November-December 2019
ISSN: 0193-4120 Page No. 1107 - 1130

Automatic Document Clustering and Indexing of Multiple Documents Using KNMF for Feature Extraction through HADOOP and LUCENE on Big Data

E.Laxmi Lydia
Professor & Big Data Consultant,Computer Science and Engineering,Vignan's Institute of Information Technology, India. E-mail: elaxmi2002@yahoo.com

N.Sharmili
AssociateProfessor in CSE Department, GayatriVidyaParishad College of Engineering for Women, Viskahpatnam, Andhra Pradesh,India.

PhongThanh Nguyen
Department of Project Management, Ho Chi Minh City Open University, Vietnam.
E-mail: phong.nt@ou.edu.vn

WahidahHashim
Institute of Informatics and Computing Energy, UniversitiTenagaNasional, Malaysia.
E-mail: wahidah@uniten.edu.my

AndinoMaseleno
Institute of Informatics and Computing Energy, UniversitiTenagaNasional, Malaysia.
E-mail: andino@uniten.edu.my

Article Info
Volume 81
Page Number: 1107 - 1130
Publication Issue:
November-December 2019

Abstract

The existence of unlabeledtext data in documents has become larger and excavating such datasets is a provocative task. The objective of Big Data is to store, retrieve and analyse multipletext documents. *Problem Statement:*The retrieval of the identical data over large databases is of major concern. *Existing Solution:*Existing problem is solved by Full-Text Search (FTS) which means pattern matching technique that allows searching of multiple keywords at specific time.*Proposed Solution:* In this paper, we consider multiple text documents as input and processed using text mining pre-processing algorithms like Key Phrase extraction, Porters stemming for tokenizing and TF_IDF toobtain all non-negative values. These values further processed to get matrix data throughNonnegative matrix factorization (NMF). On performing NMF, K-means algorithmis upgraded with NMF to obtain quality clusters of data sets.Performances of the algorithms

are tested using Newsgroup20 data in Open Source Hadoop software environment which also analyses the performance of the MapReduce framework. The final outcome is to generate clusters and index them for the Newsgroup20dataset. Later on, Apache Lucene is presented for automatic document clustering with aGUI interface developed for indexing. Thus, this proposed algorithm resultsby improving the performance of document clustering through Map Reduce framework in Hadoop.

Article History
Article Received: *3 January 2019*
Revised: *25 March 2019*
Accepted: *28 July 2019*
Publication: *25 November 2019*

Keywords*: Unlabeled, Big Data, Extraction, Porters, Hadoop, Map Reduce.*

I INTRODUCTION

The emerging mechanisms of social network platforms, modern techniques for transforming general information into digital format are operated as bits and bytes. Working platform areas like wearable devices, sensors access internetalong with generated chucks of data to enhance business and its functioningfor continual process.As per the 21st century, the basic need and use of data over the internet form different sources around the world cannot not be blocked.This is completely transformative technological platform. The developed data has to be outperformed for proper storage using categorical classification, clustering and regression techniques.

Growth of Big Data in cloud computing is being emerged with multiple sources using multiple processors till the halfway of 2000s. Most used applications through internet like Facebook reported that daily it's been activated by 955 million mobile users from different countries having more than 70 languages uploading 260 billion photos each day and maintaining billions of friend connections and transferring 40 billion data. For every 60 minutesmore number of likes and comments are generated. This has lead for the establishment of new era to study overwebsites andcultural dynamicssocial media

A Big Data search application working for several documents performs common operations like

- Acquiringraw Content: It obtains the document details on which such application needs to be organized.
- Framing the document: Obtained raw content and the search application arepredicted and interpretedquickly.
- Analysing the document: For analysis only spotted documents are collected and indexed.

November-December 2019
ISSN: 0193-4120 Page No. 1107 - 1130

- Indexing the document:For the instant retrieval of the document, entire document content will be allotted with keys. This process simulates the normal indexes in books tracked by page numbers,but not exactlyto the searching process in folders.
- User Interface is initiated when the index database is ready for searching.
- Searching query, the index database is verified to access relevant details of the documents.
- Finally, Results are collected.

II Literature Review

B. MeenaPreethi et al, [1], explained the applications and issues for digital data.The speed in technology hasadded onwith its volume of data. The structure of popularly used data, overworldwideis unstructured data, to restore and progress the unstructured data. An issue has been raised toscientists as pattern identification is mandatory. To progress the unstructured data, text mining play a major role for preprocessing, text transformation, selections of features, mining of data, evaluation of data. Finally,it will achieve large amounts of text documents through an application. Therefore, proper selection of technique will improve the document usage efficiency and speed.

Meiping Song et al, [2]have studied that data applied on images to calculate the non-negative factorization values based on spatial and abundance constraints.Mixtures of pixels are organized throughrestraints of spatial resolution and consequences obtained against object recognition and classification. The pixels are analyzedin linear mixture model, nonlinear mixture model, Bi-Linear mixture model in spectral unmixing technique. Itdirects to analyze the main materials and finds the comparable fractions generated from hyperspectral imagery of a location. Additionally, NMF is the best implemented model for linear spectral mixture. It discovers the edge points and determine the abundances at the same time. Through the compression or extraction of data, pixels are directly decomposed local minimum and reduce the convergence. The authors have implemented the NMF algorithm by considering it as new constraint based smoothness and extraction of features with effectiveness.

DipeshShrestha[3] suggested the purpose of sentiment analysis has reached immense considerationin text mining. With unstructured format of data, abundant noiseremoval will has to be done which is expensive to remove. Rapid growth in modern methodologies and software tools, are used to preprocess the noisy data using sentiment analysis.Phases like data acquisition, preprocessing, feature extraction and representation labelling of data. Algorithms like Natural Language Processing as well as machine learning algorithms are implemented. Dipesh Shrestha applied these algorithms to the unstructured text documents for sentiment analysis in text mining.

Yu-Xiong et al, [4] proposed and implemented highly accurate clustering algorithm like K-Means and PSO for optimization technique.Clustering data have reduced lots of major issues while retrieving data. K-means algorithm has grabbed lots of attention among all the clustering algorithms. This clustering algorithm k-means have resolved and identified its own advantages and disadvantages through clustering data.While performing k-means, initialization of cluster centers plays a major role.

This can be improved by hybridization of the algorithm. Most advanced optimization algorithm Particle Swarm Optimization (PSO)merge with k-means algorithm to boost its efficiency. Here the authors have proposed algorithms for cluster analysis using Benchmark datasets through machine learning repository.

JiaQiao et al, [5] have studied various clustering algorithms related to data mining in database management system. The issue obtained for the practical usage of data and collection of data in data mining involves clustering with three effective data mining technologies. Among them one of the most effective, easy, simple and practical data mining, clustering algorithm is K-means. Different clustering algorithms are more real-time based algorithms for quality clusters. They have compared clustering algorithms for most suitable data (mostly unstructured). Certainly, the recognition of clustering will be identified in different platforms like industries, business, healthcare, etc.

E. Laxmi Lydia et al, [6] proposed NMF in big data platform for document clustering. They have considered software tools like Apache Hadoop and Apache Lucene to process multiple text documents. They have identified the problem with massive data in unstructured form. Data that has been dumped into the database or system needs to be classified accurately and aligned properly. It will group the unlabeled data in clusters. A model that has been already in existence like SVD and LSI has reduced the data in a systematic manner. Addition to them, an updated rules of NMF brought raise in document clustering with an interest. They have concentrated on the NMF rules that supports the k-means clustering algorithm. In the specific context, the text documents are preprocessed and implemented based on Key feature extraction and text notation in Natural Language Processing. Later, results are generated using distributed parallel implementation through Hadoop.

SerhatSelcukBucak et al, [7][20]recommended the incremental clustering study through Nonnegative Matrix Factorization.After all NMF`s recommends new samples for batch nature requirements. It is a tool for representation of data in online processing onmassive data sets. One of the issue raised while implementing NMF is defining actual number of clusters in partitioning methods. It will determine the rank of the categorization with significant clustering achievement.It estimates the optimal rank selection problem. Results are tested and obtained by manifesting the linearly separable data for clustering. Multi viewing clusters using NMF.

Abhay Kumar et al, [8] implemented k-means clustering algorithm for modeling. It is a theoretical illustration ofthe real world mechanisms. Anticipating the similar characteristic methods in the data. The most prevailing techniques for finding hidden pattern algorithms are classified into two approaches, unsupervised (clustering) and supervised (classification). Where classes are not defined for unsupervised and classes are defined for supervised.They have focused on clustering algorithms by considering the numerical results using probability density function algorithm where objects are identified and placed into similar groups called clusters. This algorithm is used to predict weather data reports. Prediction class labels are either yes or no. This has proven that k-means clustering algorithm is efficient.

