An Implementation of Text Mining Decision Feedback Model Using Hadoop MapReduce

Swagat Khatai, Siddharth Swarup Rautaray, Swetaleena Sahoo, and Manjusha Pandey

Abstract A very large amount of unstructured text data is generated everyday on the Internet as well as in real life. Text mining has dramatically lifted the commercial value of these data by pulling out the unknown comprehensive potential patterns from these data. Text mining uses the algorithms of data mining, statistics, machine learning, and natural language processing for hidden knowledge discovery from the unstructured text data. This paper hosts the extensive research done on text mining in recent years. Then, the overall process of text mining is discussed with some high-end applications. The entire process is classified into different modules which are test parsing, text filtering, transformation, clustering, and predictive analytics. A more efficient and more sophisticated text mining model is also proposed with a decision feedback perception in which it is a way advanced than the conventional models providing a better accuracy and attending broader objectives. The text filtering module is discussed in detail with the implementation of word stemming algorithms like Lovins stemmer and Porter stemmer using MapReduce. The implementation set up has been done on a single node Hadoop cluster operating in pseudo-distributed mode. An enhanced implementation technique has been also proposed which is Porter stemmer with partitioner (PSP). Then, a comparative analysis using MapReduce has been done considering above three algorithms where the PSP provides a better stemming performance than Lovins stemmer and Porter stemmer. Experimental result shows that PSP provides 20–25% more stemming capacity than Lovins stemmer and 3–15% more stemming capacity then Porter stemmer algorithm.

S. Khatai (✉) · S. S. Rautaray · S. Sahoo · M. Pandey
KIIT Deemed to be Univerisity, Bhuneshwar, Odisha, India
e-mail: k.swagat1391@gmail.com

S. S. Rautaray
e-mail: siddharthfcs@kiit.ac.in

S. Sahoo
e-mail: swetaleenafet@kiit.ac.in

M. Pandey
e-mail: manjushafcs@kiit.ac.in

S. Rautaray et al. (eds.), *Trends of Data Science and Applications*,
Studies in Computational Intelligence 954,
https://doi.org/10.1007/978-981-33-6815-6_14

Keywords Big data · Hadoop · MapReduce · Word stemming · PSP · Decision feedback

1 Introduction

In today's world, the amount of unstructured data is growing in an enormous way that the existing relational systems are incompetent in handling them. The form of data can be audio-video clips, textual data, software program logs, flight records, etc. The information hidden inside those data leads to a complete new world of opportunity and insight. This is the reason for why every organization and individual is demanding to explore these huge amount of data, which constructs the foundation of text mining. It is also called as a practice of textual form of data to discover the key conceptions, themes, hidden trends, and relationships without prior knowledge of exact terms that has been used by author to express the concept [1]. As part of text mining algorithms of data mining, text analytics, machine learning, natural language processing, and statistics are used to extract high quality, useful information from unstructured formats. Test mining is also popular as "text analytics" is a means by which unstructured data is processed for machine use. For example, if a Twitter comment "I don't find the app useful: it's really slow and constantly crashing." is taken into consideration then text mining of the contextual information is really important to help us understand why the tone might be negative and what may be the cause of such customer disappointment as shown in Fig. 1. These analyses may lead to the answer of questions like "Is the person replying to another negative tweet? or is this the original composition? or what is the application name? or is this the only problem with the app or there are other problems too?, etc.

1.1 Conventional Process Flow of Text Mining

Textual data are in the form of unstructured data are normally available in readable document formats. These formats can be user comments, e-mails, corporate reports,

Fig. 1 Text mining

web pages, news articles, etc. According to conventional text mining process, the documents are first derived into a quantitative representation. Once the textual data is transformed into a set of numbers which precisely capture the hidden pattern in it, then any data mining algorithm or statistical forecasting model is applied on the numbers for generating insights or for discovering noble facts [2, 3].

A typical text mining process generally have the following sub-tasks to complete the process.

Data Collection Collection of textual data is the first step in any text mining research [3].

Text Parsing and Transformation The next step is to parse the words from the documents. Therefore, sentences, parts of speech, and stemming words [3] are identified from the document. Document variables associates with author, category, gender, etc., are also extracted with the parsed words.

Text Filtering After the parsing of words, there may be some irrelevant words which are not required in the analysis, and those words are removed from the document. This is done manually by browsing through the terms or words. This is the most time-consuming and subjective tasks in all of the text mining steps. A fair amount of subject knowledge and domain knowledge is required to perform this task. In case of document filtering [3], the selected keywords are searched in all the selected documents. If any document does not contain any of the keywords, then it is removed from the list of analysis.

Text Transformation In this step, the document is presented in a numerical form of matrix [3]. This matrix generally contains the occurrences of the words is also called as term frequency. Numerical presentation of the document is mandatorily required to perform any kind of analytics on the document. Therefore, this step converts the unstructured text to a workable analytical document.

Text Mining In this step, hidden patterns and knowledge are extracted using mining algorithms such as classification, clustering, association analysis, and regression analysis. As shown in Fig. 2, text mining is an iterative process where the process of filtering to mining is repeated based on the feedback received from this step [4].

1.2 Applications of Text Mining

Text mining process is being used to provide answers to industrial queries and to optimize daily operations efficiently. It is also used to develop business strategic decisions in finance, automobile, marketing, health care, etc. Hidden patterns, trends, and perceptions are discovered from a huge volume of unstructured data using techniques like data analytics, categorization, and sentiment analysis. In this research, we have discussed below applications of text mining.

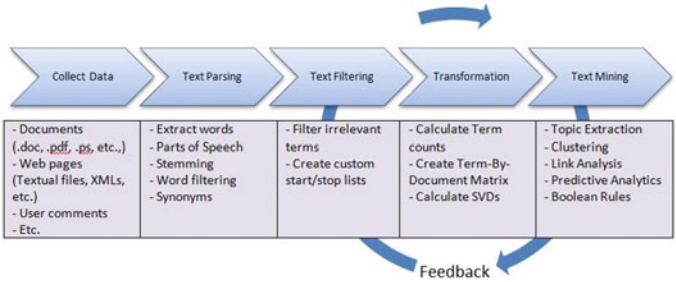


Fig. 2 Conventional text mining process

Risk Management Inadequate risk estimation is accounted for biggest reason of failures in any industry. In these cases, text mining is used to estimate the proper risk in business and also to identify the most adequate way to mitigate the risk [3]. Therefore, the application of text mining software has drastically increased the capacity of risk mitigation in industries.

Knowledge Management Managing huge volume of data containing the historical information creates many problems like huge storage space, latency in finding specific information, etc. The healthcare industries are a classic example for the above problems where the information of historical patients' data can be potentially used for medical analysis and product development [3]. Therefore, text mining is used to filter the useful informations by discarding the irrelevant ones. Then, many analytic algorithms are run on the filtered data to find and store the extracted unknown facts only, which reduces the storage issue, latency issue, etc.

Cybercrime Prevention Random availability of information over Internet can bear the brunt of cybercrimes. Text mining is used to trace the cybercrime activities and also helps to identify the source of intruders [3]. Therefore, text mining is used by law enforcement and intelligence agencies.

Customer Care Service Customer care services are better operated using text mining and natural language processing. Text analytics software improves customer experiences. These analytics use many valuable information sources such as survey and customer call notes which help effectiveness and speedy resolution of customer problems [3]. Test mining is also used for automated faster responses to customer queries.

Contextual Advertising Digital advertising has got a new height of safety and user's privacy by applying text mining as core engine of contextual retargeting [3]. It also provides better accuracy in contextual advertising.

Business Intelligence Text mining is uses to support faster decision making by taking consideration of valuable enterprise data [3]. It helps to find future insights for improving the business by monitoring huge number of data sources.

Social Media Social media is a potential source of huge amount of unstructured data inside which a lot of hidden patterns related to business, sentiment [5], and intelligence are there. Many organizations predict the future customer needs using text analytics. This information help organizations to extract the customer opinions, to understand their emotions, and also to predict their requirements. Text mining has made revolutionary modifications in social media.

2 Literature Survey

As text mining is our focus of research, therefore, some recent research artifacts are studied. All the related studies and analysis points that application of big data technologies like Hadoop MapReduce, k-means, particle swarm optimization (PSO), and cloud computing provides better result, reduced execution time and better solution for big data problems. Large data sets can be analyzed using Hadoop cluster and parallelization of clustering algorithms and using parallel k-means clustering provides a drastic reduction in execution time [1]. Document clustering, parallel k-means, and distributed computing [6] are the techniques that have been used with Hadoop MapReduce in the study. After selecting centroids randomly, every document is fed to one mapper. The mapper calculates the new centroids based on the Euclidean distance. The result of all mappers is sent to a reducer to calculate a resulting centroid which then compared with the assumed centroid [7]. If there is a difference in centroid value, then the process is iterated, otherwise, the centroid is considered as final output as shown in Figs. 3 and 4.

To settle the number of cluster and initial centroid, the parallel k-mean algorithm is modified which can be optimized using fuzzy logic, gravitational intelligence, and swarm optimization. Big data has its own challenge in terms of storing the data and retrieving it fast. Manual grouping of files is very complex when there is a huge amount of document. A new working k-means non-negative matrix factorization (KNMF) with modified guideline of non-negative matrix factorization [8, 9] is used for document clustering. Comparison study of iterated Lovins algorithm, Lovins algorithm, and Porter algorithm of text mining shows that maximum words are stemmed in iterated Lovins algorithm. Therefore, the characteristics of k-means non-negative matrix factorization help in clustering the documents with parallel implementation of MapReduce on large sized documents. This results in quick and easy clustering as well as less time consumption.

