: Proper alignment of document files is to be labelled. when large number of files increases characterizing the files are needed. therefore, here comes the clustering of data i.e., Document clustering. it groups the instances that are unlabeled. Existing system : unstructured text is easily processed and perceived, but is significantly harder for machines to understand. a model is prepared by deducing structures going through Systematic reduce of redundancy to organize the data by similarity. Updated rules of NMF raise a self interest in document clustering. When comparing with the two already existing algorithms i.e, Single value decomposition and Latent semantic Indexing these rules gave trust in its overall performances. Proposed System: In addition to NMF rules a K-means factor is added to give prominent clustering with extracted features. to achieve this in an elaborate sequential steps we have Indexing of Documents, Stop words Removal, Stemming is used to reduce the words to the root that uses most adequate algorithms. In particular for the extraction of features , the text document words need to be identified , algorithm that is used for Key feature extraction and text notation is Natural Language Processing. In this project, the work is distributed parallely among all the documents and that needs running NLP performs parallel pattern. Here the system uses Apache Hadoop Map Reduce for parallel programming.**

Keywords*— *Text mining; Pre-processing; Natural Language process; Document clustering; Map reduce.

I. INTRODUCTION

Today it has turned into an approach to transmit interactive multimedia information by means of the all-pervasive Internet. By means of the imminent electronic trade, it has ended up amazingly vital to handle the delicate issue of bearing information security, particularly in the perpetually zooming open system upbringing of the present day generation[22-24]. Improvement of technology has supported the growth of huge amounts in text documents existed on the internet, organizations, advancement databases, companies so on. the different structure types of information can be unstructured , semi-structured documents. grouping of these unstructured information from the documents is a typical problem for retrieving of the Information. Figure 1 showing the process of text mining.

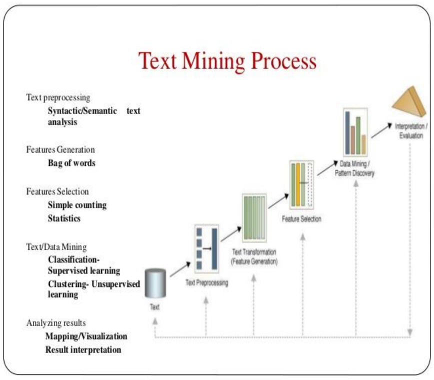


Fig. 1. Text Mining Process

Big Data term does not bother about the size of the data but has some specific terms within it know for volume, variety, velocity determining the reliability and consistency. The data that exists in the internet is mostly unstructured data i.e., about 85% [2]. This performs a challenging retrieval of information in this field.

Handling data and processing data in Big Data area is more efficient by using Divide and conquer [3]. The general objective is to partition the data into smaller sub divisions; these sub divisions are independent and processed using parallel processing. Apache Hadoop being one of the best solutions among all gathered interest to sought data processing, distributed applications, providing high-scalability and storage. Accessing data , maintaining data depends upon the Hadoop Distributed File system (HDFS) [8]. Map Reduce programming rules are the criterion on of the Hadoop HDFS file system.

Automatic extraction of knowledge that is machine readable information is driven from the unstructured natural data. The most demanding step is annotation of text, that describes the grammatical, syntactic, morphological words and phrase[12].

Text annotation mainly extracts keywords and key phrases identifying single words in the document [13]. Keywords annotation is mostly used for extraction of content and its summarization, that produces machine- readable bulk of text. The proposed system shows the implementation of general Natural Language Processing applications, executed via MapReduce.

II. Existing Work

The below description gives information regarding Indexing, text mining, pre-processing document clustering, extraction of features and NMF . Lastly it operates on NMF to cluster document.

A. Text Mining

The hidden information from the texts is extracted in large amounts of documents which are not structured is text mining.

- It can separate information data in enormous content information
- It can get connectivity and correlation with other data
- It identifies the content making use of classification.
- Data processing in computers provides the capacity in high volume of text at high speed.
- Few applications in extraction of knowledge from data are information extraction, detection basing on topic and tracking, definitions of that text, arrangement of text, clustering, concept linkage, information visualization etc.

Supervised learning is used to train data in particular data that are already labeled. Here, the machine contains a new set of examples (data) so that supervised learning algorithm determine that training data (set of training examples) and generates a correct outcome from labelled data.

Unsupervised learning is used to train data that are not labeled. Here machine performs to group unsorted information corresponding to the correlations, patterns and odds without any previous training of data.