Chengbin Peng et al, [9] considered and designed nonnegative matrix using multiplicative approach. NMF will maintain parts-based data and representation of datathrougha combination of algorithms by considering factorization. Here they have focused and implemented multiplicative algorithm. NMF using multiplicative algorithm has been broadlysupported in fields like text mining, clustering, denoising, so on. This algorithm recognizes the factorization evaluation to perform linear constraints related to individual factors. Suppose the constraint function is linear,then NMF performs multiplicative framework to merge with algorithm. Depending on the constraints from larger domains, the NMF will facilitate.

Jie Tang et al, [10] suggested new practicesfor supervised and unsupervised data based onNonnegative matrix factorization.NMF leads to new matrix decomposition method, acts like an practical tool for vast data processing and analysis. This has primarilyanalyzed and introduced that NMF algorithmsrelyon a basic system. It is performed to decrease the dimension of the actual matrix. To cluster, the encoded matrix is generated from the actual document. While generating matrix it consumes more time as well as storage, but such issues were overcome by using NMF. Atlast experiments are obtainedwith better clusters.

E.Laxmi Lydia et al, [11][18]described a thorough study on Non-Negative Matrix Factorization (NMF) as a standard criterion for dimensionality reduction.NMF organizes non-negativity to obtain the representation using a parts-based and simultaneously improving relation among the raised issues. They studied NMF extensions and its reasoning consistently. Different categories of the NMF and their architectural design, characteristic principles, implementation, issues are described and evaluatedwidely. Furthermore, various applicational areas where NMF can be implemented are discussed.It can also be implemented for systematic analysis for spectral clustering.[19] NMF acted as the solution for unsupervised polyphonic signal separation.

E. Laxmi Lydia et al, [12]described that maximum text documents need to be clustered by using open source software to generate identical documents obtaining lower complexity. They have implemented Partitional Clustering and for clustering process and estimated the similarity distances among k-means clusters and centroids.In [13], they have calculated all the values which are positive by calculating distance and by avoiding negative values for effective implementation of NMF algorithm.

Lee and Sung's[14] practiced their execution ofupdatingthe rules of NMF to achieve automatic document clustering. To help in obtaining non-negative values, they have followed a sequential series of steps like removal of similar clutter/key Phrase extraction for stop words and stemming algorithm. Finally, performed the parallel implementation using the MapReduce framework in Hadoop. In [17], NMF has processed to generate document topics by encoding matrix

MohitBhansali, et al [15] has worked on a search engine using Lucene for generating proper searching process through requested queries related to indexing as well. They tested system for indexing and searching. Through the analysis and performance of the Lucene software they have concluded that it has been highly reliable toolkit for searching and indexing documents.

November-December 2019
ISSN: 0193-4120 Page No. 1107 - 1130

Jimmy Lin et, al. [16] has dealt with full-text for indexing by scanning entire text documents throughHadoop for inefficient features while processing. As Hadoop stores the data in blocks, here the search engine will compress the data in blocks through byte offset. This helps the user to have a fast retrieval process.

This paperdevelopstop terms in documents by clusteringusing NMF along with K-means.Terms are identified using TFIDF values. Later, theNMF features and generated clusters results on the document topics. Finally, perform indexing to arrange documents in a folderusing Lucene for better searching.

III Methodology

3.1 Preprocessing by Stop words removal

Stoop word removal is a step to remove noise from the document. Whenever, the files or text documents are taken as input. Those files are taken for pre-processing. Pre-processing is done to remove the noisy data, stopwords. Following is the procedure for Stop words algorithm:

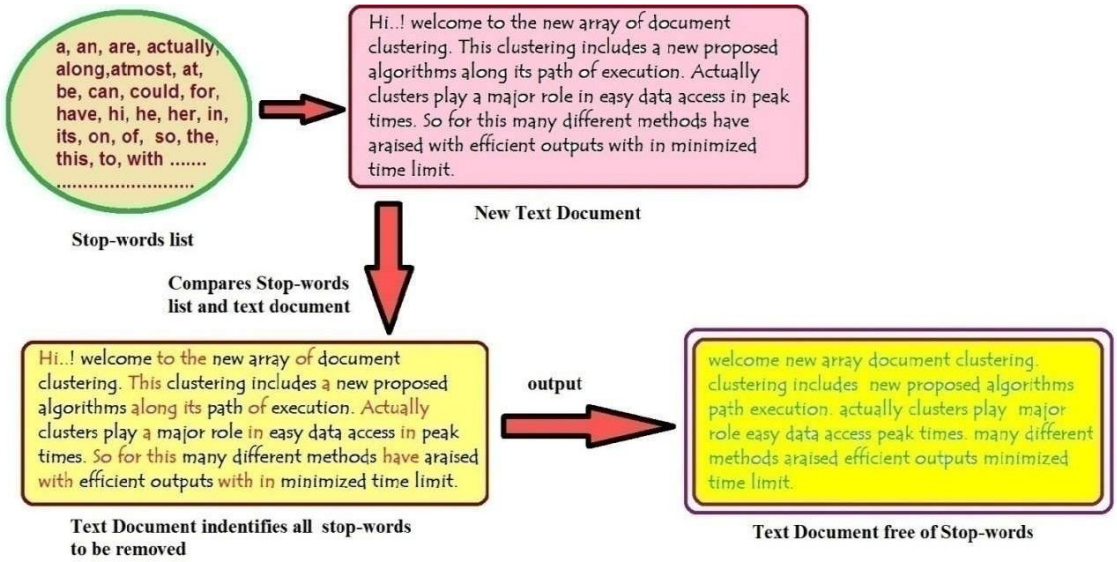


Figure1: Illustration of stop words removal

Stop words are common words which are sifted throughpreceding. A stop word is a generally utilized word in our day by day life, that an internet searcher has been customized to overlook, both while seeking and while recovering them accordingly. It will recognize the generally weighted words called catchphrases for reports to lessen the measurements of the network. Stop word end is performedbyASCII estimations of entire letters without bothering the case (either lower case or capitalized) and calculate the every letter comparing ASCII esteem for individual word and outcome

them as result. Allocate number to relating word, and keep them in sorted request. Here is an example illustrating theprocess which shown in figure1.

3.2 Pre-processing using Stemmer Algorithm

Stemming is implemented in pre-processing to reduce the document words to root. Basis or root or stem isusually written forms of word form. Stemmer algorithm identifies words in ing, er, etc. for elimination. It has derivative ending and inflexible algorithm to shorten word. The suffixes and prefixes were suppressed in stemming algorithmunder certain laws.

To minimize transformed words to their stem, base or root, the words ending with “ing”, er, and so on refer to a word stems. A stemmer is to abstract and inflect derived endings so that word forms to convert asa stem. Through the employed process, the conditions are checked and continue to remove the suffixes and prefixes of the words. Following are the stemming algorithms

3.2.1LovinsStemmerAlgorithm

Lovins'sstemmer was first efficient stemmer. It is analysedand designed specifically to develop stemming algorithms to reduce word. The algorithm is collected in four pieces in four Appendices A, B, C and D together in its Structural. Part A has listed with 294 terminations, each letter that restored using condition for termination or not to terminate the word. Following are the Lovinsrules for the words:

- All the text words that ends with 11 letters, 10 letters, 9 letters, so on till words ending with 1 letter are applicable to use Lovins stemmer
- Suppose,Alistically(a 11 letter word ends with ly) is represented by B; Antialness (a 10 letter word ends with ss) is represented by A; Allically(a 9 letter word ends with ly) is represented by C;A(a 1 letter word A) is represented by A
- There are twenty - nine such conditions, called A , Z, AA, BB and CC, are part B of the stemmer. They are here (* is a letter)
- A is determined with no restrictions; Bfixes the least possible length of the stem set by 3; C fixes the minimumlength of the stem restore by 4; D fixes the least possible length of the stem restore by 5; E remains same by not removing the end after e; Ffixes the least possible length

of the stem restoreby 3 and remains same by not removing the end after e; G determines the minimum length of the stem restoreby 3 and eliminatethe end only after f; H eliminate the end t or ll; I remains same by not removing the end after o or e; J remains same by not considering the end after a or e; Kfixes the least possible length of the stemrestoreby 3 and eliminate the end l,i or u*e , L remains same by not considering the end after a, c, e or m.Same way for Z remains same by not considering the end after f.

- AA eliminates the last letteronly after d,f,ph,th,l,er, or, es or t; BB fixes the least possible lengthof the stem restore by 3 and remains same by not considering the last letter after met or ryst; C eliminate the end only after 1.