In order to shrink the computational time, HDFS, MapReduce, and clustering algorithms are used by distributing the clustering jobs on multiple nodes which means multiple clustering tasks run parallel on different nodes. A comparative review of components of Hadoop and MapReduce has been studied to compare result with the traditional partition-based algorithms with their implementation in MapReduce paradigm to achieve various clustering objectives on different size of data sets [7]. Introduction of combiner programs between the map and reduce function helps in

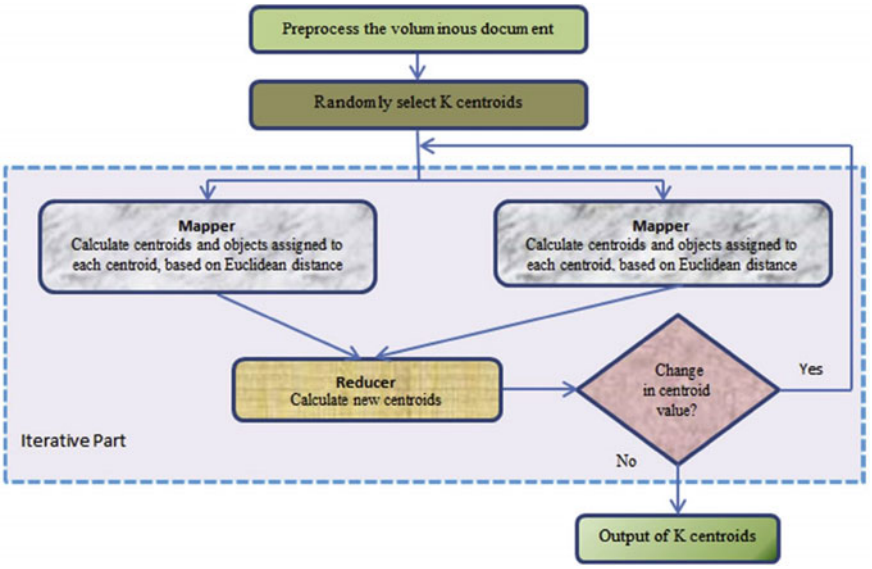


Fig. 3 Stages of document clustering using parallel k-means

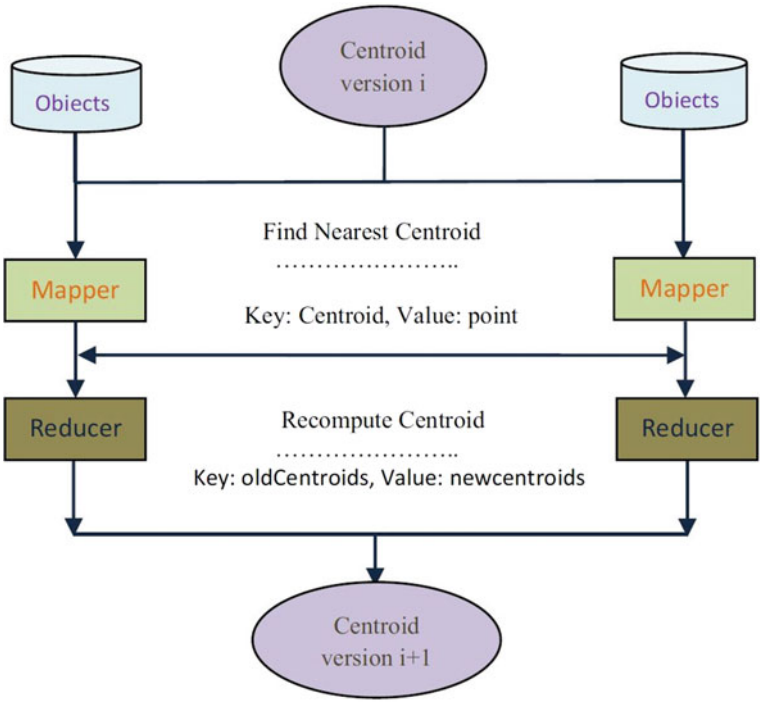


Fig. 4 Parallel k-means algorithm with MapReduce

reduction of volume of data to be written by mapper and volume of data to be read by reducers that decrease the overall operation time. The time reduction is highly realizable when the number of document is huge rather than smaller data sets [1]. The model for implementation of parallel k-means clustering in MapReduce without a combiner is shown in Fig. 4.

With above working methods, the advantage of the ability of global search in particle swarm optimization (PSO) is used for optimal generation of centroids. The power of parallel processing with global search supports data intensive distributed application with improved accuracy in generating compact clusters [4]. Some more literatures are studied in context of text mining and a comparative study is presented in Table 1 in terms of their objective, findings, and methods used. From the study, it is clear that MapReduce is the most popular technology to handle text mining problems. Therefore, in our context of research, we have implemented the proposed text mining model using Hadoop MapReduce with partitioner.

3 Proposed Decision Feedback-Based Text Mining Model

Textual content is typically available in comprehensive document format. These formats can be e-mails, text file lettering, user feedback, sentimental comments, corporate reports, sentimental comments, news reporting, Web pages, etc. The proposed text mining model tries to first instigate a quantitative representation of document and then transfer the document into a set of numbers where the numbers adequately capture the patterns of textual data. Any traditional statistic model, forecasting model, and analytical algorithm can be used on these numbers for generating insights or to produce a predictive modeling. Statistical-based systems count the word frequency of each word and calculate their statistical proximity toward related conceptual indexes. These systems may produce inappropriate concepts and miss the required words which in turn reduces the prediction model accuracy. Iterative text mining decision feedback model is the advanced form of a text mining where the process is repeated till the result is acceptable without getting completely out of process. In this model, the feedback block is the controller of number of iterations. Feature selection, data analytics, and evaluation phase constitutes the feedback block. Also, this process minimizes the interference of irrelevant words to increase the model accuracy. The iterative text mining model which we have proposed has the design as shown in Fig. 5. The steps involved in this model are:

Data Collection Collecting an unstructured data set for analysis is always the first step of any text mining process.

Text Parsing and Transformation In this step, the data set is cleaned and a dictionary of words is created from the document using NLP. This includes identification of sentence, word, parts of speech, and stemming words [1]. The extraction of each word from document is associated with a variable for further reference in the process.

Table 1 Comparative study of recent works in text mining

Title	Year	Objective	Technology used	Finding
An analysis of MapReduce efficiency in document clustering using parallel K-means algorithm [1]	2018	To design and experiment a parallel k-means algorithm using MapReduce programming model and compared the result with sequential k-means for clustering varying size of document data set. The result demonstrates that the proposed k-means obtained higher performance and outperformed sequential k-means while clustering	Hadoop and MapReduce	MapReduce programming model for Hadoop duster is a recent and popular trend in analyzing large data sets in short span of time. It is important to parallelize clustering algorithms using MapReduce for efficiency in clustering result in terms of execution time. This work proposed a parallel k-means algorithm using MapReduce for document clustering
Research trends on big data in marketing: a text mining and topic modeling based literature analysis [2]	2018	Given the research interest on big data in marketing, we present a research literature analysis based on a text mining semi-automated approach with the goal of identifying the main trends in this domain	Big data	This research literature analysis focused on the application of big data in marketing, in an attempt to identify the trends in these applied domains through different dimensions
Partition based clustering of large datasets using MapReduce framework: an analysis of recent themes and directions [7]	2018	To provide a comprehensive review of Hadoop and MapReduce and their components to compare recent research works on partition-based clustering algorithms which use MapReduce as their programming paradigm	Hadoop and MapReduce	This paper focuses on recent technologies for partition-based document clustering

(continued)

Table 1 (continued)

Title	Year	Objective	Technology used	Finding
Text mining with Lucene and Hadoop: document clustering with updated rules of NMF non-negative matrix factorization [8]	2018	Proper alignment of document files is to be labeled, when large number of files increases characterizing the files are needed, therefore, here comes the clustering of data, i.e., document clustering	MapReduce	A new processing techniques mainly in stemming algorithms that is iterated Lovins stemmer algorithm have 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"
MapReduce based analysis of sample applications using Hadoop [10]	2018	The rate of increase of structured, semi-structured, and unstructured data is very high. To discover hidden Information from different types of data is a big challenge. The two techniques, word frequency count and string matching, are applied on a single node and multi-node cluster with an input data set	Cloud computing, Hadoop, HDFS, MapReduce	This paper shows how to operate, manage, process, and analyze structured, semi-structured, and unstructured data by exploiting word count and string matching applications on Hadoop by varying MapReduce configuration

(continued)

Table 1 (continued)

Title	Year	Objective	Technology used	Finding
Distributed document clustering analysis based on a hybrid method [4]	2017	PSO is used to take advantage of its global search ability to provide optimal centroids which aids in generating more compact dusters with improved accuracy. This proposed methodology utilizes Hadoop and MapReduce framework which provides distributed storage and analysis to support data intensive distributed applications	Hadoop, MapReduce	MR-P k-means was proposed to overcome the inefficiency of PK means (or big data sets). The proposed method can efficiently be parallelized with MapReduce to process very large data sets. In MR-PK means, the clustering task that is formulated by k-means algorithm utilizes the best centroids generated by PSO
Performance evaluation of word frequency count in Hadoop environment [11]	2017	The research objective of this study is to measure the execution time on different sizes of text files by performing a simple MapReduce simulation on the word count program which is very popular in the big data arena. Also, an improved version of the word count program has been designed and simulated using the same set of text files	Big data analytics, HDFS, MapReduce parallel processing	The revolution of big data analytics has made a remarkable impact on all spheres of our modern living and also in the IT sector. The Hadoop MapReduce framework has found many important applications in big data analytics. In this present study, a simple and an improved version of word count programs have been simulated in the Hadoop MapReduce environment using different file sizes

(continued)

Table 1 (continued)