B. Pre-Processing

The pre-processing step acts a main role for the entire process of clustering and extracting process. Sentences are sub divided in preprocessing in terms of tokens. This paper, it produces simple words, word stemming of suffixes so that words having the same root(e.g., dance, dances, dancer and dancing) finds the same single word for counting of frequency. Stop lists used to filter the non-scientific English words. Figure 2 is demonstrating the pre-processing steps and figure 3 demonstrating an example of stemming

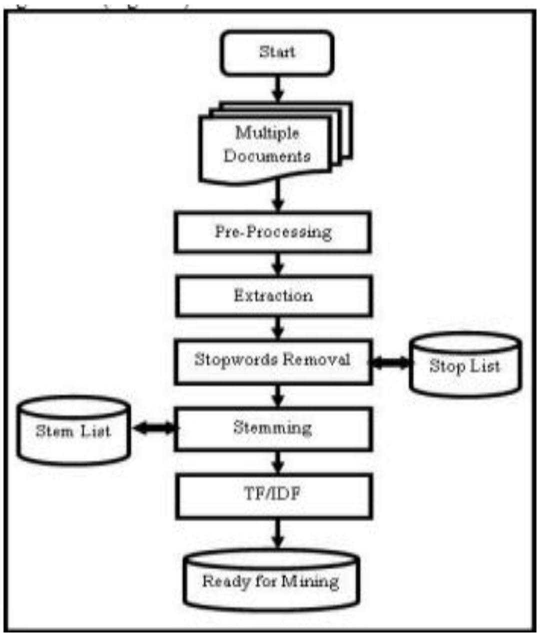


Fig. 2. Pre-processing Steps

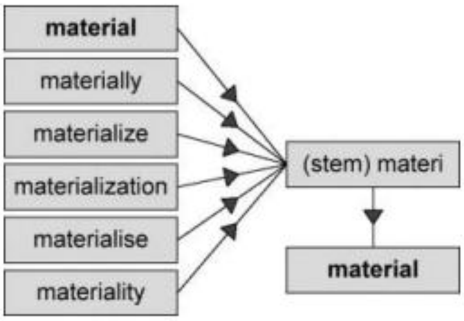


Fig. 3. Example of Stemming

C. Clustering of Documents

Clustering of documents is simply described as “clustering of documents’. Clustering is a process to identify the likeness and unlikeness/ distances between the extracted words from the document. Separating them into meaningful sub divisions by sharing similar characteristics. As clustering drops under the category of unsupervised learning, it predicts the documents and arranges them into particular class. Figure 4 demonstrating document clustering in a text document.

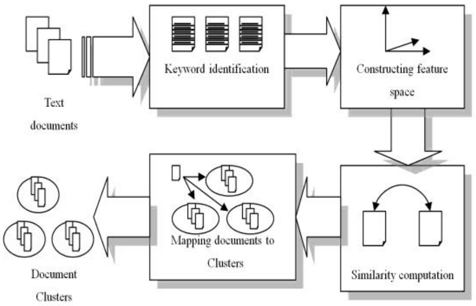


Fig. 4. General document clustering in text documents.

D. Nonnegative matrix factorization (NMF)

One of the important algorithms for distributed parallel processing and storage in memory for computing factorization for non negative values is Non-negative Matrix Factorization algorithm. Here the algorithm decomposes the data by parts of data matrix small similar parts of that information.NMF algorithm checks for the positive values in matrix by using factorization. Authors Lee and seung in 2001, suggested for the multiplicative updation of algorithm, it uses the factors of every non-negative data matrix i.e., W and H as two factor matrices. Correlating each factorization is specified by user in terms of rank. Defined terms H and W are rows and columns, where these are the samples given to the clusters existing. We start with the random approach of the matrices and use a interactive approach for cost measurement by factor matrices that reaches optimal solutions locally. Later H and W are used to patterns evaluation also. Most oftentimes NMF is used discover classes by identification of patterns. These can be best usage of clusters.

E. KNMF

KNMF clustering of documents is given by the likeness among the features that are identified from the individual document.

1) The methodology adapted

- Initially here we initiate a term document matrix naming as V by calculating term frequency inverse document frequency from the folder files .
- After acquiring of vector V values, the length is normalized to unit length using Euclidean distance.
- Now on basing Lee and Seung [2] calculations on vector V , we get the factors W and H [1].

Finally we go for similarity measures between the documents and the derived features of W. also assign document to Wax if the angle between them is minimum or very small by using cosine similarity. This process is similar to single turn in k- means algorithm.

2) Steps in indexing the document in a folder

Initially when any document is given to process here indexing plays a vital role of arranging each document orderly. the document that is given at start checks whether the document is new or already existing one. If it is new , it provides updation in non-updated document list.

- Once the document is updated or the given document is updated one we go proceeding to next steps without any new modifications in the indexing of that particular document and if the document is not modified then we delete the old document by creating new document. Extract the words from the document.
- Next we go for stop word removal to avoid unimportant words.
- Here important algorithms by using stemming is applied for better capturing of root to the word in the document is given.
- All the collected information till now from the document is stored in particular index.
- Finally all the displaced i.e, stray files are been removed from the folder.

For the parallel execution of algorithm k-means, Hadoop has given a platform to run in local reference mode and also in pseudo distributed mode. The job that is done by k-means is given to the client i.e., job client. Time that the jobs completed are mentioned properly.

III. System Architecture

Hadoop is an open source software which allows users to manipulate the huge data chosen for efficient , scalable results allowing distributed parallel processing to improve performance giving data integrity and manipulating failures in Java.