Among 294 endings, we often use 259 the left our 35 use 23 applicable conditions having less than 2 suffixes

Part C in Lovins stemmerallows a collection of 35 processing applicable rules for adapting the letters at the prefix of the stem. The rules for letters intend and replaces letters Bb with b; Ll with L; Mm with M; Lev with Lef; umpt with um; Rpt with Rb;Urs with Ur; Olv with olut; lx with lc; Uad with uas likewise.

Eventually, part D shows some flexibleapplicable rules to match query terms to index terms. Whenever stemmer usesan IR system setup, we may consider it not as afactor of the stemmer.

3.2.2 PorterStemming Algorithm

Characteristic messages of the dialect usually maintainvariouschangesfor a essential word. The moral variations (e.g. PLAY, PLAYS, PLAYED, PLAYING, PLAYER) are generally recognized by differing sources, which involvesmeaningful orthographic alternatives, misspellings and rendering andpruningmodifications. If it were probable to merge the variations for a given word so they rediscovered from an enquiry that initially defines clearly about solitary variation.

Morphological variation occurs on the right - hand side of the word - frame in English and many related dialects (Sproat, 1992), which has impeded the use of customer coordinated right - hand truncation to recover data.This way of dealing with conflation is exceptionally straightforward but one which requires impressive expertise since two notable kinds of mistakes ispossible. Overcut occurs when a stem remains too short after truncation and it can produce absolutely irrelevant words that are combined with the same root, as both MEDICAL and MEDIA are recovered through root

MED *.Under cutting, on the other hand, if a string is too short, and associated words may be depicted by different strings, like BIBLIOGRAPHICALLY being cut into BIBLIOGRAPHIC, rather than shorter - rot BIBLIOGRAPHY. Also includes 35 rules (Lovins, 1968) to be computed. Despite the large number of postfixes, few are usually plural structures, and both the additions and the recoding axioms suggest that the Lovins calculation is mainly designed for preparing logical documents (Porter, 2005). Secondly, the use of a solitary device, combined with the treatment of the environment. Many of Lovins ' delicate tones identify with the length of the stem that is left after a postfix has been evacuated: the negligible adequate length usually only has two signs with a risk of critical overlap.

3.2.3 Proposed Iterated LovinStemmer

The most popular algorithms for textual - language stemming are the Porter and Lovins.These two algorithms apply heuristic conditions to delete or modify suffixes in English.The stemmer of Lovins is more characterized than the Porter stemmer. The Lovins stemmer admits that same stem can be assigned to two words, but that two distinct words are correspondent to be mis - mapped into the same stem (Krovetz 1993).We have discovered that for keyphrase removal the aggressive stem is better than the conservative ones. Named as Iterated LovinsStemmer, the algorithm performs thestemmer repeatedly until the term changes.

3.2.4 Comparison on threeStemmers

The comparative study on Lovinsstemmer is more effective than the Porter’s stemmer. Lovinsstemmer maps the complementary words with the stems (Neurology, Neurologist) of the words, yet conditions like (beauty, beautiful) make mistakes.This paper uses Iterated Lovins stemmer algorithm, whichrepeats the Lovins stemmer procedure again and again continuously till the word remains same without changing.For instance, the Lovins stemmer generates "Sci" as result when "Scientist" is given as input. When "Sci" is given as input, "Sc" as output is generated. It shows that the given word “Scientist," the Iterated Lovins algorithm creates "Sc" as an output. In this way, the advancing of any stemmer algorithm will necessarily increase The Algorithm of the Iterated Lovinsidentifies that ‘Science’ and' Scientific' has alike root, while those words are mapped by the other stemmer in separateroots.

The resultant of the stemmer algorithm are further carried out to the pre-processing step of finding TF-IDF values. The obtained values fromTF-IDF are maintained in a separate file in matrix format. Term which has higher values considers as the important term in the document.

The generated TF-IDF matrix is now provided as input to NMF. The TF-IDF matrix is converted into terms and weight matrix by Non Negative Matrix Factorization. To extract cluster terms, NMF algorithm is been combined with pattern values.In addition, we implement the K - means algorithm slightly. This enables user to choose the number of clusters for the data. Following figure 2describes the entire process of document clustering with clear representation. It text document sets as input, pre-process it using algorithms to achieve exact term, extract it and define number of clusters to generate.

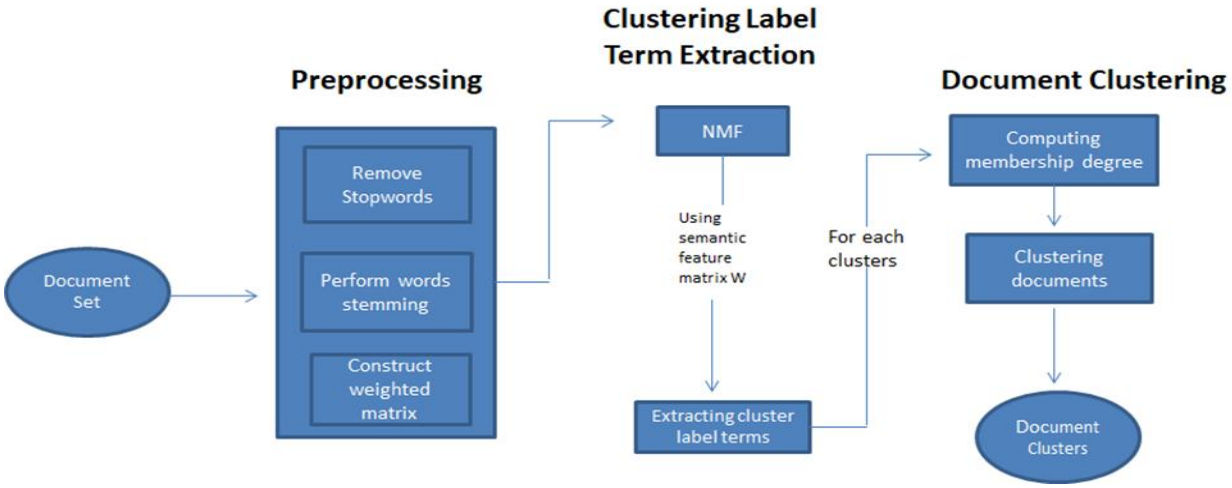


Figure2: Procedure of proposed document cluster

3.3 Pre-processing through Term Frequency-Inverse Document Frequency:

Tf-IDF is usually calculated to find the weight of the term. This weight is a numerical computationto evaluate the significance of a word. The importance of the word is identified by appearance of theterm appeared in the document, it is termed as frequency of the word. It uses two calculations to find the frequency of the term. Term Frequency (TF) and Inverse Document Frequency (IDF)

- **Term Frequency,** measures the existence of term how often appears in a document. It is possible that any term may occur more than once or once. It is therefore often divided by the total number of terms in the document. i.e,

November-December 2019
ISSN: 0193-4120 Page No. 1107 - 1130

	Doc 1	Doc 2	Doc 3	Doc 4
Term-a	2		1	3
Term-b	3	4	1	3
Term-c	2			

$$TF = \frac{Numberoftimestermsappearsinadocument}{Numberoftermsint\quad hedocument} \quad (1)$$

- **Inverse Document Frequency,** measures the importance of the term. Every term in the document is equally important when computing TF. However, some terms, such as' is," of,' and' that' may appear many times, but won't have any meaning. i.e,

$$IDF(t) = \log_{10} e \frac{Totalnumberofdocument}{Numberofdocumentwit\quad htermtinit} \quad (2)$$

Following is a table1, contains terms and documents. Suppose, there are three terms (term--a, term-b, term-c). Every term is checked in all documents as well as every term existence in single document.

Table1: Existence of terms in the documents

The Term-a is present in three documents (Doc1, Doc3,Doc4), the Term-b is present all documents and the Term-c is available at only onedocument (Doc1). The overall weight of Term-a is now equal to number ofdocument(s) and total termdocument(s), reg(4/3)= 0.124. Similarly, for Term-b log(4/4)=0 also known as global terminal weight and Term-c as log(4/1)=0.601. Therefore, the weight shows the weight of the term in the document.

- **Interpretation of TF-IDF calculation**

TF_IDF = TF*IDF (3)

Table2: Documents with term and term frequency

Document	Term	Term Frequency
Doc1	she	1
	is	1
	a	2
	girl	1
Doc2	she	1
	is	1
	playing	2
	chess	3

From table 2 textdata, term weight is calculated for, "she" and "chess"

Tf("she", doc1) = 1/5 = 0.2
Tf("she", doc2) = 1/7 = 0.14
Idf("she", DOCS) = log(2/2) = 0
Tf-Idf("she", doc1) = 0.2 * 0 = 0
Tf-Idf("she", doc2) = 0.14 * 0 = 0

As TF-IDF the word "she" is 0, which implies that the word is not so informative as it appears in all documents.