Title	Year	Objective	Technology used	Finding
Text document tokenization for word frequency count using rapid miner (taking resume as an example) [12]	2015	RapidMiner is unquestionably the world leading open-source system for data mining. It is available as a standalone application for data analysis and as a data mining engine for the integration into own products. Tokenization is the process of breaking a stream of text up into words, phrases, symbols, or other meaningful elements called tokens	RapidMiner	In this paper, the word frequency count of text document is done using RapidMiner tool—transform case and tokenize operators, with their interconnection. In order to find the frequency of occurrence of particular word, the user has to scroll the scrollbar
A distributed data mining system framework for mobile internet access log based on Hadoop [13]	2015	Single node-based data mining platform has been unable to store and analysis the massive data. According to cloud computing technology, we preset a distributed data mining framework based on Hadoop. Then, we present the implementation of this system framework and process mobile Internet access los on the Hadoop cluster	Cloud computing, Hadoop, Hive	We can write complex MapReduce of logic query language efficiently and quickly using Hive. Compared to traditional databases, Hive supports only a small set of primitive data types. Some of Hive's extensions are in view

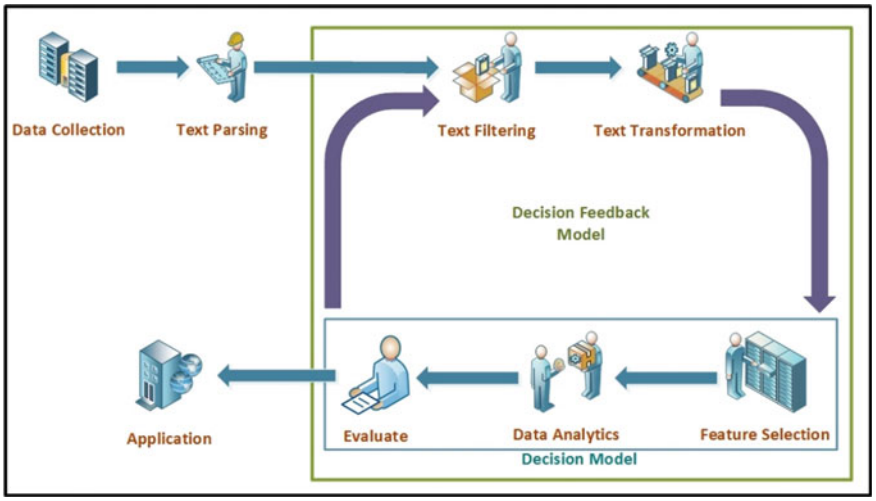


Fig. 5 Proposed model of text mining

Text Filtering In the parsed document, there will be some words which are not relevant to the mining process and those words need to be filtered out from the document called as word stopping and word stemming [14, 15]. This process requires an in-depth knowledge of the domain. Number of word stemmed are denoted by "S". The word stemming process has been discussed in more detail in further sections.

Text Transformation After text filtering, the document is presented by the occurrences words contained in it. After transformation, a document can be represented in two ways such as

- A simplified representation used in information retrieval and natural language processing where it contains the multiset of words irrespective of grammar is known as a bag of word. It is a JSON object representation.
  Bow1 = {"John": "3", "is": "1", "Good": "5"}
- Vector space model is an algebraic representation of text involving two steps. First, the document is represented in a vector of words and then the vector is transferred into a numerical format where the techniques of text mining can be applied. In this research, the documents have been represented in a vector space mode.

Feature Selection It is also known as variable selection in which we select a subset of more important features to be considered in the model creation. Irrelevant and redundant features are not to be used in model creation to improve the model accuracy.

Data Mining At this stage, the traditional data mining process is merged with text mining. Classical data mining techniques are used for clustering of the data that obtained from the quantitative representation of document to be associated in further

evaluation steps. K-mean clustering [9] or a parallel k-mean clustering [13] technique is taken into consideration in this phase.

Evaluate In this step, we evaluate the mining result. After evaluation, if the result is not acceptable, then we discard the result and continue the process as an iterative model to get the best results. Once the result is acceptable, we proceed to next step.

In this step, word stem factor (WSF) is calculated to decide the result acceptance. Word stem factor (WSF) is defined as the percentage of number of word stemmed to total number of distinct word. Word Stem Factor (WSF) = (S/T) * 100

Application The evaluated model is now have a broader area of application in the different text mining process. This model is ready as a product to be deployed in real-life problems. The model can be applied in web mining, E-consultation in medical, Twitter data analysis, and resume filtering.

4 Big Data Technologies

In hope of using data in future organizations collect and store by organizations store enormous amount of data. A number of significant global challenges have been notified as revolution in big data technologies [16]. The way organizations are collecting, using, managing, and leveraging data using big data technologies is ways beyond of imagination. In this research, we have focused on the most popular big data technology—Hadoop. It is one of the most sophisticated and ever growing ecosystems in the era of big data. Different technologies of Hadoop ecosystem have been briefly discussed.

4.1 Hadoop Distributed File System

To store huge amount of data in cluster of computers and to channel them to the required applications at a high bandwidth Hadoop distributed file system (HDFS), it is used inside Hadoop ecosystem. Large cluster constituting hundreds and thousands of server nodes built of commodity hardware to execute user application tasks [16, 17]. Storage and computation are distributed across servers and the system provides a technique of parallel processing and the required resource for each node have the capability to grow with demand while cost remains economical at every size. Data is stored in files and files are placed on nodes providing replication for fault tolerance. Some unique features of HDFS are highlighted below.

- Physical location of node is considered in rack awareness for storage allocation and task scheduling.

- Minimal data motion that process is moved to data rather than moving data to process. This technique reduces bandwidth.
- The previous versions of storage are restored using standby name node and secondary name node in case of human or system errors.

4.2 MapReduce

As a parallel processing framework, Hadoop MapReduce is used for processing huge amount of data in very less time. Large amount of data are processed clusters containing thousands of node built from commodity hardware. The cluster is highly reliable and fault tolerant. Job tracker is the single master nose and multiple task trackers acting as slave node constitutes the initial architecture of Hadoop framework. Whereas yet another resource negotiator (YARN) is the advanced Hadoop architecture [10, 13]. Resource manager is responsible for job scheduling on slave nodes, monitoring the task execution, and re-executing the failed tasks. Some more advantages of MapReduce are mentioned below.

- Commodity hardware is added to the existing server to increase the capacity is also known as scale-out architecture or horizontal scaling.
- Failed tasks are automatically recovered proving the fault tolerance of cluster.
- Flexibility for amount of file systems and facility of serialization in multiple open frameworks.
- Intelligent data placing technique to maintain the load balancing with maximum utilization and efficiency.

A MapReduce process is shown in Fig. 6. The data split files are executed in mapper parallely. After mapper phase is completed, the interim results are sorted and shuffled. Then, the results are merged and fed to the reducer. The no of reducer is defined in the MapReduce program determines the number of output files.

4.3 Pig

Pig is a data flow tool used for analyzing large data sets. It is not specific to Hadoop only rather it can be used with any parallel data processing. Though it can support all types of data that is structured, semi-structured, quasi-structured, and unstructured data but very frequently used for structured and semi-structured data. It uses Pig Latin language [18]. Each line in Pig code is converted to a logical plan and series of MapReduce tasks. It creates a directed acyclic graph (DAG) for each job. Features:

- Pig provides ease of programming where developers have to write less number of coded than MapReduce for a particular requirement.
- In case built-in functions are not available, users can create custom programming which can be easily integrated with Pig.

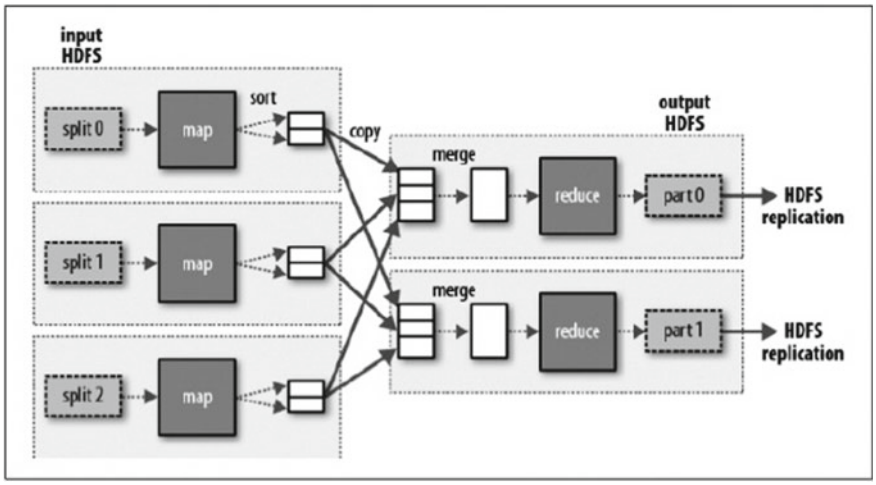


Fig. 6 MapReduce processing

4.4 Hive

As a data warehouse software Apache Hive inside Hadoop ecosystem helps to query, analyze, and manage large data sets stored distributed storage (HDFS). It provides the facility of HIVEQL, an SQL-like language for querying and retrieving data. All the Hive queries are converted into MapReduce job by Hive engine automatically and implicitly. When it is difficult to express logic in HIVEQL, it allows MapReduce programmers to be plugged in with Hive using custom mappers and reducers [17].

Hive allows indexing to provide acceleration in data search. Compaction and Bitmap indexes are also applicable in Hive. It supports different file types like plain text, RCFile, and ORC. It can operate on compressed data storages using GZIP, BZIP2 an SNAPPY. User-defined functions (UDFs) are supported by Hive when built-in functions are not available.