Here in the Hadoop Distributed File System clusters are stored or installed in multi-node. Each cluster poses a Name node master, that allocates jobs to different Data nodes(clients), while looking all the execution process it also monitors the failures that come across. Storing of data is declared by the data nodes. Figure 5 shows the framework that resides in the HDFS.

The results from the NLP are written into SQL external database. The prospective architecture of this Framework takes input into two complete file paths: here the first points are located on the local file and the different modules work asynchronously; the keyword/key phrases are scheduled for the execution process and results separately.

Finally, a constant systematic procedure saves all the extracted keywords and key phrases in the database (SQL). entire architectural procedure is followed in java.

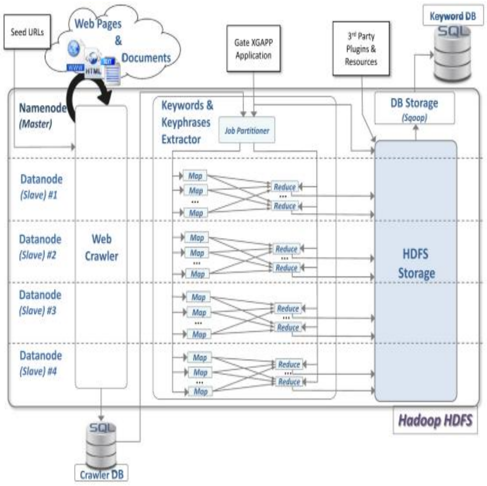


Fig. 5. Hadoop Distributed File System Frame Work.

Figure 5 is demonstrates the entire modeled system architecture. Functions namely 'Map' and 'Reduce' in the keyword & key phrase Extraction module can be built by the user, basing on the NLP tasks executed from the input documents . A pair of key/value is produced by the Map function and analyzed basing on the domain referring to the parsed text. The TF-IDF computes in accordance with the Reduce function. Storage of all the keywords from extraction, Key phrases and related Meta data in SQL database.

A. MapReduce

The foremost best processing technique in Hadoop is MapReduce which divides the process into smaller jobs in executing. here functions namely Map and Reduce act as functions and implement the functions accordingly, that defines the specific executed input data.

Hadoop splits the input data into fixed sizes basing on the block size i.e,64MB. Job that is assigned by the MapReduce is to split using Key-value pairs because it gives the output values using arrays, lists and also numerical values, strings.

The main implementation that involved in Map function is represented by the logical records from the input source. Different shuffle and sort process combines all intermediate values added with the similar key .The sorted output pairs are reduced to process the splitted data. Here Figure 6 shows an example of an inverted word index creation through MapReduce.

Since Hadoop is worked with Java, the below screens shoots has been generated after executing in the Big data analytics cluster which was created as phase1 of the project. Sample Dataset from Newsgroup20 is considered and screen shoots through experimental setting are taken. Figure 7 screenshot shows the piece of Input data taken from newsgroup20.

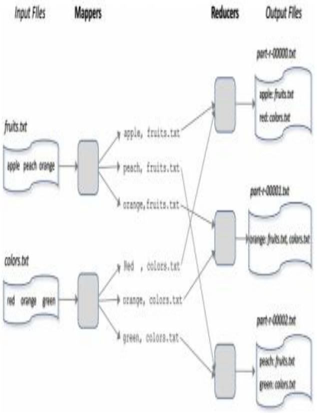
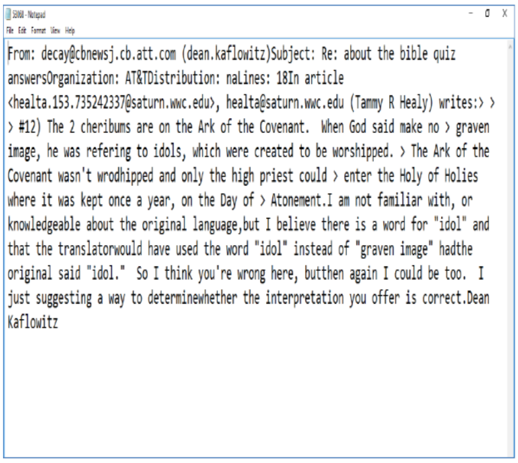


Fig. 6. Example for mixed-up word index generation in MapReduce.

Figure 8 shows the adding of the given input file into HDFS. it also shows execution of Stopwords program by creating stopwords.jar file along with some paths needed for compiling.

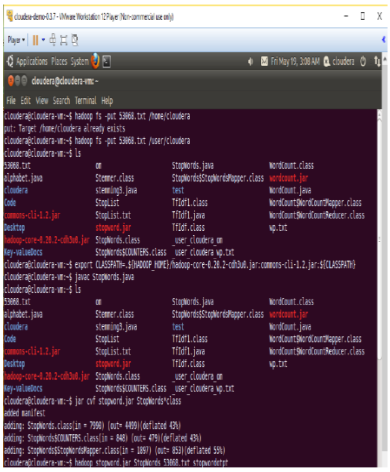
Figure 9 shows the output screen for compiling of Stopwords program with the required Stopwords count in a given input file.