Tf("chess", doc1) = 0/5 = 0
Tf("chess", doc2) = 3/7 = 0.429
Idf("chess", DOCS) = log(2/1) = 0.301
Tf-Idf("chess", doc1) = 0 * 0.301 = 0
Tf-Idf ("chess", doc2) = 0.429 * 0.301 = 0.13

"Chess" TF-IDF value is obtained, so the "chess" term in document 2 is informative.

3.4 CLUSTERING

Clustering is a mechanism of grouping similar objects in one division. Clustering is classified as

- *Document Partitioning (Flat Clustering)*

Itseparates documents into sub-division clusters. Various methods like k-mean clustering, probabilistic clustering of the Naira Bayes or Gaussian model, Latent Semantic Indexing (LSI), Non - negative Matrix Factorization (NMF).

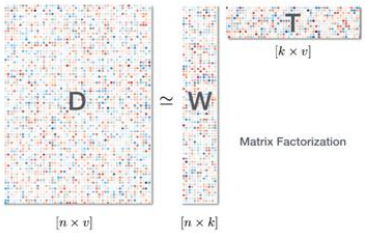
- *Hierarchical clustering*

Clusters are found in successive document clusters, the present method uses a bottom - up or top - bottom approach (divisive).

In this paper, Non Negative Matrix Factorization(NMF) is applied to cluster the documents.

3.4.1 Non Negative Matrix Factorization(NMF)

NMF is anexact type of matrix factorization where the non-negative values has limitation based on the lower rank matrices.Itfollows the multiplication matrix for linear combinations of computations.



November-December 2019
ISSN: 0193-4120 Page No. 1107 - 1130

The factorization of a term document matrix X is split into two non-negative matrices W and H which are responsible for non-negative matrix factorization (NMF) calculation.

It breaks V_{mn} matrix toward the product of two lower-ranking W_{mk} and H_{kn} matrices, so that V_{mn} is about equal to W_{mk} times H_{kn}.

$$V_{mn} \approx W_{mk} \cdot H_{kn} \quad (4)$$

Here, all entries in A, W, and H have to be non-negative, and because we usually impose a low rank on W and H, precise factorization is consistently rare, so we have to settle for an approximate factorization where WH is near A. However, despite the inaccuracy, the low range of W and H forces the solution to describe A using fewer parameters that tend to find underlying patterns in A. These underlying patterns are what make NMF useful and applicable.

Where the application relies on k<<min(m, n). K declares the topics need to be extracted from the documents through clustering. V shows the relation between terms and documents. W maintains weight columns connecting columns as feature vectors or base vector in W.

Thus, the linear combination of the base vectors from W weighted by the corresponding columns from H can compose each document vector from th i be any document vector from matrix V, W column v mn corresponding components be{ h_{i1},h_{i2},...,h_{ik}} then,

Figure3: Matrix factorization for document clustering

$$v_i \approx W_1.h_{i1}+W_2.\ h_{i2}+\ldots..+W_k.h_k \quad (5)$$

NMF uses an iterative procedure to change the initial values of W_{mk} and H_{kn} so that the product approaches V_{mn}. Whenever the approximation error converges or if the number of iterations statedareattained, transaction terminates. The decomposition of the NMF is not unique; the W and H matrices depend on the NMF algorithm used and the error measure used to check convergence. Some of the types of NMF algorithms are, Lee and Seung's multiplicative update algorithm, Hoyer's sparse encoding, Pauca's lowest-square gradient descent, and Pattero's lowest-square algorithm. They differ in the cost function of measuring the divergence between V and WH or by regularizing the matrices W and/or H.

3.4.2 K-Means Algorithm

- Standard k-means:

K-means is the mostly used clustering algorithmfor clustering. It is one of the most efficient algorithm for the generation of clusters. K-means algorithm is an algorithm that tries to find the center of cluster and minimizes the overall inter-cluster variance, or, the squared error function.

$$V= \sum_{i=1}^{k} \sum_{x_i \in S_i} \left(x_j - \mu_i\right)^2 \quad (6)$$

Here,

Si, i=1,2,…,k are the clusters and µi is the centroid for all the pointsxj∈ Si.

K-means takes the number of clusters as an input from the user and then performs clustering. The complexity of this algorithm is O(t.d.k.m) where,

d = number of documents,
t = the number of terms,
k = the clusters required and
m =maximum number of iteration

The basic concept of this algorithm is to categorize points into closest of the pre-defined number of clusters until a certain number of iterations has been achieved.The following pseudo code represents the k-means algorithm.

Input

- k (the number of clusters)
- D (a set of lift ratios)

Output

- a set of k clusters

Method
Choose k objects as the original cluster centers from D arbitrarily;
Repeat:
Step1:To cluster, every object has to be reassigned by calculating the mean of the cluster objects.
Step2: Later, update the mean cluster, i.e. compute the average value for each group until no changes are made

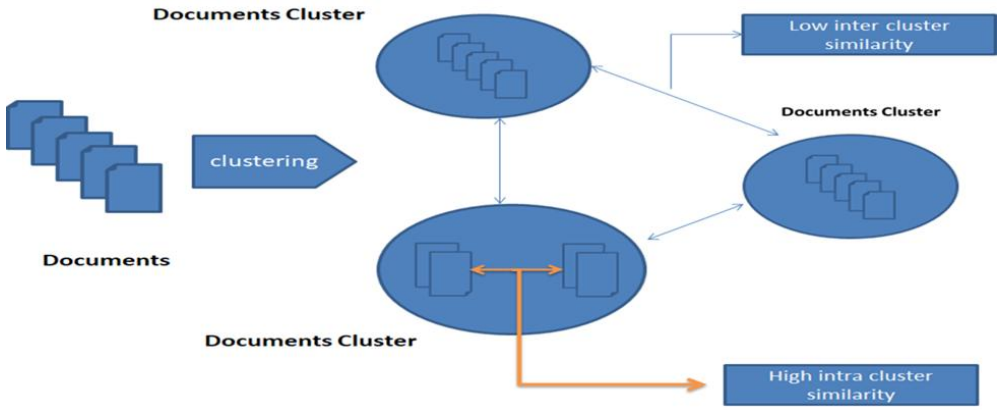


Figure4: Different text document for Clustering

The above figure shows us how the similarity of the contents of the clusters results in formation of different types of clusters. Here, low inter cluster similarity denotes that although the terms inside the clusters are having greater similarity (as seen in figure), the similarity between two clusters are less due to which the distance between their centroids increases. Similarly, the high intra cluster similarity denotes that the files or contents of that particular cluster are having significant similarity due to which they are located inside same clusters.

3.4.3 KNMF

The clustering of documents is carried out in KNMF algorithm based on the resemblance among theextracted features and the individual documents. Assume feature extracted vectors as $F=\{f_1,f_2,f_3....f_k\}$ which are calculated by NMF. Consider term-document matrix documents as $V = \{d_1,d_2,d_3....d_n\}$. When the angle between the d_i and f_x is minimum then, the document d_i is supposed to belong to cluster f_x.

Procedure

1. Build the document term matrix V using the TFIDF values from the records of a given input folder

2. The length of columns of V is standardized by using the Euclidean distance

3. NMF is applied on V and calculate the values of W and H by using the below equation

$$V_{ab} \approx W_{ab} \times H_{bc} \quad (7)$$

4. To calculate the distance between the documents d_i and extracted vectors of W, K-means algorithm is used. When the angle between d_i and w_x is minimum, allocate d_i to w_x. This is correspondent to a k-means algorithm by a particular turn.k-means is one of the most important unsupervised algorithms for the process of clustering. Clustering is the mechanism of splitting the points into classes based on the resemblance. If the value of K is given as input, then the process of the K-means as follows

- Divide the objects or data into K subsets in which data is not null.
- Recognize the mean point of the clusters for the current split.
- Allocate each point to a particular cluster.
- Calculate the distances from each point and then allocate the points to the cluster based on the minimum distance from the centre.
- After rearranging the points calculate the new mean based on the newly assigned points.

5. To run the K-means parallel Hadoop is executed in local and pseudo mode.

The procedures of NMF, MM, and KNMF have explained above. The proposed method follows the below procedure.

- Initially, the dataset NEWSGROUP20 is downloaded for the implementation.

- By using pre-processing techniques the term-document matrix of a particular document is obtained in a dataset
- To obtain features of a particular document NMF method is used
- For cluster formation of an individual document based on similarity, k-means algorithm is used.
- Finally, from formed clusters identify top terms in that.

