UNDERSTANDING SENTIMENT ANALYSIS FROM SOCIAL MEDIA MESSAGES

Project Report submitted in partial fulfillment of
The requirements for the degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE And ENGINEERING

Of

MAULANA ABUL KALAM AZAD UNIVERSITY OF TECHNOLOGY

By

SONALI PRIYA, ROLL NO.-93, WBUT ROLL NO.-10900115093

SOFIA JAMIL, ROLL NO.- 92, WBUT ROLL NO.- 10900115092

ANKITA, ROLL NO.-16, WBUT ROLL NO.- 10900115016.

ZAID ALAM, ROLL NO.-121, WBUT ROLL NO.- 10900115121

Under the guidance of

Dr.Piyali Chatterjee

DEPARTMENT OF COMPUTER SCIENCE And ENGINEERING

NETAJI SUBHASH ENGINEERING COLLEGE
TECHNO CITY, GARIA, KOLKATA – 700 152

2018-2019

CERTIFICATE

This is to certify that this project report titled **Understanding Sentiment Analysis from Social Media Messages** submitted in partial fulfillment of requirements for award of the degree Bachelor of Technology(B. Tech)in Computer Science And Engineering of Maulana Abul Kalam Azad University of Technology is a faithful record of the original work carried out by,

SONALI PRIYA, **Roll no**. 10900115093, **Regd. No.** 151090110094 Of 2015-16

SOFIA JAMIL, **Roll no**. 10900115092, **Regd. No.** 151090110093 Of 2015-16

ANKITA, **Roll no**. 10900115016, **Regd. No.** 151090110016 Of 2015-16

ZAID ALAM, **Roll no**. 10900115121, **Regd. No.** 151090110122 Of 2015-16

under my guidance and supervision.

It is further certified that it contains no material, which to a substantial extent has been submitted for the award of any degree / diploma in any institute or has been published in any form , except the assistances drawn from other sources, for which due acknowledgement has been made.

Date:………..

Dr. Piyali Chatterjee	Dr. Piyali Chatterjee
Head of Department	Associate professor
Computer Science And Engineering	Computer Science And Engineering
Netaji Subhash Engineering College	Netaji Subhash Engineering College
Techno city, Garia,	Techno city, Garia,
Kolkata-700152	Kolkata-700152

DECLARATION

We hereby declare that this project report titled **Understanding Sentiment Analysis from Social Media Messages** is our own original work carried out as a under graduate student in Netaji Subhash Engineering College except to the extent that assistances from other sources are duly acknowledged.

All sources used for this project report have been fully and properly cited. It contains no material which to a substantial extent has been submitted for the award of any degree/diploma in any institute or has been published in any form, except where due acknowledgement is made.

Student's names:	Signatures:	Dates:
SONALI PRIYA ………………………..	………………………..	……………………
SOFIA JAMIL ………………………..	………………………..	……………………
ANKITA ………………………..	………………………..	……………………
ZAID ALAM ………………………..	………………………..	……………………

CERTIFICATE OF APPROVAL

We hereby approve this dissertation titled **Understanding Sentiment Analysis from Social Media Messages** carried out by:

SONALI PRIYA, **Roll no**. 10900115093, **Regd. No.** 151090110094 Of 2015-16

SOFIA JAMIL, **Roll no**. 10900115092, **Regd. No.** 151090110093 Of 2015-16

ANKITA, **Roll no**. 10900115016, **Regd. No.** 151090110016 Of 2015-16

ZAID ALAM, **Roll no**. 10900115121, **Regd. No.** 151090110122 Of 2015-16

under the guidance of Dr. Piyali chaterjee of Netaji Subhash Engineering College, Kolkata in partial fulfillment of requirements for award of the degree Bachelor of Technology (B. Tech) in Computer science & engineering of Maulana Abul Kalam Azad University of Technology.

Date:………..

Examiners' signatures:

1. ………………………………………….
2. ………………………………………….
3. ………………………………………….