4.5 Sqoop

The facility to transfer data between HDFS and RDBMS (MySQL and ORACLE) is provided by Sqoop inside the Hadoop ecosystem. It imports data from RDBMS to HDFS to process it and again export the data to RDBMS [18]. It facilitates the connection of different database servers, controlling of import and export process. It can import data to Hive and HBase.

4.6 Oozie

Hadoop ecosystem provides the facility of a Web application based on Java used for scheduling Hadoop jobs is known as Apache Oozie. It sequentially combines multiple jobs to one logical unity of work. It supports MapReduce jobs, Pig scripts, Hive query, and Sqoop import exports. Jobs of a specific system like Java or a shell script program can also be scheduled in Oozie. Oozie workflow and Oozie coordinator are two categories of Oozie jobs. Multiple workflow and coordinators are bundled in Oozie to manage the lifecycle of running jobs. It is scalable and reliable.

4.7 Flume

Flume is used for efficient collection, aggregation, and movement of large amount streaming data like record logs. It has failover and recovery mechanism and it is used for online analytic application. Flume has a new data set sink Kiite API that is used to write data to HDFS and HBase.

4.8 ZooKeeper

Zookeeper is a centralized configuration and synchronization service in Hadoop ecosystem [17]. Every time a service is scheduled a lot of configuration need to be changed and resources are synchronized and this makes the service more fragile. Zookeeper is very fast with workloads with the ideal read-write ratio of 10:1. It can be replicated over multiple servers to avoid single point of failure.

5 Word Stemming

In context of information retrieval and linguistic morphology stemming, it is the process of tumbling any transformed word to its original stem word. Stem word is the base or morphological root form of any word. Stemming is a process that maps all related words to its stem [14]. Word stemming is an essential part of natural language processing and it is done by removing any suffix or prefix attached to the stem word. This conversion is also required in text clustering, categorization, and summarization as part of pre-processing in text mining.

5.1 Pre-requisites for Stemming

Word stemming requires tokenization and filtering from the document first. These two processes bring the document into the granular level required for word stemming.

Tokenization In tokenization, a document is split into a set of word based on some tokenizer or separator [12]. The separators can be a blank space or any special character. An example is illustrated as below.

Text = "Science brings the society to the next level."

The output of tokenization assuming blank space (" ") as a separator: ["Science", "brings", "the", "society", "to", "the", "next", "level"].

The punctuation marks and non-text characters are removed from the document in tokenization. Hence, the words are finally converted to nouns, verbs, etc. Another approach of word tokenization is focused on the statistical distribution of the words inside the document instead of following the occurrences of words. In the statistical analysis, it is important to index the texts into vectors. In this research, as the bag of word (BOW) approach has been adopted part of statistical representation of document.

Filtering This process removes the words which are not important for text mining process or which may degrade the result of analysis and it is also called as stop word filtering. Stop words [19] are the words which are not required in the text mining process. This filtering is controlled as per the requirement, i.e., a strong stop word list will create the best result in text mining process. The stop word lists are available in World Wide Web. One of the resource available in http://www.lextek.com has been considered in this research [20].

5.2 Classification of Stemming

Stemming algorithms are broadly classified into three groups. The classification of stemming algorithm is shown in Fig. 7.

Truncating Method This method removes the prefix and suffix of a word. Truncate (n) is the most basic stemming algorithm in which each nth position word is truncated and words existing in positions less than n are not truncated as well as no stemming rule is applied on them. Therefore, the chance of over stemming is increased. Another stemming algorithm where the plural words are transferred to singular form by removing the suffix 'S' [14]. There are four types of algorithm in truncating method as highlighted below.

Lovins Stemmer This algorithm contains 294 ending rules, 35 transformation rules, and 29 conditions. The longest suffix from any word can be removed by this stemmer.

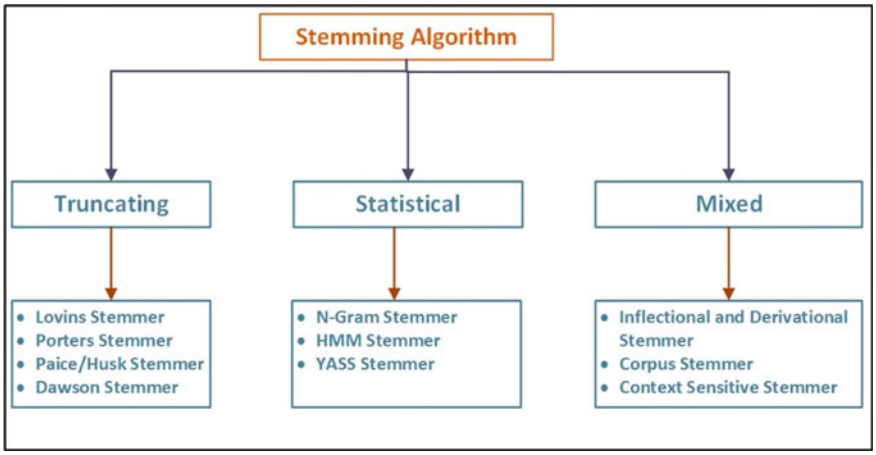


Fig. 7 Classification of stemming

After removing the suffix from the word, the word is referenced with different tables to convert it to a valid stem or root word after making some adjustments [15]. As a single pass algorithm maximum of one suffix is removed from a word. This algorithm can transfer the double letter words like "setting" to its original stem words very fast, i.e., "set" and also handles many asymmetrical plural forms to their singular transformations, for instance, "feet to foot," "men" to "man," etc. Lovins stemmer algorithm consumes more data and many suffixes are not available in the ending rules. Sometimes, it is very unreliable as it cannot match the stems of similar meaning.

Porter Stemmer Porter stemmer algorithm was proposed in 1980. Many modifications have been suggested and done on the elementary algorithm. There are 1200 suffix rules in the algorithm having five steps in each rule. The algorithm is iterated through the rules until one of them is accepted. Once a rule is satisfied, the suffix from the word is removed, then the resultant stem word is returned and next step is performer [15]. Also, there are 60 comprehensive conditions in this algorithm in the form of <Conditional Rules> with <Suffix> constitutes a <New Suffix>. For example, if a word ends with "EED" and has at least one consonant and vowel then the suffix can be changed to "EE." For instance, "Emceed" will be changed to "Emcee" but "Speed" will remain as it is. Porter stemmer algorithm is designed as a detail stemming framework where the key intension of the framework is that the programmers can develop new stemming rules for different sets of suffix.

Paice/Husk Stemmer It contains 120 rules indexed by suffixes and is iterative in nature. In each iteration, algorithm tries to find a match with the suffix and then either the suffix is deleted or it is replaced. Advantage of this algorithm is that it takes care of both deletion and replacement. But this is a very heavy algorithm which may create over stemming error [new paper].

Dawson Stemmer This is an extension of Lovins stemmer algorithm which have a 1200 extensive list of suffix transformation [15]. It is also a single pass algorithm therefore it is fast. The suffix is stored in reverse order indexed by their length and last letter.

Statistical Method These types of stemming algorithm remove the affixes (suffix and prefix) after applying any statistical analysis and technique. N-Gram stemmer, HMM stemmer, and YASS stemmer are statistical stemming algorithms. N-Gram stemmer is language independent and is based on n-gram and string comparison [14]. HMM stemming algorithm is unsupervised and language-independent stemming and it is based on hidden Markov model. YASS stemming corpus based and can be implemented without knowing the morphology. It uses hierarchical clustering and distance measure approach.

Mixed Method This type of stemming algorithms are composition of inflectional and derivational morphological methods, corpus-based methods, and context-sensitive methods [15]. As part of inflectional methods, the algorithms are correlated to syntactic variations such as plural, cases, and genders of a specific language. Krovetz and Xerox stemmers are example of inflectional and derivational methods. Corpus-based methods use the occurrences of word variants. Some drawbacks of Porter stemmer algorithm have been taken care here like "Iteration" is not converted to "Iter" and "General" is not converted to "Gener".

6 Proposed Porter Stemmer with Partitioner Algorithm (PSP)

This algorithm has about 60 rules which can be coded using MapReduce. When "Partitioner" technique is applied with all the porter rules, it provides better result. In partitioner of MapReduce, multiple partitions [21] are created based on conditions for data before data goes to reducer. The simplest partition technique is a hashing partition, but based on the condition, we can create required number of partition. For example, if special characters are not required for text mining process, then we can separate them in one partition and other alphabets and numbers will be in another partition. For this technique, the number of reducers needs to be set in the MapReduce program. Figure 8 shows the model for the proposed algorithm Porter stemmer partitioner which combines the rules of Porter stemmer implemented in MapReduce partitioner.

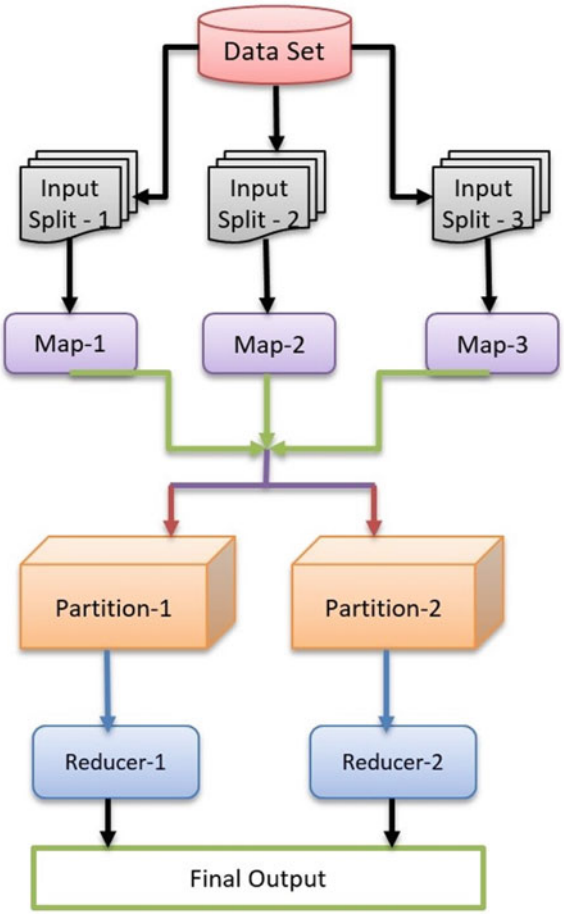


Fig. 8 Proposed Porter stemmer algorithm with partitioner (PSP)

7 Hadoop Cluster Operation Modes

For this research, the selected documents have unstructured format of data. Therefore, Hadoop MapReduce and HDFS have been chosen for implementation. The selected documents are stored in HDFS and a MapReduce program is run on each document parallel [22]. For the purpose, a Hadoop cluster with Hadoop Archirtecture-2 has been set up. A Hadoop can run in three different modes as shown in Fig. 9.