4 Procedure for Indexing Text Document using Lucene

Step 1: Initially, the text documents are loaded and the searchapplicationcontainsdetails of the document.
Step 2: With the help of Lucene, the generated document is been analysed
Step 3: After analysing the document, Lucene keeps the document in an indexed format.
Step 4: Search application provides query options within the process and Lucene provides search index among the existing documents.
Step 5: Finally, the results from the search will be generated by the application.
First three steps are part of Indexing and the second two steps are part of searching. Functionalities like delete and update are provided by the search application.

5. RESULTS

Results were analysed on Newsgroup20 dataset. It consists of 20,000 documents separated into 20 groups, used to cluster and classify the text documents. Each article in a newsgroup is stored in separate files which are pre-processed, extractedand clustered.The following table3 has the Newsgroup20 dataset articles

Table3: Different articles containing text documents in Newsgroup20 dataset

comp.graphics comp.os.ms-windows.misc comp.sys.ibm.pc.hardware comp.sys.mac.hardware comp.windows.x		rec.autos rec.motorcycles rec.sport.baseball rec.sport.hockey	sci.crypt sci.electronics sci.med sci.space
misc.forsale	talk.politics.misc talk.politics.guns talk.politics.mideast	talk.religion.misc alt.atheism soc.religion.christian	

Figure5 describes thesample input text document from newsgroup20 dataset.

Figure6 describes the implementation of the NMF algorithm by initializing TF_IDF values. Here the values are read by rows and columns. After the initialization of TF_IDF values, it follows K-means clustering algorithm along with NMF, therefore the number of clusters need to be declared. Figure7 describesOutput for Processing Non-Negative Matrix Factorization of 4 Clusters by presenting top 10

November-December 2019
ISSN: 0193-4120 Page No. 1107 - 1130

terms for cluster1 and cluster2.Figure8 describesOutput for Processing Non-Negative Matrix Factorization of 4 Clusters by presenting top 10 terms for cluster3 and cluster4.

Figure9 describes the GUI for automatic indexing and searching text documents using Apache Lucene, text from selected documents by Click File menu and open File.Figure10shows theselection of text documents from the data folder through open dialogfor automatic indexing and searching text documents using an interface.Figure 11 shows theselection of Single or multiple documents for automatic indexing and searching text documents using a GUI interface.Figure 12Loads the input text documents from the data folder.

Figure 13Selection of single or multiple words which needs to be searched from the indexed data.Figure 14 shows the final output for Indexing and searchingFigure15 shows the indexed text documents automatically stored inside the index folder which we created to stored.

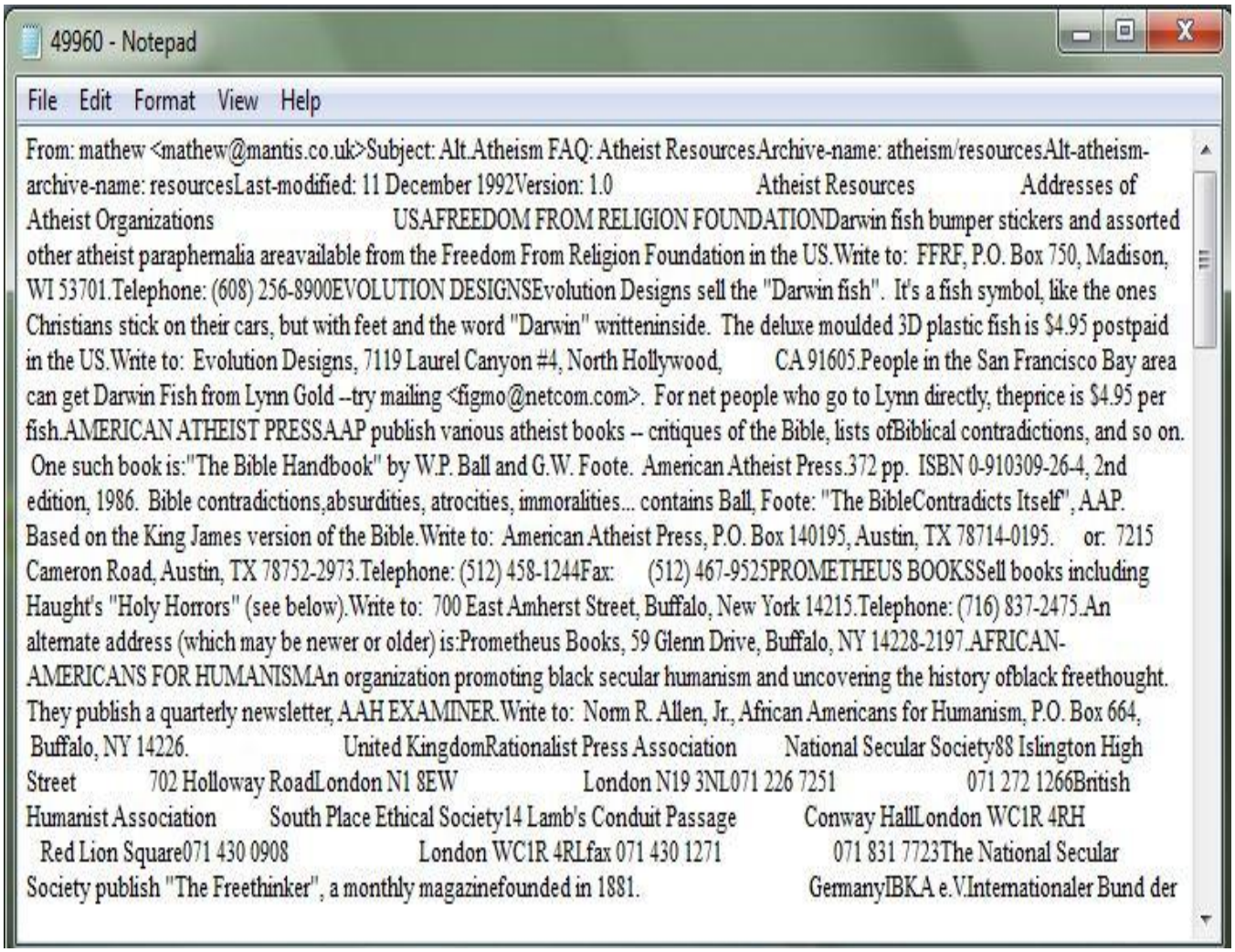


Figure5: Sample text document 49960.txt for clustering

November-December 2019
ISSN: 0193-4120 Page No. 1107 - 1130

```
NON NEGATIVE MATRIX FACTORIZATION:

 Reading C:\Users\bigdata\Downloads\Compressed\NMFProject-20190314T051417Z-001\NMFProject\newsgroup datasets
 ..................................
 80011 rows retrieved

 Processing C:\Users\bigdata\Downloads\Compressed\NMFProject-20190314T051417Z-001\NMFProject\newsgroup datas
 ..................................
C:\Users\bigdata\Downloads\Compressed\NMFProject-20190314T051417Z-001\NMFProject\newsgroup datasets\newsgrou

 Reading C:\Users\bigdata\Downloads\Compressed\NMFProject-20190314T051417Z-001\NMFProject\newsgroup datasets
 ..................................
 5942 rows retrieved

 Processing C:\Users\bigdata\Downloads\Compressed\NMFProject-20190314T051417Z-001\NMFProject\newsgroup datas
 ..................................
C:\Users\bigdata\Downloads\Compressed\NMFProject-20190314T051417Z-001\NMFProject\newsgroup datasets\newsgrou

 Initializing TF IDF Normalization:
 ..................................
Please wait...
 Processing and storing the normalized TF-IDF values
 Creating and storing values inC:\Users\bigdata\Downloads\Compressed\NMFProject-20190314T051417Z-001\NMFPro
 Complete Processing Data Matrix.

 Initializing Non Negative Factorization.
 ..................................
 Enter the number of cluster: 4
```

Figure6: Output for NMF algorithm, reading newsgroup dataset by rows and columns and initializing TF-IDF with declaring clusters as 4

```
 Enter the number of cluster: 4

 Processing Non Negative Factorization
 ..................................
Processing, please wait...

 Top 10 Terms for Cluster 1
 ..................................
 Rank 1: recommend
 Rank 2: rpiedu
 Rank 3: info
 Rank 4: interchang
 Rank 5: himself
 Rank 6: sell
 Rank 7: agian
 Rank 8: depend
 Rank 9: that
 Rank 10: ps

 Top 10 Terms for Cluster 2
 ..................................
 Rank 1: rpiedu
 Rank 2: write
 Rank 3: zamenhofcsriceedu
 Rank 4: current
 Rank 5: check
 Rank 6: agian
 Rank 7: sec
 Rank 8: dan
 Rank 9: uxacsouiucedu
 Rank 10: dem
```