Acknowledgement and/or Dedication

A study or a project of this volume can never be the outcome of a single person or just a mere group of dedicated students, so we express our profound sense of gratitude to those who extended their whole hearted help and support to us in completing our project because successful completion of any work requires guidance and help from a number of people. Firstly, it gives us immense pleasure to acknowledge our institute **NETAJI SUBHASH ENGINEERING COLLEGE** for providing us an opportunity in developing a project **Understanding Sentiment Analysis from Social Media Messages.** In addition, we wish to express our deep sense of gratitude to **Dr. Piyali Chatterjee CSE,**for permitting us to carry out this project work and for her guidance, support and for her special concern and providing sufficient information related to the topic, which helped us in completing the project work in time.

SONALI PRIYA

SOFIA JAMIL

ANKITA

ZAID ALAM

Dated:…………………

Table of Contents

ABSTRACT

World Wide Web has become a huge repository containing views, feeling, opinion etc. Using social media, such as Twitter, Facebook etc., people express their opinion etc. about a fact or interact with their friends which reflects their sentiments in most of the cases. These data contain useful information which may give insight about a fact. To extract such data, analysis of sentiment is becoming a promising area of research. It aims at identifying different views or opinionative data in the web and classifying them according to their positive or negative nature. *Sentiment Analysis* is a problem of text based analyzer but there are challenges that make it difficult as compared to traditional text based analysis. These clearly states that there is a need of attempt to work towards these problem and also gives a direction for handling negations, hidden sentiments , slangs, polysemy etc.

The online social networking platforms (e.g. Facebook, Twitter, Linkedin) where users are enabled to send short texts/messages and express opinions on specific topics and their sentiments on them have increased rapidly in the last few years. The requirement for data analysis techniques that process posts online and assist with extracting interesting patterns of knowledge from them is stronger than ever.

This project will collect tweets of the user in real time and thus provide a basis to identify each tweet into positive and negative based on the mindset of user thereby providing a real time analysis of the users regarding a certain topic.

In this work, we are going to present a system for scalable and real time sentiment analysis of twitter using apache Pig. Sentiment analysis refers to the task of natural language processing to determine whether a piece of text contains some subjective information and what subjective information it expresses, i.e., whether the attitude behind this text is positive, negative or neutral using Afinn Dictionary. Understanding the opinions behind user-generated content automatically is of great help for commercial and political use, among others. The task can be conducted on different levels, classifying the polarity of words, sentences or entire documents.

CHAPTER 1

INTRODUCTION

1.1 SOCIAL MEDIA

In the advent and rapid rise in social networking, people started to use and share information through various social networking sites like facebook, twitter, Instagram etc. Social Media is a group of internet-based applications that improved on the concept and technology of web 2.0 and enable the formation and exchange of user generated content such as text posts or comments, digital photos or videos, and data generated through all online interactions, is the lifeblood of social media. Social media facilitate the development of online social networks by connecting a user's profile with those of other individuals or groups. SM is an important source of learning of opinions, sentiments, subjectivity, assessments, approaches, evaluation, influences, observations, feelings, borne out in text, views, blogs, discussions, news, remarks, reactions, or some other documents. Before the vent of SM, the homepages was popularly used in the late 1990s which made it accessible for wage internet users to share information. However, the activities on today's SM seem to have transformed the World Wide Web into its intended original creation.

User's opinions/sentiments on SM such as Twitter, Facebook, Youtube and Yahoo are basically positive, negative or neutral (neutral being commonly regarded as no opinion expressed). Online opinions can be discovered using traditional methods but this in conversely inadequate considering the large volume of information generated on all SM sites as present.

1.2 SOCIAL NETWORK ANALYSIS

Social network analysis (SNA) is the process of investigating social structures through the use of ne- -tworks and graph theory . It characterizes networked structures in terms of *nodes* (individual actors, people, or things within the network) and the *ties* , *edges* , or *links* (relationships or interactions) that connect them.Social network analysis has emerged as a key technique in modern sociology.It has also gained a significant following in anthropology, biology, demography, communication studies , econo- -mics, geography, history information science, organizational studies, political science, social psycol- -ogy, development studies, sociolinguistics. In 1954, Barnes started using the term systematically to denote patterns of ties, encompassing concepts traditionally used by the public and those used by soc- -ial scientists: bounded groups (e.g., tribes, families) and social categories (e.g. gender, ethinicity). Considering that personality , which uniquely identifies each one of an , effects a lot of aspects of human behavior,mental processes and affective reactions,there is an enormous opportunity for adding new personality- based qualities to user interfaces . In personalized systems used in domains such as , e-learning information filtering and e commerce would greatly benefit from users interface that adapts the interaction using motivational strategies, presentation styles, interaction modalities and recomme- -dations.