Figure 10, 11, 12 screenshots shows the values of Iterated Lovins Stemming, Lovins stemming, Porter Stemming output. Figure 13 shows If-Idf count values of sample input data files



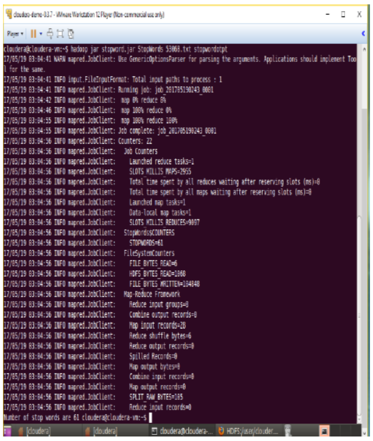
```
From: decay@cbnewsj.cb.att.com (dean.kaflowitz)Subject: Re: about the bible quiz
answersOrganization: AT&TDistribution: naLines: 18In article
<healta.153.735242337@saturn.wwc.edu>, healta@saturn.wwc.edu (Tammy R Healy) writes:> >
> #12) The 2 cheribums are on the Ark of the Covenant.  When God said make no > graven
image, he was refering to idols, which were created to be worshipped. > The Ark of the
Covenant wasn't wrodhipped and only the high priest could > enter the Holy of Holies
where it was kept once a year, on the Day of > Atonement.I am not familiar with, or
knowledgeable about the original language,but I believe there is a word for "idol" and
that the translatorwould have used the word "idol" instead of "graven image" hadthe
original said "idol."  So I think you're wrong here, butthen again I could be too.  I
just suggesting a way to determinewhether the interpretation you offer is correct.Dean
Kaflowitz
```

Fig. 7. Sample Dataset from Newsgroup20



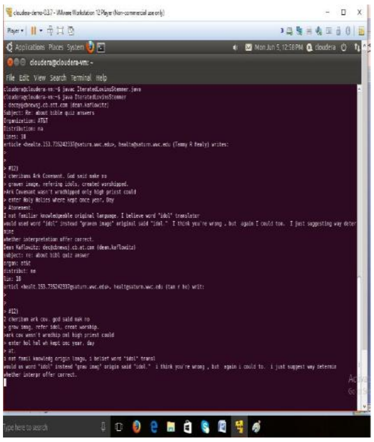
```
cloudera@cloudera-vm:~$ hadoop fs -put 53068.txt /home/cloudera
put: Target /home/cloudera already exists
cloudera@cloudera-vm:~$ hadoop fs -put 53068.txt /user/cloudera
cloudera@cloudera-vm:~$ ls
53068.txt                 om                      StopWords.java                 WordCount.class
alphabet.java             Stemmer.class           StopWords$StopWordsMapper.class wordcount.jar
cloudera                  stemming3.java          test                           WordCount.java
Code                      StopList                TfIdf1.class                   WordCount$WordCountMapper.class
commons-cli-1.2.jar       StopList.txt            TfIdf1.java                    WordCount$WordCountReducer.class
Desktop                   stopword.jar            TfIdf.class                    wp.txt
hadoop-core-0.20.2-cdh3u0.jar StopWords.class     _user_cloudera_om
Key-valueDocs             StopWords$COUNTERS.class _user_cloudera_wp.txt
cloudera@cloudera-vm:~$ export CLASSPATH=.${HADOOP_HOME}/hadoop-core-0.20.2-cdh3u0.jar:commons-cli-1.2.jar:${CLASSPATH}
cloudera@cloudera-vm:~$ javac StopWords.java
cloudera@cloudera-vm:~$ ls
53068.txt                 om                      StopWords.java                 WordCount.class
alphabet.java             Stemmer.class           StopWords$StopWordsMapper.class wordcount.jar
cloudera                  stemming3.java          test                           WordCount.java
Code                      StopList                TfIdf1.class                   WordCount$WordCountMapper.class
commons-cli-1.2.jar       StopList.txt            TfIdf1.java                    WordCount$WordCountReducer.class
Desktop                   stopword.jar            TfIdf.class                    wp.txt
hadoop-core-0.20.2-cdh3u0.jar StopWords.class     _user_cloudera_om
Key-valueDocs             StopWords$COUNTERS.class _user_cloudera_wp.txt
cloudera@cloudera-vm:~$ jar cvf stopword.jar StopWords*class
added manifest
adding: StopWords.class(in = 7990) (out= 4499)(deflated 43%)
adding: StopWords$COUNTERS.class(in = 848) (out= 479)(deflated 43%)
adding: StopWords$StopWordsMapper.class(in = 1897) (out= 853)(deflated 55%)
cloudera@cloudera-vm:~$ hadoop stopword.jar StopWords 53068.txt stopwordotpt
```