Standalone Mode Standalone mode is the default operation mode of a Hadoop cluster also known as local mode. In this mode, none of the demons like name node, resource manager, secondary name node, data node, and node manager run inside the cluster. Therefore, it is mainly used for learning, debugging, and testing [23]. In this mode, the cluster runs faster than the two other modes. In this mode, HDFS storage architecture is not utilized, so it is like a system having the same kind of storage

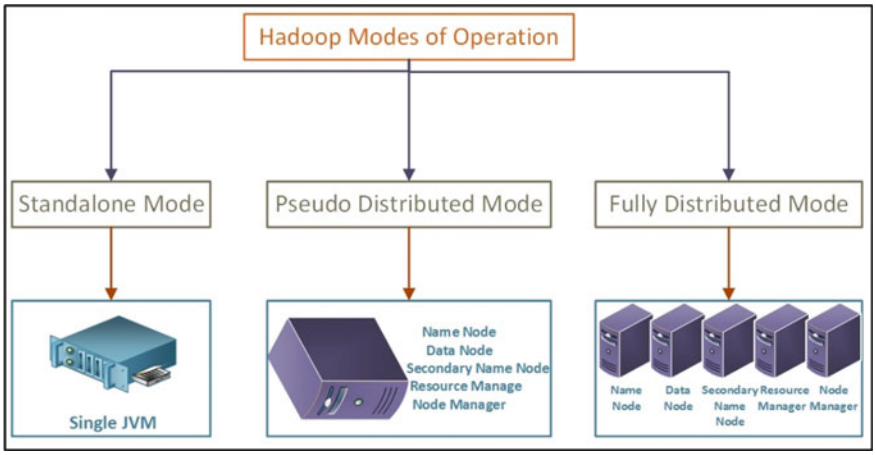


Fig. 9 Hadoop cluster operation modes

as in windows like an NTFS or FAT32 system. When this mode starts to run none of the configuration files like mapred-site.xml, hdfs-site.xml, core-site.xml, etc., are needed. All the processes run in a single JVM in this mode.

Pseudo-Distributed Mode In pseudo-distributed operation mode, all the demons run on a single node. This mode is a simulation of the cluster, therefore, all the processes run independently. Name node, resource manager, secondary name node, data node, and node manager run on separate Java virtual machines (JVMs) inside a single node. This mode mimics the operation of fully distributed mode on a single node [23].

The master-slave architecture of Hadoop cluster also exists in this mode is handled by a single system. Resource manage and name node are run as master, whereas data node and node manager run as slave. The secondary name node in this mode is used to handle the hourly back up of the name node. When this mode starts to run the configuration files (core-site.xml, mapred-site.xml, and hdfs-site.xml) need to be set up in the environment.

Fully Distributed Mode This is the production mode of Hadoop cluster where multiple nodes are used. Some of the nodes run master demons resource manager and name node, whereas rest of nodes in the cluster run slave demons node manager and data node. The HDFS storage architecture is fully followed here therefore the files are stored on multiple nodes [23]. The configuration parameters of the cluster environment need to be specified in this mode. This mode is highly scalable supporting both horizontal and vertical scaling. Also, this mode is completely reliable, fault tolerant and have the full capability of distributed computing.

Standalone mode has a very limited scope, whereas fully distributed mode is highly expensive and need a lot of configurations to be handled. Therefore, for this

research, a pseudo-distributed cluster mode has been chosen. The chosen mode is a Horton works pseudo-distributed cluster running on Hadoop-2 architecture.

8 Environment Setup

A Hadoop cluster has been set up for implementation taking the Hortonworks Hadoop 2.2 version. It provides a command line interface to interact with the cluster and an easy accessible Web interface for displaying cluster-related informations.

Commands to make up the Hadoop cluster [24]. Figure 10 shows the Hadoop version installed on the cluster.

As the used Hadoop architecture is a second generation architecture, five demons always run on the cluster to make it operational [25, 26]. The running demons are shown in Fig. 11.

- Name node
- Data node

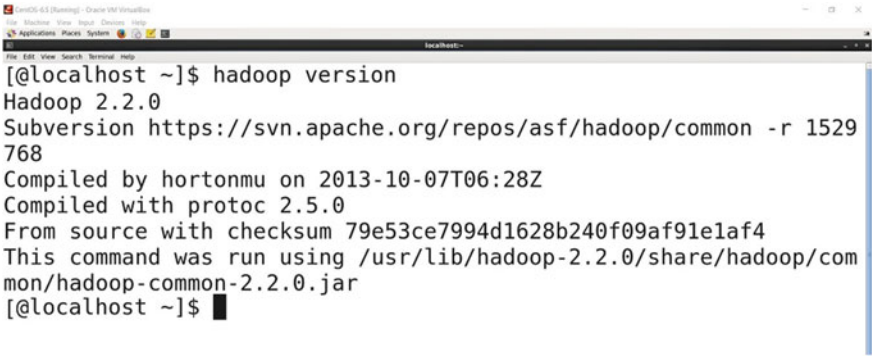
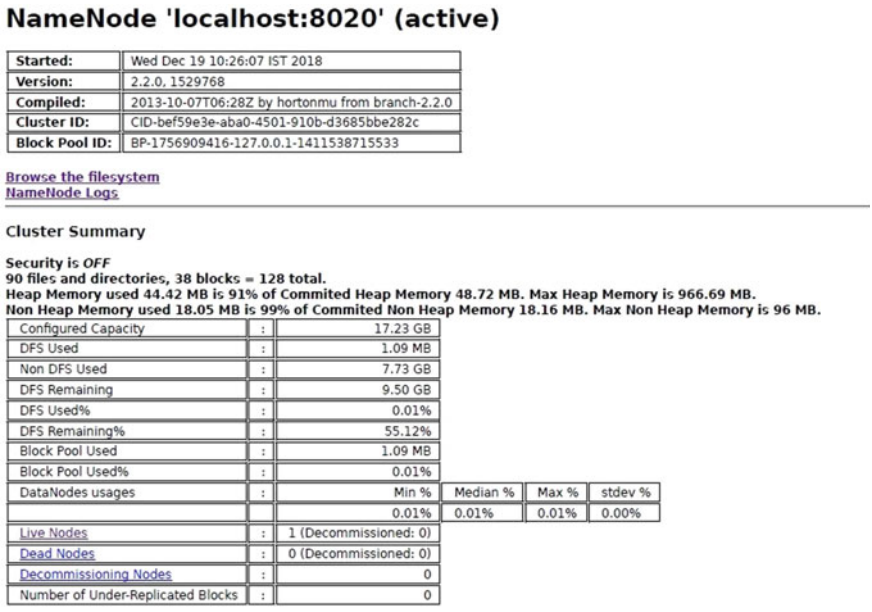


Fig. 10 Installed Hadoop version

Fig. 11 Running demons on Hadoop cluster

- Node manager
- Resource manage
- Job history server.

Information about the name node are shown in Figs. 12 and 13. The name node runs on port 8020. There are total 38 blocks in the cluster. The cluster have 10.60 GB storage for Hadoop distributed file system out of total ~18 GB storage. Figure 14 shows internal storage structure of HDFS. This server has a block size of 128 MB and the files are stored as part files inside the blocks of HDFS. Part files are the logical partitioning of a bigger data set [24, 26]. The replication factor of cluster is



NameNode 'localhost:8020' (active)

Started:	Wed Dec 19 10:26:07 IST 2018
Version:	2.2.0, 1529768
Compiled:	2013-10-07T06:28Z by hortonmu from branch-2.2.0
Cluster ID:	CID-bef59e3e-aba0-4501-910b-d3685bbe282c
Block Pool ID:	BP-1756909416-127.0.0.1-1411538715533

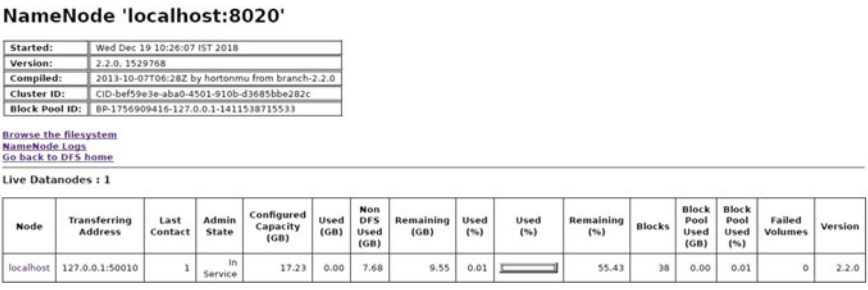
Browse the filesystem
NameNode Logs

Cluster Summary

Security is *OFF*
90 files and directories, 38 blocks = 128 total.
Heap Memory used 44.42 MB is 91% of Commited Heap Memory 48.72 MB. Max Heap Memory is 966.69 MB.
Non Heap Memory used 18.05 MB is 99% of Commited Non Heap Memory 18.16 MB. Max Non Heap Memory is 96 MB.