1.3 BIG DATA HADOOP

Due to the advent of new technologies, devices, and communication means like social networking sites, the amount of data produced by mankind is growing rapidly every year. The amount of data produced by us from the beginning of time till 2003 was 5 billion gigabytes. If we pile up the data in the form of disks it may fill an entire football field. The same amount was created in every two days in 2011, and in every ten seconds in 2018. This rate is still growing enormously. Though all this information produced is meaningful and can be useful when processed, it is being neglected.

Big Data is a term used for a collection of data sets that are large and complex, which is difficult to store and process using available database management tools or traditional data processing applications. The challenge includes capturing, curating, storing, searching, sharing, transferring, analyzing and visualization of this data.

Hadoop is an open-source software framework for storing data and running applications on clusters of commodity hardware. It provides massive storage for any kind of data, enormous processing power and the ability to handle virtually limitless concurrent tasks or jobs.

1.4 Hadoop Architecture

- **Hadoop Common** – the libraries and utilities used by other Hadoop modules.
- **Hadoop Distributed File System (HDFS)** – The Java-based scalable system that stores data across multiple machines without prior organization.
- **YARN** – (**Yet another Resource Negotiator**) provides resource management for the processes running on Hadoop.
- **MapReduce** – A parallel processing software framework . It is comprised of two steps . Map step is a master node that takes inputs and partitions them into smaller sub-problems and then distributes them to worker nodes. After the map step has taken place, the master node takes the answers to all of the sub problems and combines them to produce output.

1.5 Apache Pig :

Pig is a high-level programming language useful for analyzing large data sets. In a Map-Reduce framework, programs need to be translated into a series of Map and Reduce stages. However, this is not a programming model which data analysts are familiar with. So, in order to bridge this gap, an abstraction called Pig was built on top of Hadoop.

Apache Pig enables people to focus more on analyzing bulk data sets and to spend less time writing Map-Reduce programs. Similar to Pigs, who eat anything, the Pig programming language is designed to work upon any kind of data. That's why the name, Pig!

1.6 Features of Pig :

Apache Pig comes with the following features:-

- **Rich set of operators** − It provides many operators to perform operations like join, sort, filer, etc.
- **Ease of programming** − Pig Latin is similar to SQL and it is easy to write a Pig script if you are good at SQL.
- **Optimization opportunities** − The tasks in Apache Pig optimize their execution automatically, so the programmers need to focus only on semantics of the language.
- **Extensibility** − Using the existing operators, users can develop their own functions to read, process, and write data.
- **UDF's** − Pig provides the facility to create **User-defined Functions** in other programming languages such as Java and invoke or embed them in Pig Scripts.
- **Handles all kinds of data** − Apache Pig analyzes all kinds of data, both structured as well as unstructured. It stores the results in HDFS.

1.7 Pig Architecture:

Pig consists of two components**:**

1. **Pig Latin**, which is a language.
2. **A runtime environment**, for running Pig Latin programs.

A Pig Latin program consists of a series of operations or transformations which are applied to the input data to produce output. These operations describe a data flow which is translated into an executable representation, by Pig execution environment. Underneath, results of these transformations are series of MapReduce jobs which a programmer is unaware of. So, in a way, Pig allows the programmer to focus on data rather than the nature of execution.