Fig. 8. Adding the input file to HDFS and creating stop- words jar file



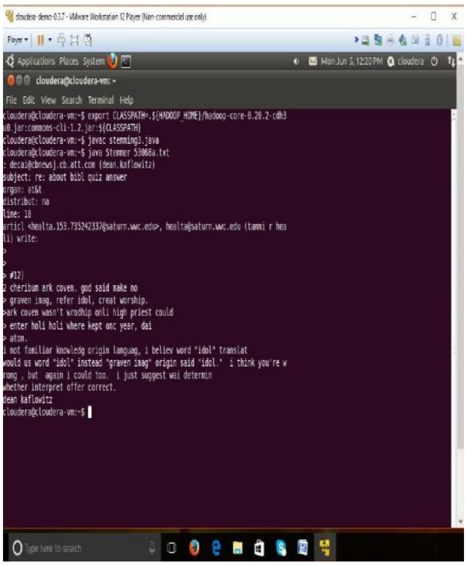
```
cloudera@cloudera-vm:~$ hadoop jar stopword.jar StopWords 53068.txt stopwordotpt
17/05/19 03:04:41 WARN mapred.JobClient: Use GenericOptionsParser for parsing the arguments. Applications should implement Too
l for the same.
17/05/19 03:04:41 INFO input.FileInputFormat: Total input paths to process : 1
17/05/19 03:04:41 INFO mapred.JobClient: Running job: job_201705190243_0001
17/05/19 03:04:42 INFO mapred.JobClient:  map 0% reduce 0%
17/05/19 03:04:46 INFO mapred.JobClient:  map 100% reduce 0%
17/05/19 03:04:55 INFO mapred.JobClient:  map 100% reduce 100%
17/05/19 03:04:55 INFO mapred.JobClient: Job complete: job_201705190243_0001
17/05/19 03:04:56 INFO mapred.JobClient: Counters: 22
17/05/19 03:04:56 INFO mapred.JobClient:   Job Counters
17/05/19 03:04:56 INFO mapred.JobClient:     Launched reduce tasks=1
17/05/19 03:04:56 INFO mapred.JobClient:     SLOTS_MILLIS_MAPS=2955
17/05/19 03:04:56 INFO mapred.JobClient:     Total time spent by all reduces waiting after reserving slots (ms)=0
17/05/19 03:04:56 INFO mapred.JobClient:     Total time spent by all maps waiting after reserving slots (ms)=0
17/05/19 03:04:56 INFO mapred.JobClient:     Launched map tasks=1
17/05/19 03:04:56 INFO mapred.JobClient:     Data-local map tasks=1
17/05/19 03:04:56 INFO mapred.JobClient:     SLOTS_MILLIS_REDUCES=9097
17/05/19 03:04:56 INFO mapred.JobClient:   StopWords$COUNTERS
17/05/19 03:04:56 INFO mapred.JobClient:     STOPWORDS=61
17/05/19 03:04:56 INFO mapred.JobClient:   FileSystemCounters
17/05/19 03:04:56 INFO mapred.JobClient:     FILE_BYTES_READ=6
17/05/19 03:04:56 INFO mapred.JobClient:     HDFS_BYTES_READ=1068
17/05/19 03:04:56 INFO mapred.JobClient:     FILE_BYTES_WRITTEN=104048
17/05/19 03:04:56 INFO mapred.JobClient:   Map-Reduce Framework
17/05/19 03:04:56 INFO mapred.JobClient:     Reduce input groups=0
17/05/19 03:04:56 INFO mapred.JobClient:     Combine output records=0
17/05/19 03:04:56 INFO mapred.JobClient:     Map input records=28
17/05/19 03:04:56 INFO mapred.JobClient:     Reduce shuffle bytes=6
17/05/19 03:04:56 INFO mapred.JobClient:     Reduce output records=0
17/05/19 03:04:56 INFO mapred.JobClient:     Spilled Records=0
17/05/19 03:04:56 INFO mapred.JobClient:     Map output bytes=0
17/05/19 03:04:56 INFO mapred.JobClient:     Combine input records=0
17/05/19 03:04:56 INFO mapred.JobClient:     Map output records=0
17/05/19 03:04:56 INFO mapred.JobClient:     SPLIT_RAW_BYTES=105
17/05/19 03:04:56 INFO mapred.JobClient:     Reduce input records=0
Number of stop words are 61 cloudera@cloudera-vm:~$
```

Fig. 9. Output screen showing the count of stop words in a given input file through Map Reduce phase.