Configured Capacity	:	17.23 GB			
DFS Used	:	1.09 MB			
Non DFS Used	:	7.73 GB			
DFS Remaining	:	9.50 GB			
DFS Used%	:	0.01%			
DFS Remaining%	:	55.12%			
Block Pool Used	:	1.09 MB			
Block Pool Used%	:	0.01%			
DataNodes usages	:	Min %	Median %	Max %	stdev %
		0.01%	0.01%	0.01%	0.00%
Live Nodes	:	1 (Decommissioned: 0)			
Dead Nodes	:	0 (Decommissioned: 0)			
Decommissioning Nodes	:	0			
Number of Under-Replicated Blocks	:	0			

Fig. 12 Name node information-1



NameNode 'localhost:8020'

Started:	Wed Dec 19 10:26:07 IST 2018
Version:	2.2.0, 1529768
Compiled:	2013-10-07T06:28Z by hortonmu from branch-2.2.0
Cluster ID:	CID-bef59e3e-aba0-4501-910b-d3685bbe282c
Block Pool ID:	BP-1756909416-127.0.0.1-1411538715533

Browse the filesystem
NameNode Logs
Go back to DFS home

Live Datanodes : 1

Node	Transferring Address	Last Contact	Admin State	Configured Capacity (GB)	Used (GB)	Non DFS Used (GB)	Remaining (GB)	Used (%)	Used (%)	Remaining (%)	Blocks	Block Pool Used (GB)	Block Pool Used (%)	Failed Volumes	Version
localhost	127.0.0.1:50010	1	In Service	17.23	0.00	7.68	9.55	0.01		55.43	38	0.00	0.01	0	2.2.0

Fig. 13 Name node information-2

Goto : /tmp go

Go to parent directory

Name	Type	Size	Replication	Block Size	Modification Time	Permission
hadoop-yarn	dir				2014-09-24 11:46	rwxrwxrwx
partitions_0d09db61-0095-4ba0-9b10-d28674932007	file	153 B	1	128 MB	2014-09-29 19:22	rw-r--r--
partitions_1c630cec-4d2e-4d71-9c49-8d47c4ae15e7	file	153 B	1	128 MB	2014-09-29 19:24	rw-r--r--
partitions_2dc8649e-e659-4bc1-bef6-2b3708fabb1f	file	153 B	1	128 MB	2014-09-29 19:22	rw-r--r--
partitions_3150588f-ebaf-47d9-b651-aba13c77d07e	file	153 B	1	128 MB	2014-09-29 19:33	rw-r--r--
partitions_470591af-e561-4bed-a185-ba99d3045bbb	file	153 B	1	128 MB	2014-09-30 17:29	rw-r--r--
partitions_558af272-15a4-454a-a46f-ce2626c556eb	file	153 B	1	128 MB	2014-09-29 19:11	rw-r--r--
partitions_6e64cd18-b3ab-40bb-b0fe-4f45a2296966	file	153 B	1	128 MB	2014-09-30 17:40	rw-r--r--
partitions_792dcf70-c9e6-49f1-9f7b-73690755edb9	file	153 B	1	128 MB	2014-09-29 19:02	rw-r--r--
partitions_7d9d7132-ac58-4b78-a12d-fc463b42605d	file	153 B	1	128 MB	2014-09-30 19:11	rw-r--r--
partitions_84843b53-c0b5-4618-9166-14a9c857eca8	file	153 B	1	128 MB	2014-09-29 19:04	rw-r--r--
partitions_9703047c-92ca-4721-9367-76721914935d	file	153 B	1	128 MB	2014-09-29 19:36	rw-r--r--
partitions_98c394c9-8781-4cf0-8fe7-f00e1856fe43	file	153 B	1	128 MB	2014-09-29 19:00	rw-r--r--
partitions_9a5140c4-afeb-47fb-a38d-113484df5743	file	153 B	1	128 MB	2014-09-29 19:35	rw-r--r--
partitions_bd105f44-8133-4b62-9845-2fb574fd3c1f	file	153 B	1	128 MB	2014-09-30 12:47	rw-r--r--
partitions_df97fdb2-7533-4bca-9687-99f2c473042b	file	153 B	1	128 MB	2014-09-29 19:11	rw-r--r--
partitions_ee8ca7b1-6968-4844-bcdf-48261def63fe	file	153 B	1	128 MB	2014-09-30 17:12	rw-r--r--

Fig. 14 HDFS storage structure

set to 1, therefore, every file is present in a single rack only according to the rack awareness of Hadoop.

9 Implementation

Implementation of this research follows all the steps of proposed text mining model. Implementation of this research has compared the stemming performance of Lovins stemmer algorithm, Porter stemmer algorithm, and Porter stemmer with partitioner algorithm.

9.1 Data Collection

Three different data sets have been considered for the implementation of this research. All data sets are of different sizes and have different structures of data as described below.

Data Set-1 (CV Data Set) A CV structure has been considered as the first and smallest data set for this research. It has the text that is relevant to a CV like technologies, expertise, work experience, etc. This data set has been collected from an open source [27] of size 2 KB and total 260 words. The data set has the text data so it is unstructured in nature. A portion of the data set has been shown in Fig. 15.

Data Set-2 (Speech data set) Speeches have the most complex linguistic morphology. The second data set has been considered as a speech data set of PMO

CHRISTOPHER MORGAN
ADDRESS: 177 GREAT PORTLAND STREET, LONDON W5W 6PQ
PHONE: +44 (0)20 7666 8555
EMAIL: CHRISTOPER.MORGAN@GMAIL.COM
OBJECTIVE PROVIDE ANALYSIS DATA SUPPORT IN A COMPANY AS DATA ANALYST.
WORK
EXPERIENCE 04/2014 - 04/018
DATA ANALYST, GHT COMPANY, MADRID SPAIN
RESPONSIBILITIES:
ESTABLISH OPERATION STRATEGY IN A TEAM FOR IMPROVING SALES
PREPARE DATA AND INFORMATION FOR MAKING REGULAR REPORT DATA ANALYSIS
PERFORM DATA ANALYSIS FOR COMPLEX DATA AND FILES JAVA JAVA JAVA JAVA JAVA JAVA
03/2012 – 05/2014
DATA ANALYST, STARTUP CORPORATION, MADRID SPAIN
RESPONSIBILITIES:
COMPOSED JAVA PROGRAM FOR INTERFACING WITH ORACLE DATABASE
PERFORMED DATA ANALYSIS ESPECIALLY FINANCIAL DATA
PERFORMED STATISTICAL DATA ANALYSIS USING STATA
SHOWED DATA ANALYSIS IN REGULAR MEETINGS FOR CREATING NEW PROGRAM JAVA JAVA JAVA JAVA JAVA
EDUCATION
2004 - 2008
BACHELOR DEGREE OF COMPUTER SCIENCE, TECHNICAL UNIVERSITY OF MADRID
2002 - 2004
CERTIFIED AS DATA ANALYST, DATA ANALYST CERTIFICATION, TECHNICAL UNIVERSITY OF MADRID

Fig. 15 Data set-1

India on 72nd Independence Day. Data has been collected from official site [28] of the India's Prime Minister. The data set has total 8000 words and unstructured in nature. A part of the data set is shown in Fig. 16.

Data Set-3 (Twitter data set) The third data set has been collected from the American microblogging site Twitter [29]. The data set contains social media comments and is the largest data set considered for this research. It have total ~52,00,000 words and is of 185 MB. A part of data set is shown in Fig. 17.

After the data sets are collected, they are transferred to HDFS, because data has to be present in HDFS for MapReduce processing. Command used to move data from local storage to HDF are given below.

- To check if file exists in local storage—"ls".
- To move file from local storage to HDFS—"hdfs dfs-copyFromLocal/home/local/textdata/textmining".

Figure 18 shows the data sets presence inside the HDFS.

9.2 Text Parsing

Text parsing is a technique to read the input data set and break it into granular levels which is a word. Text parsing is a logic that performs the above task inside

I hail from Gujarat. There is a saying in Gujarati 'Nishan Chuk Maaf Lekin Nahi Maaf Nichu Nishan' which means one should have big aims and dreams. However, for that, one has to strive hard and be answerable. But if the aims are not big, targets are not far-sighted, decisions are also not taken. Development comes to a halt. That is why my dear brothers and sisters, it is necessary for us to move ahead with big aims and resolve. When targets are vague, when the spirit is not strong, then important decisions in our social life also get held up for years. Take for instance the case of MSP – economists, farmer organizations, farmers as well as political parties had been demanding, that farmers should get an MSP which is one and a half times of their investment. The matter was debated for years, files moved to and fro, and but was stuck. Finally, we took the decision. We took a bold decision of giving the farmers the MSP which is one and half times their investment. There was unanimity on GST. Everyone wanted GST but they could not arrive at a decision because while on the subject, they were thinking in terms of their vested interests and whether this will translate into electoral gains. Today, with the help of small traders, their open mindedness and their attitude of accepting the new, the country has implemented GST. A new found confidence has been generated in the business community. The small entrepreneurs, small businessmen who faced teething difficulties in adopting GST, accepted the challenge and the country is now moving ahead. Today, we have enacted laws on Insolvency and Bankruptcy to strengthen the banking sector. Who opposed them earlier? Taking decisions requires conviction, force, confidence and complete dedication to the good of the common man. Why was the Benami property law not enforced earlier? The Benami property laws are implemented only when there is courage and a determination to do something for the country. The Jawans of our defence forces were demanding one-rank one-pension for several decades. They were not resorting to agitation because they are disciplined, but nobody was paying heed to their voices. Somebody needed to take a decision in this regard. You gave us the responsibility of taking this decision and we fulfilled it positively. My dear brothers and sisters, we are not the kind of people who work in the interest of the party. We are capable of taking tough decisions because national interest is topmost in our priority.