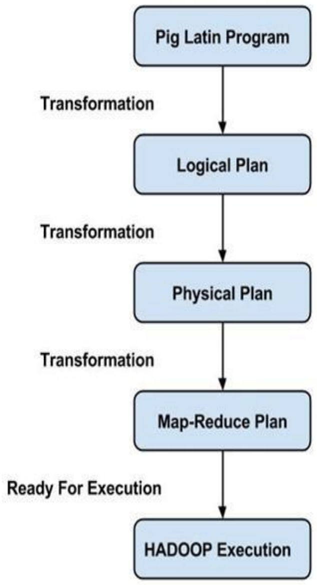


FIG.1

PigLatin is a relatively stiffened language which uses familiar keywords from data processing e.g., Join, Group and Filter.

1.8 Execution Mode

Pig has two execution modes:

1. **Local mode:** In this mode, Pig runs in a single JVM and makes use of local file system. This mode is suitable only for analysis of small datasets using Pig.

2. **Map Reduce mode :** In this mode , queries written in Pig Latin are translated into Map - Reduce jobs and are run on a Hadoop cluster (cluster may be pseudo or fully distributed). MapReduce mode with the fully distributed cluster is useful of running Pig on large datasets.

1.9 Map Reduce

MapReduce is a framework using which we can write applications to process huge amounts of data , in parallel, on large clusters of commodity hardware in a reliable manner.

1.9.1 What is MapReduce?

MapReduce is a processing technique and a program model for distributed computing based on java. The MapReduce algorithm contains two important tasks, namely Map and Reduce. Map takes a set of data and converts it into another set of data, where individual elements are broken down into tuples (key/value pairs). Secondly, reduce task, which takes the output from a map as an input and combines those data tuples into a smaller set of tuples. As the sequence of the name MapReduce implies, the reduce task is always performed after the map job.

The major advantage of MapReduce is that it is easy to scale data processing over multiple computing nodes. Under the MapReduce model, the data processing primitives are called mappers and reducers. Decomposing a data processing application into mappers and reducers is sometimes nontrivial. But, once we write an application in the MapReduce form, scaling the application to run over hundreds, thousands, or even tens of thousands of machines in a cluster is merely a configuration change. This simple scalability is what has attracted many programmers to use the MapReduce model.

1.9.2 The Algorithm

- Generally MapReduce paradigm is based on sending the computer to where the data resides!
- MapReduce program executes in three stages, namely map stage, shuffle stage, and reduce stage.
    - **Map stage** : The map or mapper’s job is to process the input data. Generally the input data is in the form of file or directory and is stored in the Hadoop file system (HDFS). The input file is passed to the mapper function line by line. The mapper processes the data and creates several small chunks of data.

- **Reduce stage** : This stage is the combination of the **Shuffle** stage and the **Reduce** stage. The Reducer's job is to process the data that comes from the mapper. After processing, it produces a new set of output, which will be stored in the HDFS.

- During a MapReduce job, Hadoop sends the Map and Reduce tasks to the appropriate servers in the cluster.
- The framework manages all the details of data-passing such as issuing tasks, verifying task completion, and copying data around the cluster between the nodes.
- Most of the computing takes place on nodes with data on local disks that reduces the network traffic.
- After completion of the given tasks, the cluster collects and reduces the data to form an appropriate result, and sends it back to the Hadoop server.

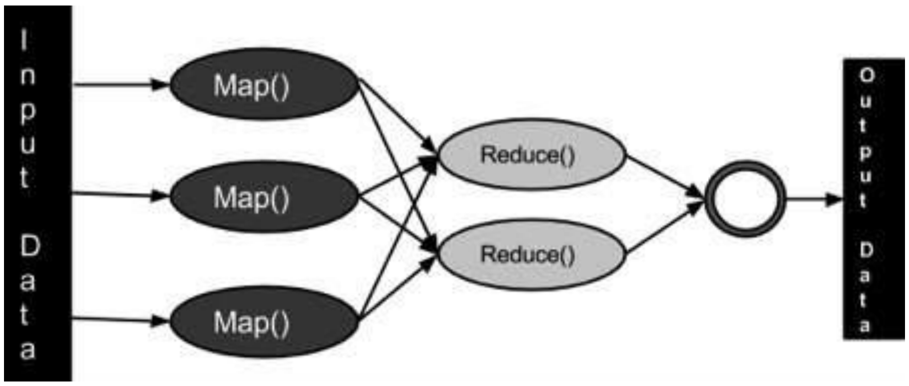


FIG 1.2

1.9.3 Inputs and Outputs (Java Perspective)

The MapReduce framework operates on <key, value> pairs, that is, the framework views the input to the job as a set of <key, value> pairs and produces a set of <key, value> pairs as the output of the job, conceivably of different types.