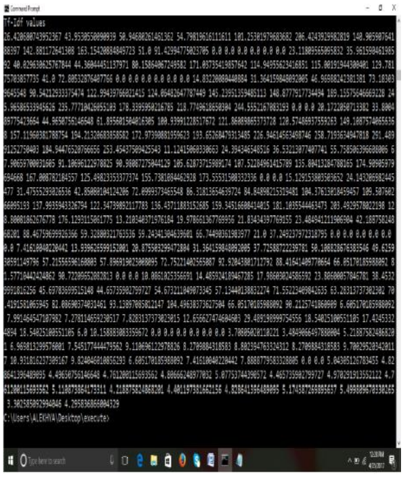
```
cloudera@cloudera-vm:~$ javac IteratedLovinsStemmer.java
cloudera@cloudera-vm:~$ java IteratedLovinsStemmer
: decoy@cbnewsj.cb.att.com (dean.kaflowitz)
Subject: Re: about bible quiz answers
Organization: AT&T
Distribution: na
Lines: 18
article <healta.153.735242337@saturn.wwc.edu>, healta@saturn.wwc.edu (Tammy R Healy) writes:
>
>
> #12)
2 cheribums Ark Covenant. God said make no
> graven image, refering idols, created worshipped.
>Ark Covenant wasn't wrodhipped only high priest could
> enter Holy Holies where kept once year, Day
> Atonement.
I not familiar knowledgeable original language. I believe word "idol" translator
would used word "idol" instead "graven image" original said "idol."  I think you're wrong , but  again I could too.  I just suggesting way deter
mine
whether interpretation offer correct.
Dean Kaflowitz: dec@cbnewsj.cb.at.com (dean.kaflowitz)
subject: re: about bibl quiz answer
organ: at&t
distribut: na
lin: 18
articl <healt.153.735242337@saturn.wwc.edu>, healt@saturn.wwc.edu (tam r he) writ:
>
>
> #12)
2 cheribum ark cov. god said mak no
> grav imag, refer idol, creat worship.
>ark cov wasn't wrodhip onl high priest could
> enter hol hol wh kept onc year, day
> at.
i not famil knowledg origin langu, i belief word "idol" transl
would us word "idol" instead "grav imag" origin said "idol."  i think you're wrong , but  again i could to.  i just suggest way determin
whether interpr offer correct.
```

Fig. 10. Output Screen for Lovins Stemming Algorithm



```
cloudera@cloudera-vm:~$ export CLASSPATH=.${HADOOP_HOME}/hadoop-core-0.20.2-cdh3
u0.jar:commons-cli-1.2.jar:${CLASSPATH}
cloudera@cloudera-vm:~$ javac stemming3.java
cloudera@cloudera-vm:~$ java Stemmer 53068a.txt
: decai@cbnewsj.cb.att.com (dean.kaflowitz)
subject: re: about bibl quiz answer
organ: at&t
distribut: na
line: 18
articl <healta.153.735242337@saturn.wwc.edu>, healta@saturn.wwc.edu (tammi r hea
li) write:
>
>
> #12)
2 cheribum ark coven. god said make no
> graven imag, refer idol, creat worship.
>ark coven wasn't wrodhip onli high priest could
> enter holi holi where kept onc year, dai
> aton.
i not familiar knowledg origin languag, i believ word "idol" translat
would us word "idol" instead "graven imag" origin said "idol."  i think you're w
rong , but  again i could too.  i just suggest wai determin
whether interpret offer correct.
dean kaflowitz
cloudera@cloudera-vm:~$
```

Fig. 11. Output Screen for Porter Stemming Algorithm.



```
Tf-Idf values
26.420680743952367 43.9530650090939 50.9460261461362 54.7981961611611 101.25301979683682 206.4243929982819 140.905907641
88397 142.881172641308 163.15420084049723 51.0 91.42994775023705 0.0 0.0 0.0 0.0 0.0 0.0 23.11809565095832 35.961590461985
92 40.029630625767844 44.36044451137971 80.15864067249582 171.03735419857642 114.94955623416851 115.00191944300401 129.781
75703857735 41.0 72.80532876407766 0.0 0.0 0.0 0.0 0.0 0.0 14.83220008440884 31.364159840092005 46.96988242381381 73.18303
9645540 90.54212933375474 122.99439766021415 124.06482647787449 145.23951359485113 148.8777917734494 109.15575646669228 24
5.96586533945626 235.77710420055103 178.3395950216705 218.7749618650304 244.5552167083193 0.0 0.0 20.1722050713382 33.8004
89775423664 44.9650756146648 61.895601504016305 100.93991228517672 121.86009865373728 120.57406937559263 149.1087574065636
8 157.11960381788754 194.21320683850582 172.97390881959623 193.65268479313485 226.94614563498746 258.7193634947818 291.489
91252750403 104.94476520766656 253.45437509425543 11.12415060330663 24.394346548516 36.53213077407741 55.75850639660806 6
7.90659700031605 91.10696122978825 90.90087275044129 105.62873715989174 107.52284961415789 135.80413284788165 174.98905979
694668 167.008782184557 125.49823353377374 155.7381004462928 173.55531500332336 0.0 0.0 15.129153803503652 24.143206982445
477 31.47555293026536 42.85080104124206 72.0999373465548 86.31813654639724 84.84098215319401 104.37613018459457 109.507602
66095193 137.9935943326794 122.34739892117783 136.43711883152685 159.3451600414015 181.1035544463473 203.4929578022198 12
8.80081862676778 176.1293115061775 13.210340371976184 19.978661367769956 21.83434397769155 23.484941211906904 42.188758248
68201 88.46759699926366 59.32800321763536 59.24341304639601 66.74490361903977 21.0 37.249237972318795 0.0 0.0 0.0 0.0 0.0
0.0 7.41610040220442 13.939626599152001 20.875503299472804 31.364159840092005 37.72588722239781 50.10082876383546 49.6259
30591149796 57.21555961608003 57.89691902300095 72.75221402565807 92.92043801712792 88.41641409770664 66.05170185988092 8
1.57710442424862 90.72209652082813 0.0 0.0 10.0861025356691 14.485924109467285 17.98603024586592 23.80600057846781 38.4532
9991816256 45.69783699515148 44.65735902799727 54.673211049073345 57.13440138832274 71.55223409842635 63.28313737302302 70
.4191581065945 82.00690374031461 93.13097085812147 104.49638373627504 66.05170185988092 90.2125741860909 6.605170185988092
 7.991464547107982 7.278114659230517 7.8283137373023015 12.656627474604603 29.489198999754556 18.54025100551105 17.4245332
4894 18.54025100551105 6.0 10.158883083359672 0.0 0.0 0.0 0.0 0.0 0.0 3.70805020110221 3.4849066497800004 5.21887582486820
1 6.969813299576001 7.545177444479562 9.110696122978826 8.2709084318583 8.802394763324312 8.2709084318583 9.70029520342011
7 10.931816237309167 9.824046010856293 6.605170185988092 7.41610040220442 7.8088779583328005 0.0 0.0 5.04305126783455 4.82
8641396489095 4.49650756146648 4.761200115693562 4.80666240977032 5.07753744390572 4.465735902799727 4.970291913552122 4.7
61200115693562 5.110873864173311 4.218875824068201 4.401197381662156 4.828641396489095 5.174387269895637 5.499809670330265
 3.302585092994046 4.295836866004329
C:\Users\ALEKHYA\Desktop\execute>
```