Figure7: Output for Processing Non negative Matrix Factorization for 4 Clusters

November-December 2019
ISSN: 0193-4120 Page No. 1107 - 1130

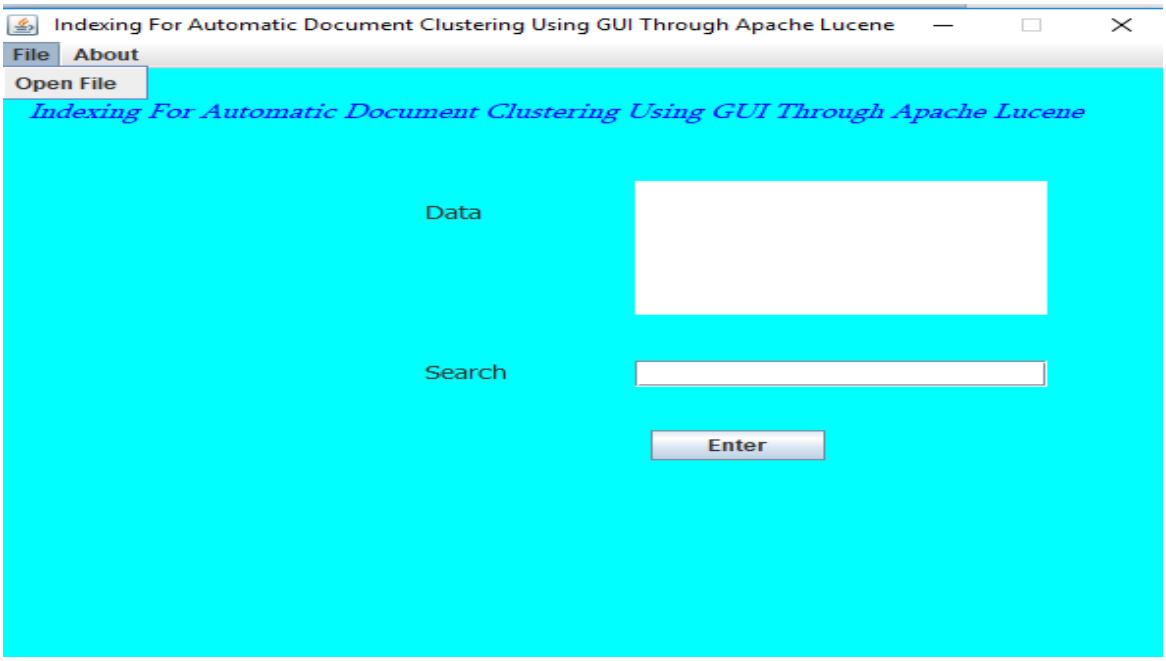
```
Rank 5: check
Rank 6: agian
Rank 7: sec
Rank 8: dan
Rank 9: uxacsouiucedu
Rank 10: dem

Top 10 Terms for Cluster 3
..............................
Rank 1: recommend
Rank 2: rpiedu
Rank 3: articl
Rank 4: stand
Rank 5: interchang
Rank 6: see
Rank 7: nice
Rank 8: happen
Rank 9: depend
Rank 10: freehand

Top 10 Terms for Cluster 4
..............................
Rank 1: pleas
Rank 2: rpiedu
Rank 3: gif
Rank 4: somebodi
Rank 5: sound
Rank 6: electron
Rank 7: farsight
Rank 8: agian
Rank 9: burst
Rank 10: 68070

Non-negative Matrix Fatorization Completed
```

Figure8:Output for Processing Nonnegative Matrix Factorization for 4 Clusters



November-December 2019
ISSN: 0193-4120 Page No. 1107 - 1130

Figure9: Click File menu and open File.

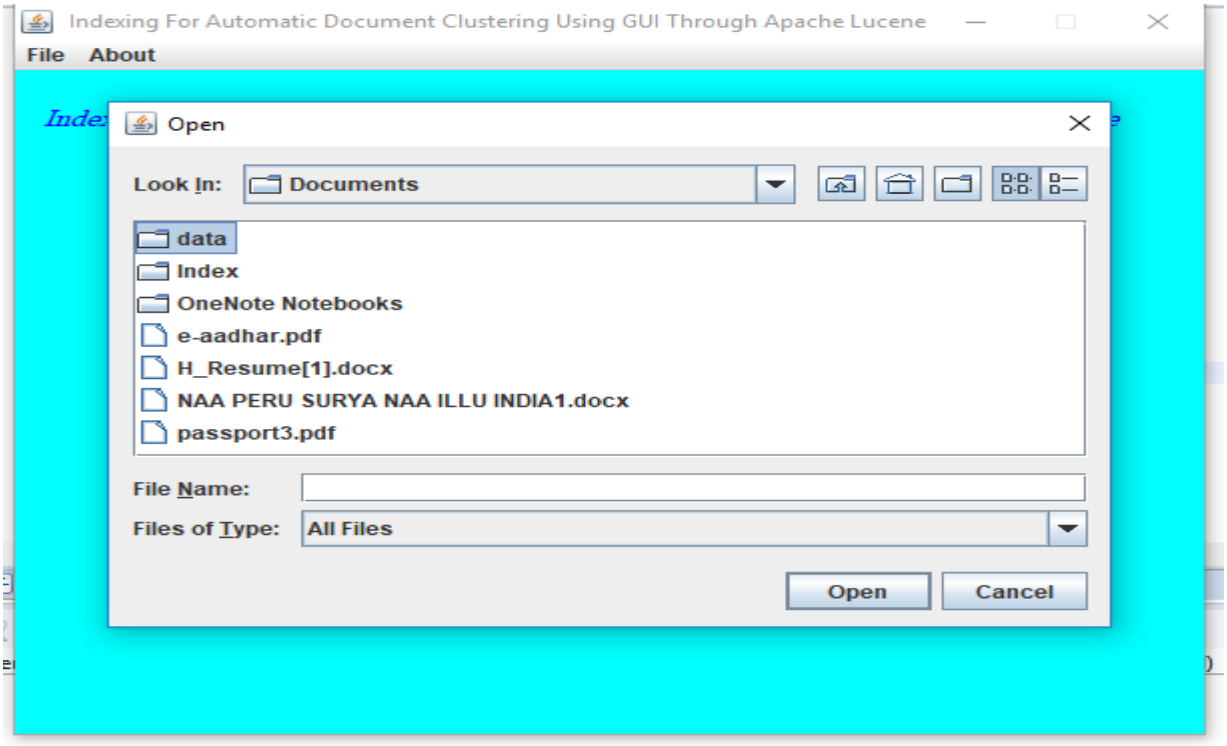
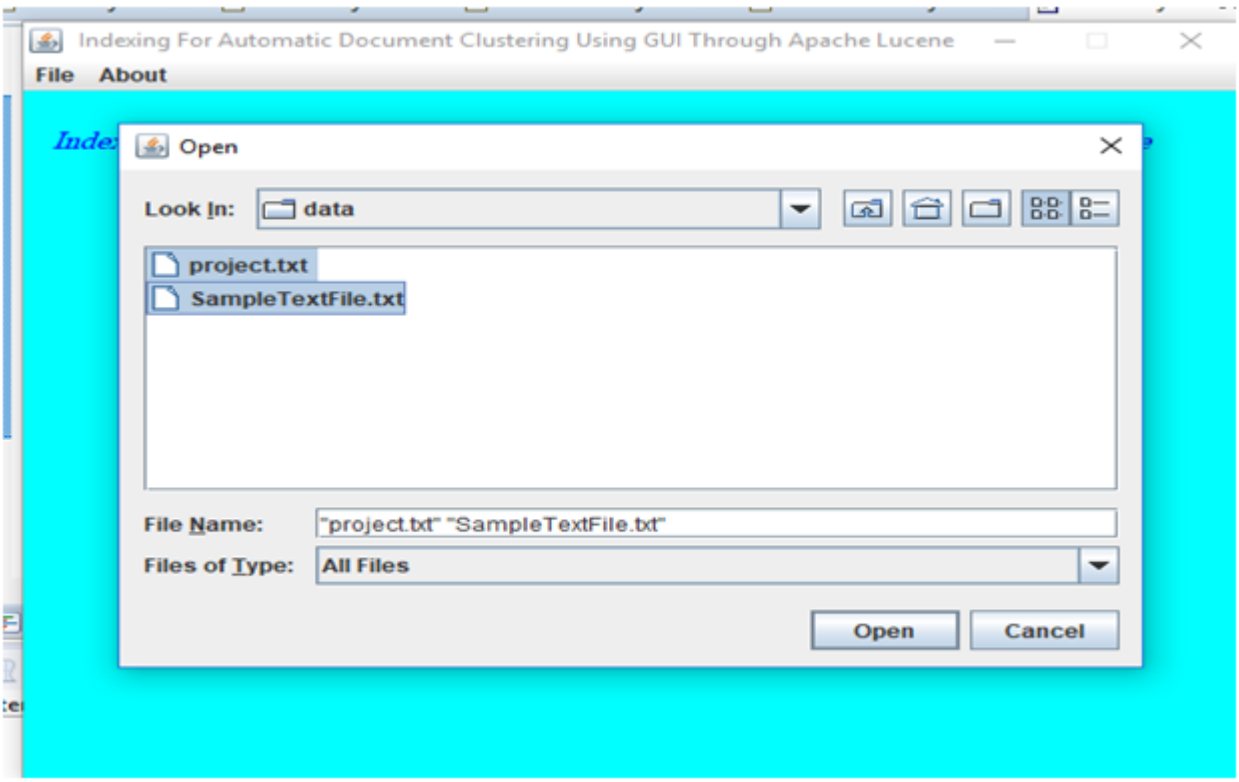


Figure10: Choose the data folder from open dialog.