The key and the value classes should be in serialized manner by the framework and hence, need to implement the Writable interface. Additionally, the key classes have to implement the Writable-Comparable interface to facilitate sorting by the framework. Input and Output types of a MapReduce job: (Input) <k1, v1> -> map -> <k2, v2>-> reduce -> <k3, v3>(Output).

	Input	Output
Map	<k1, v1>	list (<k2, v2>)
Reduce	<k2, list(v2)>	list (<k3, v3>)

1.9.4 Terminology

- **PayLoad -** Applications implement the Map and the Reduce functions, and form the core of the job.
- **Mapper -** Mapper maps the input key/value pairs to a set of intermediate key/value pair.
- **NamedNode -** Node that manages the Hadoop Distributed File System (HDFS).
- **DataNode -** Node where data is presented in advance before any processing takes place.
- **MasterNode -** Node where JobTracker runs and which accepts job requests from clients.
- **SlaveNode -** Node where Map and Reduce program runs.
- **JobTracker -** Schedules jobs and tracks the assign jobs to Task tracker.
- **Task Tracker -** Tracks the task and reports status to JobTracker.
- **Job -** A program is an execution of a Mapper and Reducer across a dataset.
- **Task -** An execution of a Mapper or a Reducer on a slice of data.
- **Task Attempt -** A particular instance of an attempt to execute a task on a SlaveNode.

.

1.10 SENTIMENT ANALYSIS

Sentiment analysis can be defined as a computational process of identifying, mining user attitudes, opinions, views and emotions expressed in a piece of text, speech, tweets and database sources through Natural Language Processing(NLP) to determine the user’s attitude towards a particular topic. Sentiment analysis involves classifying opinions in text into categories like “positive” or “negative” or “neutral”. It’s also referred as subjectivity analysis, opinion mining, and appraisal extraction.

Analysis can be done at document, sentence and phrase level. In document level, the entire document is taken then it is analyzed whether the sentiment is positive, negative or neutral. Generally speaking, sentiment analysis aims to determine the attitude of a speaker, writer, or other subject with respect to some topic or the overall contextual polarity or emotional reaction to a document, interaction, or event. The attitude may be a judgment or evaluation, affective state (that is to say, the emotional state of the author or speaker), or the intended emotional communication (that is to say, the emotional effect intended by the author or interlocutor).

Existing approaches to sentiment analysis can be grouped into three main categories: knowledge-based techniques, statistical methods, and hybrid approaches. Knowledge-based techniques classify text by affect categories based on the presence of unambiguous affect words such as happy, sad, afraid, and bored.Some knowledge bases not only list obvious affect words, but also assign arbitrary words a probable "affinity" to particular emotions.Statistical methods leverage on elements from machine learning such as latent semantic analysis, support vector machines, "bag of words" and "Semantic Orientation—Pointwise Mutual Information". More sophisticated methods try to detect the holder of a sentiment (i.e., the person who maintains that affective state) and the target (i.e., the entity about which the affect is felt).To mine the opinion in context and get the feature about which the speaker has opined, the grammatical relationships of words are used. Grammatical dependency relations are obtained by deep parsing of the text. Hybrid approaches leverage on both machine learning and elements from knowledge representation such as ontologies and semantic networks in order to detect semantics that are expressed in a subtle manner, e.g., through the analysis of concepts that do not explicitly convey relevant information, but which are implicitly linked to other concepts that do so.

1.11 Apache Flume

Apache Flume is a Hadoop ecosystem component used to collect, aggregate and moves a large amount of log data from different sources to a centralized data store.
It is an open source component which is designed to locate and store the data in a distributed environment and collects the data as per the specified input key(s).