Fig. 12. Output Screen showing TF-Idf values of sample input files taken.

IV. Result Analysis

Here, in this section elevates the performance strategies of different stemming algorithms like Iterated Lovins Stemming Algorithm, Lovins Algorithm, and Porter Stemming Algorithm. In the execution of Stop words, we illustrated different steps in removal of Stop words in Document clustering. Basing on the factors obtained from the text document we say that it is free of cluster and reduces the size of the document. The analysis metrics considered here are : Index Compression Factor (ICF), Word Stemming Factor (WSF), and Correct Stemming Word Factor (CSWF).

Index Compression Factor (ICF): It defines the percentage of the total number of apparent words to that of number of apparent stems after stemming. The strength of the stemmer increases along with this ICF value.

$$ICF=(N-S)/S\times100$$

Words Stemmed Factor(WSF): It defines as percentage of words that have been stemmed by the stemming process out of the total words in a sample, strength of Stemming increases along with number of words stemmed.

$$WSF=WS/TW \ \times100$$

Correct Stemming word Factor(CSWF): It defines as percentage of words that have been stemmed correctly out of the number of words stemmed. The accuracy of the stemmer increases with increased percentage of CSWF.

$$CSWF=CSW/WS \ \times100$$

Figure 14,15,16 are graphically represented by considering three documents (doc1, doc2, doc3) and compared among three stemmer algorithms that are considered.

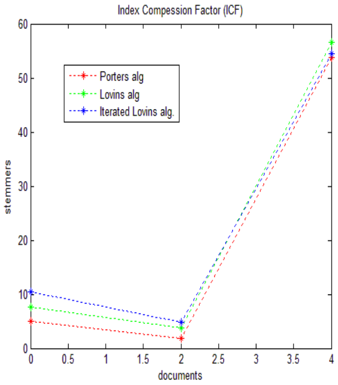


Fig. 13. Graph showing ICF(Index Compression Factor) values using Stemmers

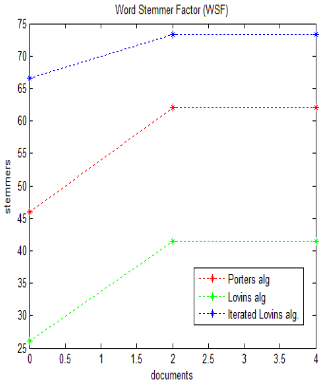


Fig. 14. Graph showing WSF(Word Stemmer factor) values using Stemmers

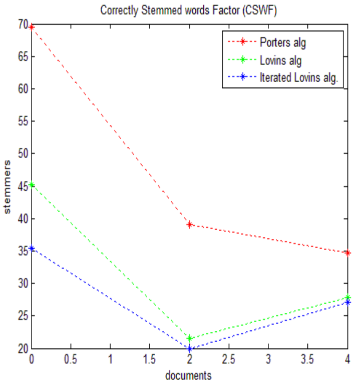


Fig. 15. Graph showing CSWF(correctly Stemmed words factor) values using Stemmer

V. Conclusion

To operate huge data “Big Data” approaches gave maintenance from the database management tools and traditional data processing applications. A new processing techniques mainly in stemming algorithms that is Iterated Lovins Stemmer algorithm has given better results when compared to Porter Stemmer and Lovins stemmer algorithm. and a new algorithm KNMF which is furthurly used and application named as "Text mining Lead". Therefore with those defined characteristics of KNMF help to cluster the documents by the K-means clustering labels. Likewise a parallel implementation with Map Reduce for huge sized documents lead to minimize time computation and increase the average computation speed. The entire process from token generation to clustering, Every process is done in Apache Hadoop.