November-December 2019
ISSN: 0193-4120 Page No. 1107 - 1130

Figure11: Choose Single or multiple documents.

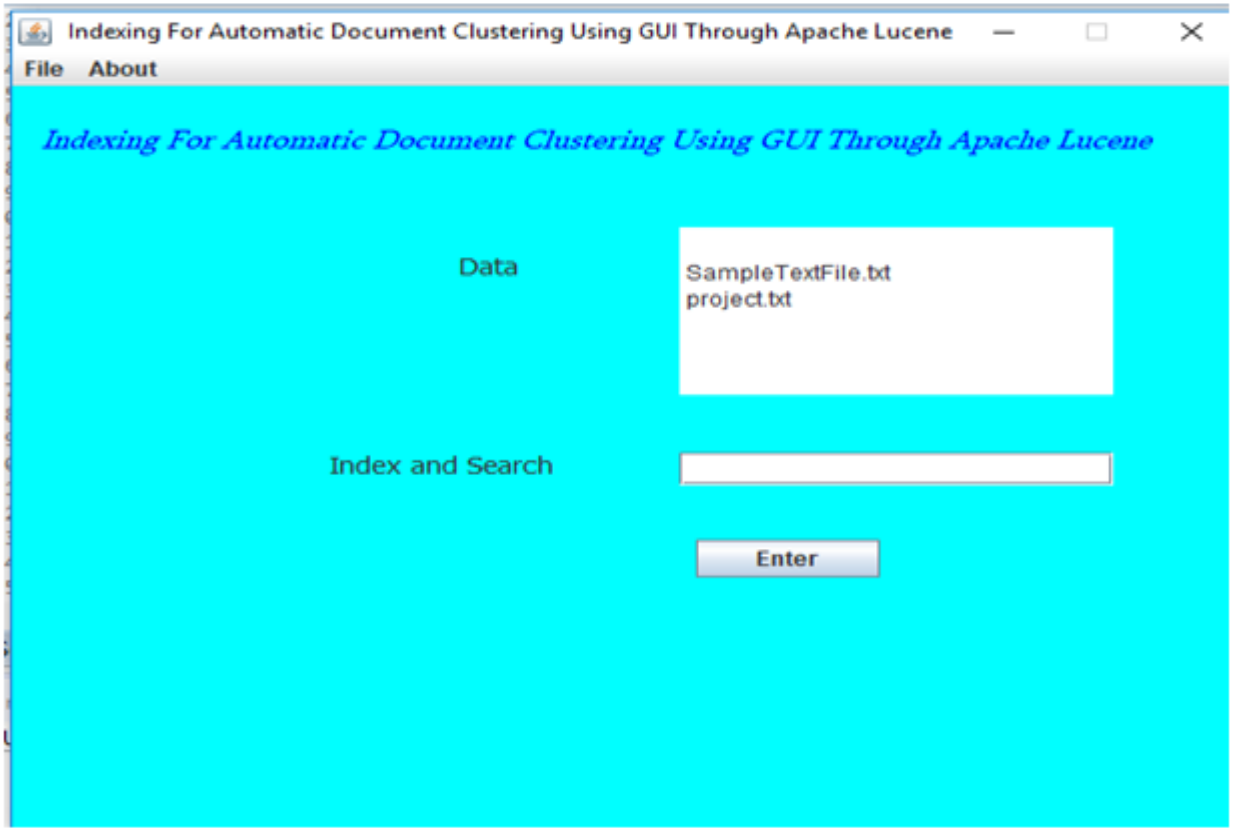
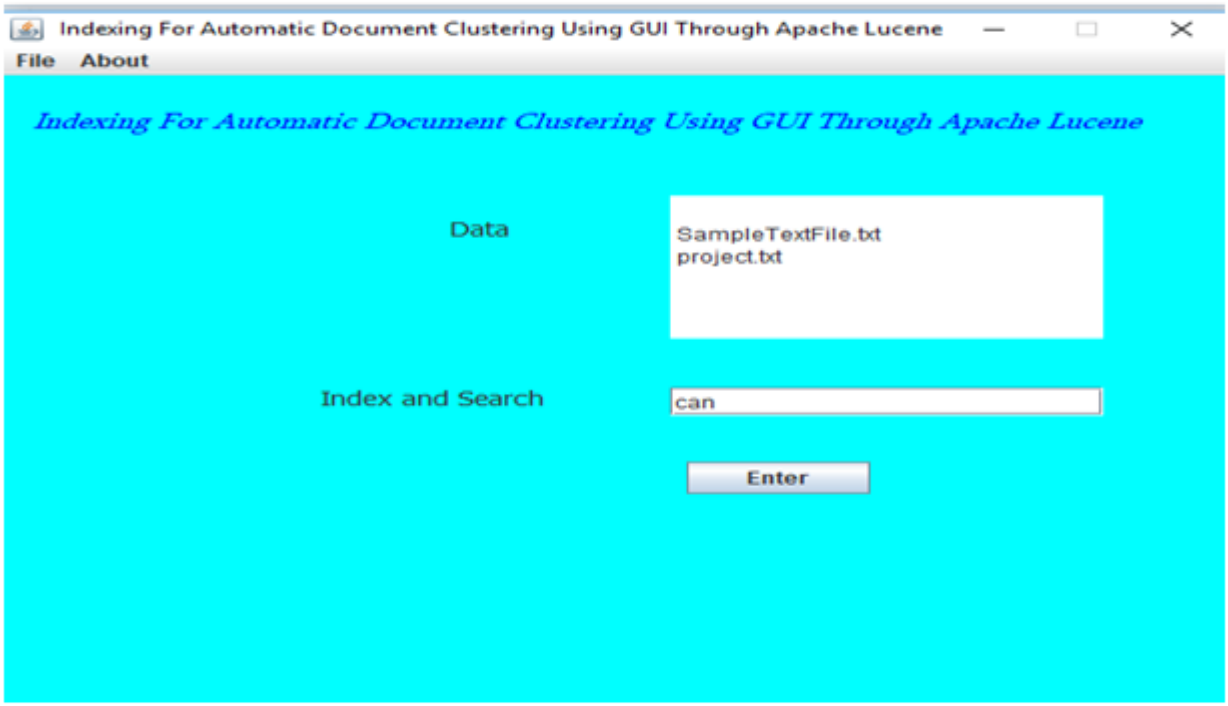


Figure12: Input has been taken from the data folder.



November-December 2019
ISSN: 0193-4120 Page No. 1107 - 1130

Figure13: Select single or multiple words which should be Index and search.