Fig. 16 Data set-2

As from title. What kind of visa class do I have to apply for, in order to work as an academic in Japan ?
What kind of Visa is required to work in Academia in Japan?
visajob-search
Which online resources are available for job search at the Ph.D. level in the computational chemistry field
As a computational chemist, which online resources are available for Ph.D. level jobs?
job-search
As from title. Not all journals provide the impact factor on their homepage. For those who don't where can I find their impact factor
Where can I find the Impact Factor for a given journal?
journalsbibliometrics
8 I have seen many engineering departments want professional engineer registration. Why do they care?
8 In U.S., why do many engineering departments care about professional engineer registration?
8 job-searchengineering
What is the h-index, and how does it work
What is the h-index exactly and how does it work?
bibliometrics
8 If your institution has a subscription to Journal Citation Reports (JCR), you can check it there. Try this URL:http://isiknowledge.com/jcr
If I publish a pre-print paper on arXiv, how can I guarantee exclusive rights to the publisher afterwards? Am I unable to publish on non-open access journals after I publish a pre-prin
Does publishing a paper on arXiv prevent me to submit it to a non-open access journal?
copyrightarxiv
2 An increasing number of funding organisations require publications on the research that they fund to be open access, i.e. available to the public without having to subscribe to a jour

Fig. 17 Data set-3

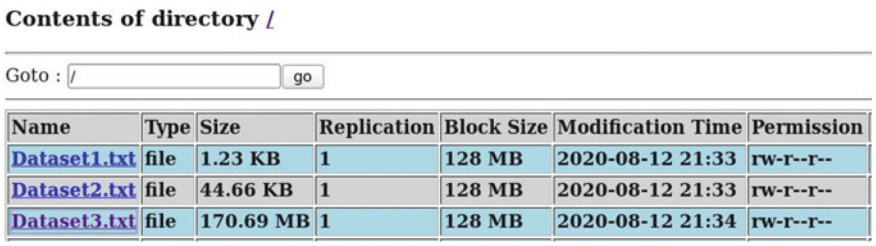


Fig. 18 Data sets in HDFS

a MapReduce program [11]. As data in all the data set is separated by space we have used line offset value and string tokenizer [12] to parse the data sets. Parsing of data sets produce a bag of words (BoW) pseudocode for text parsing is defined as TEXT-PARSING(A).

TEXT-PARSING(A)

```
1    tokenizer ← tokenize(line)
2      for j ← 1 to length[tokenizer]
3       A[j] ← tokenizer[j]
```

For example assuming "My name is xyz" is a line.

Tokenize converts the line into an array of words by splitting them based on blank space. For the above example, tokenizer will create ['My', 'name', 'is', 'xyz'] for a line "My name is xyz".

Text parsing is done in mapper side and all the further steps are done in reducer side.

9.3 Text Filtering

Text filtering removes the unexpected words from the bag of words. It is done by passing the tokenized array into a stop word filter. For example, the word "an" do not contribute any morphological interpretation in the analysis. Therefore, it needs to be removed from the analysis. When a word passes through the stop word list than it is checked for its presence in the list [12]. If the word is present in the stop word list, then it is removed from the bag. Pseudocode for text filtering defined as TEXT-FILTERING(A) is explained below.

```
TEXT-FILTERING(A)
1       stopwords[] =["a","as", "able", "about", "above", "accord-
        ing","accordingly","across","actually","after","afterwards","again",
        "against", "aint" .....]
2       for k ← 1 to length[A]
3         flag = true
4         for i ← 1 to length[stopwords]
5           if A[k] = stopwords[i]
6             flag = false
7             if flag = true
8             Insert A[k] into a new array B
```

After stop word removal three stemming algorithms such as Lovins stemmer, Porter stemmer, and proposed partitioned Porter stemmer (PSP) have been used for implementation of word stemming process. The proposed PSP stemmer takes care of the punctuation marks, special characters, etc., which are also not relevant to our analysis. Therefore, in the proposed PSP stemmer, a partitioner program has been used in this research. It removes all the special characters, punctuations into an unused partition, and only consider the words containing alphabets. Post-partitioning the terms are passed to the stop word list. Number of word stemmed are denoted by "S". The comparison of stemming results has been discussed in further sections.

9.4 Text Transformation

After the text filtering, the document is now converted to a numerical matrix form call as document matrix [19]. The symptomatic presentation of different terms has been explained below.

Term Frequency (λ) It is defined as the total occurrence of a stem word with respect to the total number of distinct words present in a document. Whereas the total occurrence of a root or stem word is defined as *Term Count*.

Term Count = Total count of existence of a stem word in a document.

Term frequency = Total count of occurrence of a word in document/Total Number of Word in document.

If N = Total number of word in document.

T = Term count then term frequency ($\boldsymbol{\lambda}$) = T/N

Document Matrix It is a numerical representation of the document. After finding the term frequency [12] of each unique term in document, the document is presented in form of matrix as [Word (Term) Term frequency (F)] which constructs the document matrix. Figure 19 shows a portion of document matrix of data set-2.

Fig. 19 Document matrix of data set-"2"

CONTRIBUTING	1742.5
CONTRIBUTION	1161.6666
CONVEY	3485.0
CONVICTION	3485.0
COOKING	3485.0
CORNER	3485.0
CORRIDORS	3485.0
CORRODED	3485.0
CORRUPT	3485.0
CORRUPTION	1161.6666
COST	3485.0
COULD	1161.6666
COUNCILS	3485.0
COUNTED	3485.0
COUNTRIES	1742.5
COUNTRY	50.507248
COUNTRYMEN	112.41936
COUNTRY'S	580.8333
COUPLE	3485.0
COURAGE	1742.5
COURSE	3485.0
COURT	3485.0

Word Stem Factor (α) The percentage of total number of word stemmed with respect to the total unique word present in a document is defined as word stem factor. Algorithms providing higher percentage of stemming are known as *Dense Stemmer*.

S = Total number of word stemmed

U = Total unique word present in document

α = (S/U) * 100

Stop Word Factor (β) It is defined as the percentage of total number of word stopped with respect to the total unique word present in a document.

X = Number of stopped words

β = (X/U) * 100

Cumulative Word Stem Factor (γ) It is defined as the percentage of total number of word stemmed and stopped with respect to the total unique word present in a document.

$\gamma = ((S + X)/U) * 100$
and $\gamma = \alpha + \beta$

9.5 Feature Selection

In this research, attributes such as "term," "frequency," "word stem factor," and "stop word factor" have been considered.

9.6 Evaluate

The values obtained from the model has been accepted to complete the evaluation process. The values obtained have been analyzed in further sections.

10 Result and Discussion

Table 2 shows a comparative study for word stemming capacity of the three stemming algorithms. From the result, it is clear that Porter stemmer with partitioner algorithm provides dense stemming than Lovins stemmer and Porter stemmer. Also, PSP is more accurate in stop word filtering. This performance improvement is applicable to documents of all sizes. Figure 20 shows the graph plotted for word stem factor with respect to the different stemming algorithms. From the graph, it is observed that with increase of data set size, PSP algorithm shows better result which resolves the big data volumetric issue [18], i.e., the model provides the better result when operate on huge data set. Similarly, a graph is plotted between the stop word factor and the stemming algorithms as shown in Fig. 21. From the figure, it is observed that a Porter stemmer algorithm when operated with a partitioner provides better stopping capability than Lovins stemmer and Porter stemmer. Another graph between cumulative word stem factor and stemming algorithm is shown in Fig. 22. The plot clearly points toward the better performance of Porter stemmer with partitioner than the other stemming algorithms according to the increased size of data sets. The accuracy of analysis depends on the stop word list and word stemmed. So, the stop word list is continuously updated for better results.

11 Conclusion and Future Work

From the above result analysis, it is clear that Porter stemmer algorithm with Hadoop MapReduce partitioner provided better result than Lovins stemmer and traditional Porter stemmer algorithm. Therefore, PSP algorithm can be used with big data to create an operation module which can be used in industrial applications, health care, social media, etc. The Porter stemmer with partitioner is capable of providing better result for huge amount of data sets than other stemming algorithms. The

Table 2 Comparison of stemming results

Result set	Data set	Algorithm	Total unique word in document (U)	Total word stemmed (S)	Total word stopped (X)	Word stem factor (α)	Stop word factor (β)	Cumulative word stem factor (γ)
1	CV	Lovins stemmer	221	17	15	7.49	6.79	14.48
2	CV	Porter stemmer	220	38	15	17.27	6.82	24.09
3	CV	Porter stemmer with partitioner	204	55	15	26.96	7.35	34.31
4	PM speech	Lovins stemmer	3484	304	254	8.72	7.29	16.02
5	PM speech	Porter Stemmer	3347	1028	254	30.71	7.59	38.3
6	PM speech	Porter stemmer with partitioner	3297	1075	277	32.6	8.4	41.01
7	Twitter	Lovins stemmer	5,209,760	28,756	10,015	0.55	0.19	0.74
8	Twitter	Porter stemmer	5,166,699	336,965	10,015	6.52	0.19	6.72
9	Twitter	Porter stemmer with partitioner	5,166,699	1,120,535	12,882	21.68	0.25	21.94

proposed methodology also has an extensible capability of reducing unnecessary words from the text mining and also has the capability to reduce the error in the following an iterative approach. The model can be used for CV filtration, online exam evaluation of subjective question answer, sentiment analysis, etc. In future, the model and algorithm will be implemented in other application domains such as health care and the obtained results will be compared. Also, the optimization techniques like particle swarm optimization (PSO) will be applied to enhance the model.