1.11.1 Flume Architecture

Before moving forward to know the working of flume tool, It is mandatory to know the Flume architecture first.

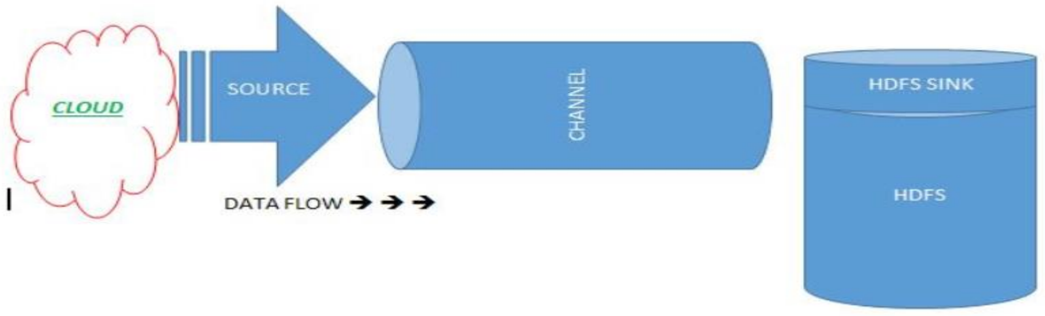


Fig 1.3

1.11.2 FLUME COMPONENTS

Flume is composed of the following components:

- **Flume Event:** It is the main unit of the data that is transported inside the **Flume** (Typically a single log entry). The basic unit of the data which is transported inside Flume is what we call an Event. Generally, it contains a payload of the byte array. Basically, that can be transported from the source to the destination accompanied by optional headers.

  A Flume event will be in the following structure:

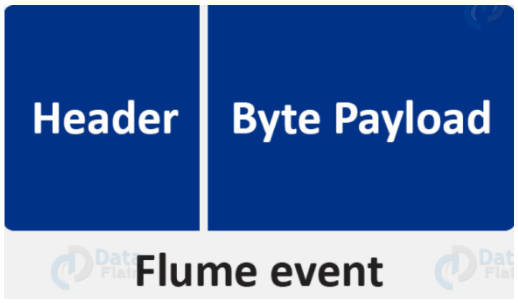


FIG 1.4

  When data is sent over the Internet, each unit transmitted includes both header information and the actual data being sent. The header identifies the source and destination of the packet , while the actual data is referred to as the **payload**.

- **Flume Agent:** Is an independent Java virtual machine daemon process which receives the data (events) from clients and transports to the subsequent destination (sink or agent).

- **Source:** Is the component of Flume agent which receives data from the data generators say, twitter,facebook,weblogs from different sites and transfers this data to one or more channels in the form of Flume event.
  The external source sends data to Flume in format that is recognized by the target Flume source.
  - Example, an Avro Flume source can be used to receive Avro data from Avro clients or other Flume agents in the flow that send data from an Avro sink, or the Thrift Flume source will receive data from a Thrift sink, or a Flume Thrift RPC client or Thrift Clients are written in any language generated from the Flume thrift protocol.

- **Channel:** Once, the Flume source receives an Event,it stores this data into one or more channel and buffers them till they are consumed by sinks. It acts as a bridge between the source and sinks. These channels are implemented to handle any number of sources and sinks.

- **Sink:** It stores the data into the centralized stores like HDFS and HBase.

CHAPTER 2

LITERATURE REVIEW

The existing model develops several simple Map/Reduce programs to analyze one provided dataset. The dataset contained 18 million Twitter messages captured during the London 2012 Olympics period. All messages are related in some way to the events happening in London (as they have a term such as London2012). Previous work are:

- Text analysis- You can find a Histogram plot that depicts the distribution of sizes (measured in number of characters) among the Twitter dataset.
- Time analysis- You can find a Plot with time series with the number of Tweets that were posted each day of the event (one bar per day).
- Hashtag analysis- During the Olympics the supporters from several countries were expressing their support by adding specific hashtags to its messages, in many cases with the form team__ (e.g. #teamgb). or, go___ (#gousa). You can find a table with all the team support hashtags you can find (based on these patterns), and the number of support messages.