Funding Acknowledgement

This work is financially supported by the Department of Science and Technology(DST),Science and Engineering Research Board (SERB) under the scheme of ECR. We thank DST,SERB for the financially support to carry the research work.

References

[1] A.K.Jain,”Data Clustering”(beyond 50 years): K-Means, Pattern Recognition”, 2010,Page No.:651-666.

[2] H.Zhang, J.E.Fritts, et,al.,”Image Segmentation Evaluation: Survey of Unsupervised Methods”,Computer Visual Image Understand., 2008, Page No:260-280.

[3] R.Baeza Yates,B.RibeiroNeto,et.al., “Modern Informaton Retrieval”,1999,ACM Press,New York.

[4] D.J.Miller, Y.Wang,et,al.,”Emergent unsupervised clustering paradigm with potential application to bioinformatics”,2007,Front Biosci.,page no:677-690.

[5] NeelimaGuduru,”Text Mining with support vector machines(SVM) and Non-Negative Matrix Factorization (NMF) Algorithm”, 2006,Master Thesis,University of Rhode Island, C S Department.

[6] M.W.Berry, S. T. Dumais and G.W. O’Brien,” Using Linear Algebra for Intelligent Information Retrieval”, CS-94-270, December 1994, SIAM Rev., 37(4), Page No.:573-595.

[7] Thomas K Landauer, Peter W.Foltz, et.al., “Introduction to Latent Semantics Analysis”,1998, Page No.:256-284.

[8] DD Lee& Seung H,"Learning the parts of Objects by Non-Negative matrix factorization(NMF)",1999, Nature- Volume 401, Page No.: 788-791.

[9] DD Lee &Seung H,"Algorithm for Non- Negative Matrix Factorization", T.G.Dietterich and V.Tresp editors,"Advances in Neural Information Processing Systems",Volume 13, Proceedings of the Conference: 556562, the MIT Press.

[10] Chris ding,Xiaofeng He,Horst D.Simon,"On the Equivalence of Non-Negative Matrix factorization(NMF) and Spectral Clustering",Proceedings in SIAM International Conference on Data mining, Page NO.:606-610.

[11] Wei Xu,Xin Liu and Yihong Gong,"Document Clustering based on Non-Negative matrix factorization", Proceedings in ACM SGIR, 2003,Page No.: 267-273.

[12] Yang CF,Ye M and Zhao J, " Document clustering based on Non-Negative Sparse Matrix Factorization, 2005,International Conference on Advances in Natural Computation, Page No:557-563.

[13] KhushbooKanjani,"Parallel Non-Negative matrix factorization for Document Clustering",2007,CpSC-659 Spring 2007 course project.

[14] Porter M F,"An Algorithm for Suffix Stripping", 1980, Volume 14,NO.:3, Page No.: 130-137.

[15] Julie Beth Lovins,"Development of a Stemming Algorithm", March & June 1986, MTCL,Volume 11,No.1 and No.2.

[16] CH V T E V Laxmi,Dr.K. Somasundaram,"2HARS: Heterogeneity-Aware Resource Scheduling in Grid Environment using K-Centroids Clustering and PSO techniques",2015, IJAER Journal, ISSN 0973-4562 Volume 10,No.7,Page.No: 18047-18062.

[17] Dr.E.Laxmi Lydia, Dr.M.BenSwarup, Dr.ChallaNarshimham," A Disparateness- Aware Scheduling using K-Centroids Clustering and PSO techniques in Hadoop Cluster".

[18] Cutting, D,Karger,D.Pederson,J &Tukey,J(1992).Scatter/gather: A cluster-based approach to browsing large document collections, In Proceedings of ACM SIGIR.

[19] Porter, MF (1980), "An Algorithm for suffixing stripping", Program,Vol.14,No.3,Pages 130-137 http://tartarus.org/~martin/porterStemmer/def.txt

[20] Key Phrase Extraction Algorithm (KEA) http://www.nzdl.org/Kea/

[21] Ding, C,He X, &Simon, HD(2005), on the equivalence of Non negative matrix Factorizationand spectral Clustering. Proceedings in SLAM International Conference on Data mining, Pages 606-610.

[22] Shankar, K., and P. Eswaran. "RGB based multiple share creation in visual cryptography with aid of elliptic curve cryptography." China Communications 14.2 (2017): 118-130.

[23] Shankar, K., and P. Eswaran. "RGB-Based Secure Share Creation in Visual Cryptography Using Optimal Elliptic Curve Cryptography Technique." Journal of Circuits, Systems and Computers 25.11 (2016): 1650138.

[24] Shankar, K., and P. Eswaran. "An Efficient Image Encryption Technique Based on Optimized Key Generation in ECC Using Genetic Algorithm." Artificial Intelligence and Evolutionary Computations in Engineering Systems. Springer, New Delhi, 2016. 705-714.