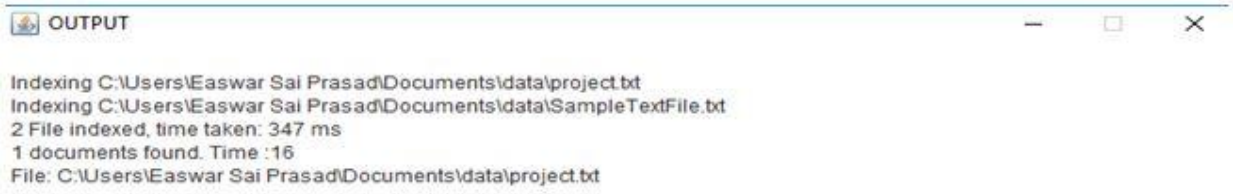


Figure14: Final output for Indexing and searching

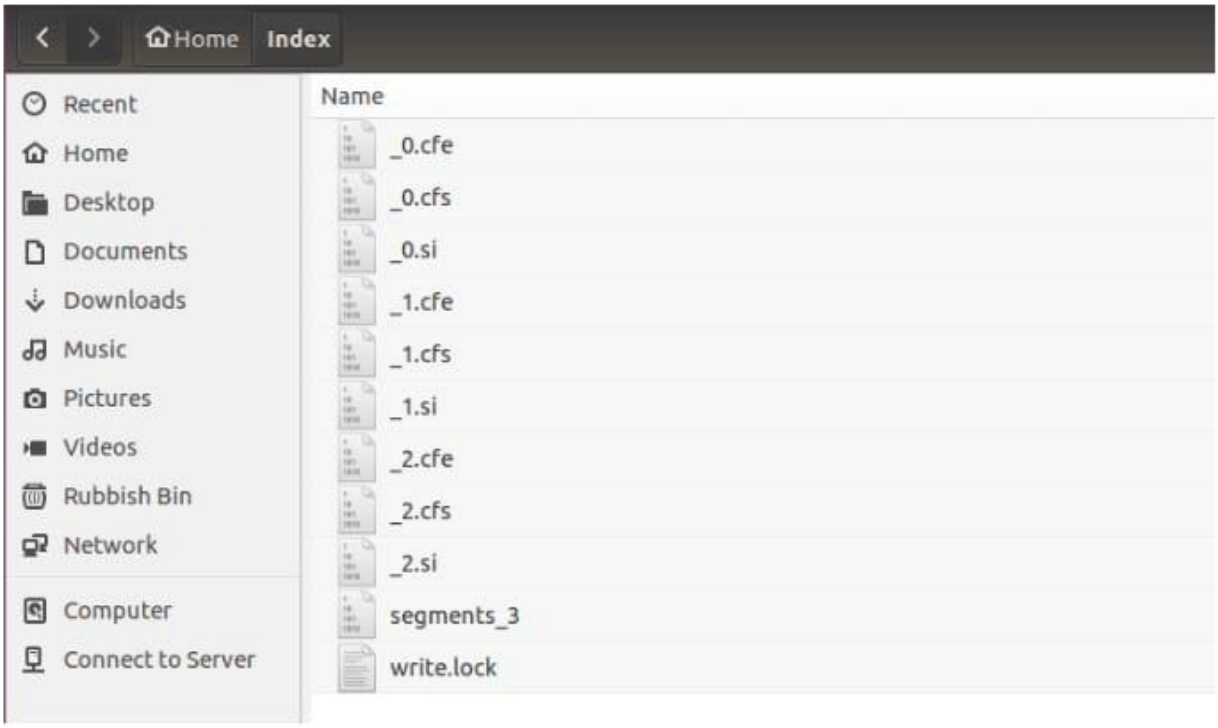


Figure15: Indexed text documents

6. Conclusion

Thepaper demonstrates an easy way to solve clustering text documents and indexing them automatically with the help of Apache software's like HADOOP and LUCENE. For clustering, NMF is upgraded to K-means algorithmi.e, KNMF.Clustering documents were performed by Non Negative Matrix factorization including K-means to it. Input to this KNMF is given by the pre-processed data of text document i.e, TFIDF. The input is considered in matrix format and processed. Declarations of clusters are defined by the user. Each cluster contains the top ten topics used in the document. This

has led to separate documents automatically into sub-groups, to reduce manual filtering of documents. The performance of the models with proposed technique was conducted and found to be 80% for 4 clustered data. On the other hand, Lucene performs indexing and searching for multiple documents act as fastest search engine. A GUI act as an interface between the user and system has been developed to reduce unnecessary information and achieve exact data what the user is looking for. Thus, the paper helps the user to get output quickly within limited time when compared to other software environments.

FUNDING ACKNOWLEDGMENT

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. Mrs.B. MeenaPreethi, Dr. P. Radha, *"A Survey Paper on Text Mining - Techniques, Applications And Issues",* IOSR Journal of Computer Engineering(IOSR-JCE), e-ISSN: 2278-0661, p-ISSN:2278-8727, pp 46-51.
2. Meiping Song, Qiaoli Ma, Jubai An & Chein Chang, *"An Improved NMF Algorithm Based on Spatial and Abundance Constraints",* 2016 progress in electromagnetic research symposium(PIERS), Shanghai, China, 8-11 August, pp 4532-4537.
3. Dipesh Shrestha,*"Text Mining with Lucene and Hadoop:Document Clustering with feature extraction ",* thesis - WakhokUniversity, 2009.
4. Yu-XiongWang,Yu-Jin Zhang, *"Nonnegative Matrix Factorization: A comprehensive review*", IEEE Transactions on knowledge and data engineering, Vol.25, No.6, June 2013,pp 1336-1353
5. Jia Qiao& Yong Zhang,*"Study of K-means Method Based on Data-Mining",*2015 Chinese Automation Congress(CAC). DoI: 10.1109/CAC. 2015.7382468
6. E. Laxmi Lydia and D. Ramya,*"Text Mining With Lucene And Hadoop: Document Clustering With Updated Rules Of NMF Non Negative Matrix Factorization",* International Journal of Pure and Applied Mathematics, Volume 118, No.7 2018, pp 191-198.
7. Serhat Selcuk Bucak and Bilge Gunsel,*"Incremental Clustering via Nonnegative Matrix Factorization",*2008 19th International Conference on Pattern Recognition. DoI: 10.1109/icpr.2008.4761104.
8. Abhay Kumar, Ramnish Sinha, Daya Shankar Verma and Vandana Bhattacherjee Satendra Singh, "*Modeling using K-Means Clustering Algorithm",*2012 1st International Conference on recent Advances in Information Technology.
9. Chengbin Peng, Ka-Chun Wong, Alyn Rockwood, Xiangliang Zhang and Jinling Jiang, David Keyes,*"Multiplication Algorithms for Constrained Nonnegative Matrix Factorization",* IEEE computer society, 2012 IEEE 12th International Conference on data mining. DoI-10.1109/ICDM.2012.106.
10. Jie Tang, Xinyu Geng and Bo Peng*"New methods of Data Clustering and Classification based on NMF"* , 2011 International conference on business computing and Global informatization. DoI:10.1109/bcgin.2011.114.

11. Dr.E.Laxmi Lydia, P.Krishna Kumar, K.Shankar, S.K.Lakshmanaprabu, R.M.Vidhyavati, AndinoMaseleno, *"Charismatic Document Clustering through novel K-means Nonnegative Matrix Factorization(KNMF) Algorithm using Key Phrase Extraction",* International Journal of Parallel Programming, https://doi.org/10.1007/s10766-018-0591-9, Springer, 2018.
12. Dr.E.Laxmi Lydia, P.Govindaswamy, SK. Lakshmanaprabu, D. Ramya, "*Document Clustering based on Text Mining K-means algorithm using Euclidean Distance Similarity*", Journal of Advanced research in Dynamical & Control Systems, Vol.10, 02-Special Issue, 2018.
13. Dr.E.Laxmi Lydia, Dr.K.Vijaya Kumar, P.Amaranatha Reddy, D. Ramya, "*Text Mining with Hadoop: Document Clustering with TF_IDF and Measuring Distance using Euclidean*", Jour of Adv. Research in Dynamical & Control Systems, vol.10, 14-Special Issue, 2018.
14. Lee, D &Seung, H (2001), "*Algorithms for nonnegative matrix factorization*", Proceedings of the 2000 Conference:556562, The MIT Press. In T.G. Dietterich and V. Tresp, editors, Advances in Neural Information Processing Systems, volume 13.
15. MohitBhansali, Praveen Kumar, "*Searching and Analyzing qualitative data on personal computer*", IOSR Journal of Computer Engineering, e-ISSN: 2278-0661,p-ISSN:2278-8727 Volume 10, Issue2, April 2013, PP 41-45.
16. Jimmy Lin, DmitriyRyaboy, Kevin Wells, "*Full-text indexing for optimizing selection operations in large-scale data analytics*", ACM, San Jose, California, USA, 978-1-4503-0700-0/11/06, June, 2011.
17. Yang CF, Ye M and Zhao J, " *Document clustering based on Nonnegative Sparse Matrix Factorization*", International Conference on Advances in Natural Computation, Page No: 557-563,2005.
18. Chris Ding, Xiaofeng He, D. Simon, "*On the Equivalence of Non- Negative Matrix Factorization (NMF) and Spectral Clustering* ", Proceedings of SIAM International Conference on Data Mining, Page No:267-273.
19. HirokazuKameoka, Takuya Higuchi, Mikihiro Tanaka, Li Li, "*Nonnegative Matrix Factorization with Basis Clustering using Cepstral Distance regularization*", IEEE/ACM transactions on Audio, Speech, and Language Processing, Vol 26, issue:6, June 2018. DOI:10.1109/TASLP.2018.2795746
20. Xlumei Wang, Tianzhen Zhang, XinboGao, "*Multiview Clustering Based on Nonnegative Matrix Factorization and Pairwise Measurements*", IEEE Transactions on Cybernetics, June 2018, DOI: 10.1109/TCYB.2018.2842052