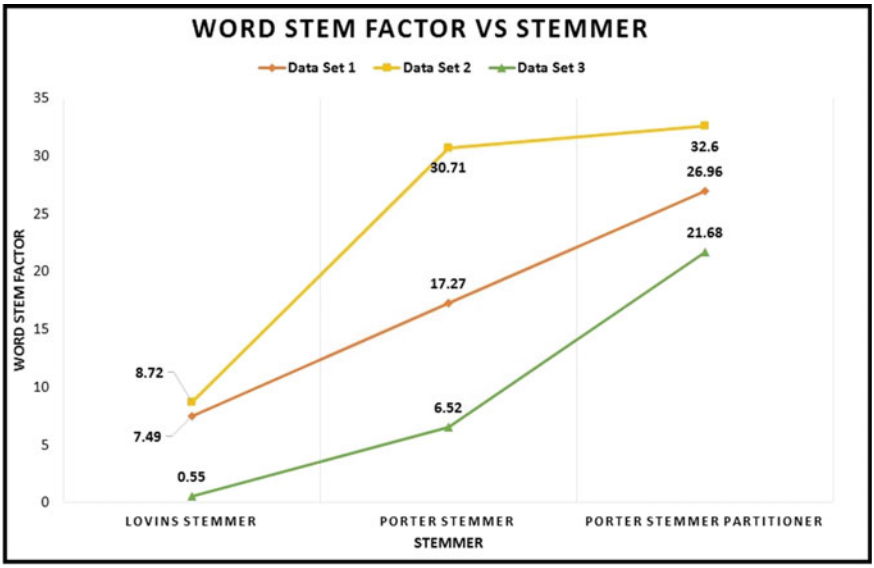


Fig. 20 WSF versus stemmer

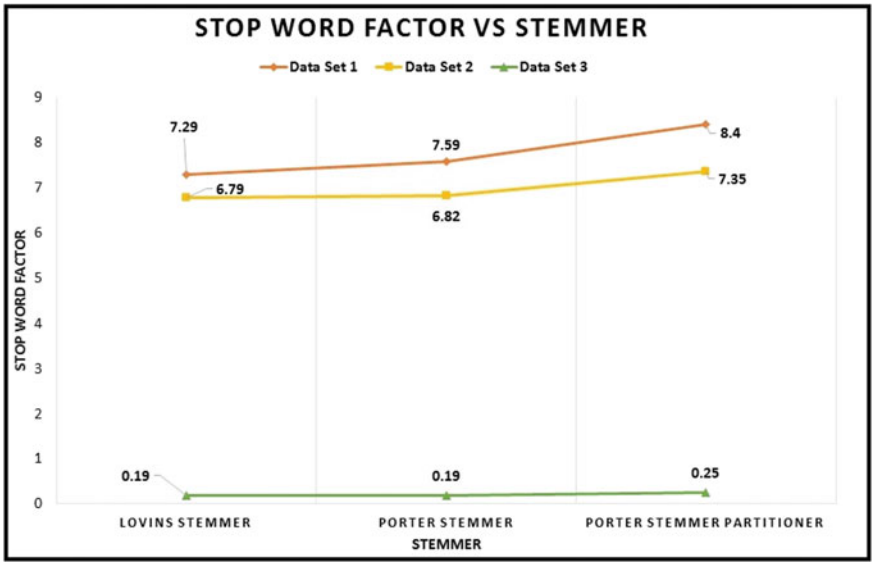


Fig. 21 SWF versus stemmer

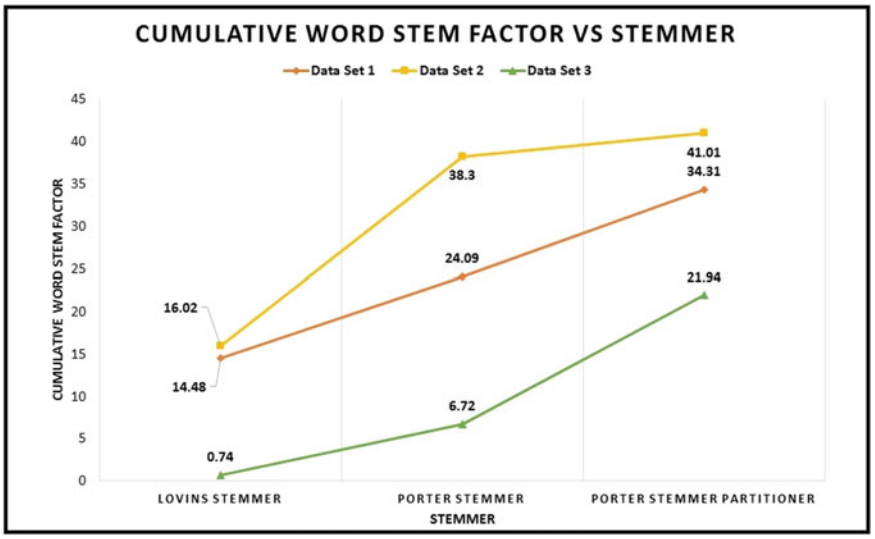


Fig. 22 CWSF versus stemmer

References

1. Sardar, T.H., and Z. Ansari. 2018. An analysis of MapReduce efficiency in document clustering using parallel K-means algorithm. *Future Computing and Informatics Journal* 3: 200–209.
2. Alexandra, A., C. Paulo, and M. Paulo. 2018. Research trends on big data in marketing: A text mining and topic modeling based literature analysis. *European Research on Management and Business Economics* 24: 1–7.
3. Sarkar, D. 2018. *Text Analytics with Python*, 109–319. New York: Apress Publication.
4. Judith, J., and J. Jayakumari. 2018. *Distributed Document Clustering Analysis Based on a Hybrid Method*, 131–142. New York: China Communications, IEEE.
5. Ramanujam, R.S., and R. Nancyamala. 2015. *Sentiment Analysis Using Big Data.*
6. Jain, A.K. 2010. Data clustering: 50 years beyond K-means. *Pattern Recognition Letters* 31: 651–666.
7. Sardar, T.H., and Z. Ansari. 2018. Partition based clustering of large data sets using MapReduce framework: An analysis of recent themes and directions. *Future Computing and Informatics Journal* 3: 247–261.
8. Lydia, E., and D. Ramya. 2018. Text mining with Lucene and Hadoop: Document clustering with updated rules of NMF non-negative matrix factorization. *International Journal of Pure and Applied Mathematics* 118: 191–198.
9. Ding, C., X. He, H.D. Simon. 2005. On the equivalence of nonnegative matrix factorization (NMF) and spectral clustering. In *International Conference on Data Mining*, 606–610. SIAM.
10. Ghazi, M.R., N.S. Raghava. 2018. MapReduce based analysis of sample applications using Hadoop. In *International Conference on Application of Computing and Communication, Technologies*, 34–44. Berlin: Springer.
11. Madasamy, K., and M. Ramaswami, 2017. Performance evaluation of word frequency count in Hadoop environment. *International Journal of Innovative Research in Science, Engineering and Technology.*
12. Gupta, G., and S. Malhotra. 2015. Text document tokenization for word frequency count using rapid miner (taking resume as an example). In *IJCA Proceedings on International Conference on Advancements in Engineering and Technology.*

13. Jiang, Y., J. Yang, L. Tang, Y. Liu, X. Zhao, and X. Hao. 2015. A distributed data mining system framework for mobile internet access log based on Hadoop. *Transactions on Edutainment* 5: 243–252.
14. Singh, A., N. Kumar, G. Sahil, and A. Mittal. 2010. Achieving magnitude order improvement in Porter stemmer algorithm over multi-core architecture. In *International Conference on Informatics and Systems*. IEEE.
15. Arianti, N.D., I. Mohamad, U. Syaripudin, D. Mariana. 2019. Porter stemmer and cosine similarity for automated essay assessment. In *International Conference on Computing Engineering and Design*. IEEE.
16. Jiang, H., K. Wang, Y. Wang, M. Gao, and Y. Zhang. 2016. *Energy Big Data: A Survey*. IEEE, 3844–3861.
17. Reinsel, D., J. Gantz, J. Rydning. 2017. *Data Age 2025—The Evolution of Data to Life-Critica*. Seagate-WP.
18. Singh, K., K. Kaur. 2014. *Hadoop—Addressing Challenges of Big Data, International Advance Computing Conference*. IEEE.
19. Kim, J., and K. Chung. 2019. Associative feature information extraction using text mining from health big data. *Wireless Personal Communications* 105: 691–707.
20. http://www.lextek.com/manuals/onix/stopwords1.html.
21. Khawlaab, T., M. Fatihaa, Z. Azeddineb, and N. Said. 2018. *A Blast Implementation in Hadoop MapReduce Using Low Cost Commodity Hardware*, 69–75. Amsterdam: Elsevier.
22. Mishra, B. 2016. Improved MapReduce K mean clustering algorithm for Hadoop architecture. *International Journal of Advanced Trends in Computer Science and Engineering*.
23. Li, H., and X. Lu. 2015. Challenges and trends of big data analytics. In *International Conference on P2P, Parallel, Grid, Cloud and Internet Computing*. IEEE.
24. Wankhede, P., and N. Paul. 2016. Secure and multi-tenant Hadoop cluster —an experience. In *International Conference on Green High Performance Computing*. IEEE.
25. https://docs.cloudera.com/documentation/enterprise/6/6.3/toics/cm_mc_service_config_overview.html.
26. https://docs.cloudera.com/HDPDocuments/.
27. www.freecv.com.
28. https://www.pmindia.gov.in/en/news_updates/pms-address-to-the-nation-from-the-ramparts-of-the-red-fort-on-the-72nd-independence-day/?comment=disable.
29. https://archive.org/download/stackexchange.