2.1 Text Analysis

For this part of the assignment I implemented two different Hadoop jobs. First one – Text Analysis – retrieves the distribution of tweet lengths; and second one – Text AnalysisAvg – is responsible for calculating the average tweet length. TextAnalysis. First, I implemented a mapper class – TMapper, which emits key-value pairs in a form of (tweet message length, 1), where each tweet message is the last field of the tweet data string (retrieved with StringUtils). This allows to group values by tweet length and then sum up all the occurrences of tweets of a certain length, which is done in TReducer. Then, I ordered the keys and grouped them by 10 using Excel to produce a graph presented below. As there are values that exceed the allowed tweet length (140), I assumed that any results over 140 come from the tweets in foreign languages or containing some special characters.

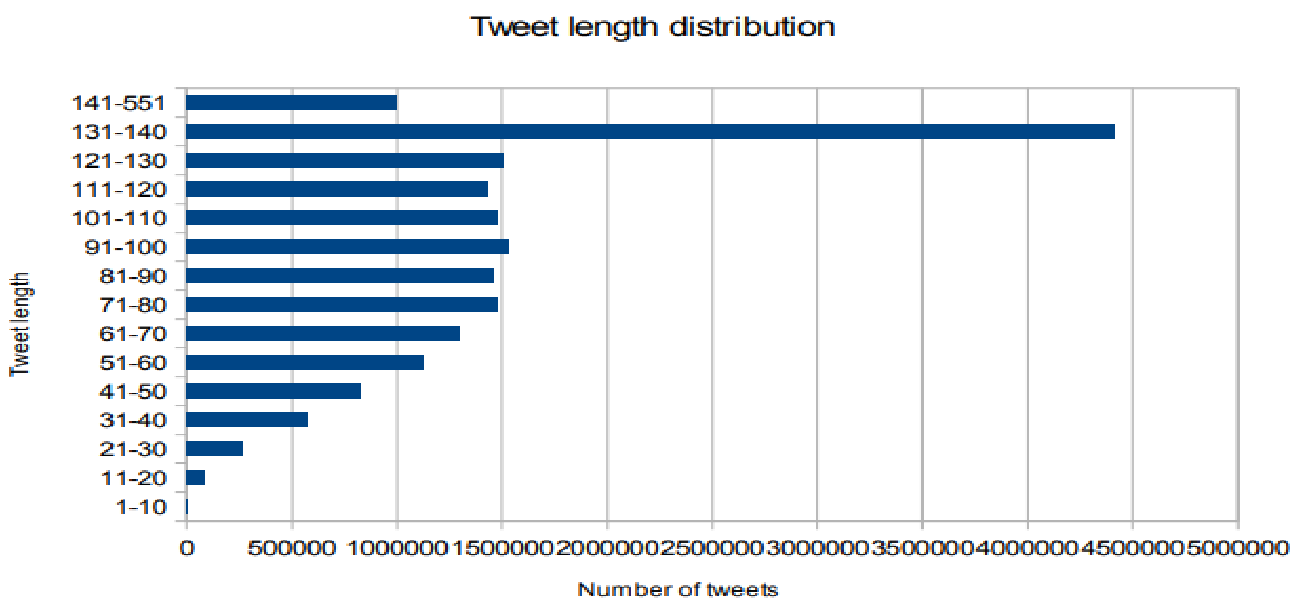


FIG 2.1

From the graph above it can be seen that the number of tweets that are over 140 characters long, is too significant to be ignored, which is why I decided to alter my code in order to do more accurate filtering. I replaced all non – alphanumeric characters in each tweet by "1" with a use of regular expressions for alphabetic and numeric characters , so that the tweet length would be calculated 1-10 11-20 21-30 31-40 41-50 51-60 61-70 71-80 81-90 91-100 101-110 111-120 121-130 131-140 141-551 0, 500000 1000000 1500000 2000000 2500000 3000000 3500000 4000000 4500000 5000000 Tweet length distribution Number of tweets Tweet length more precisely. There were still 481 tweets whose length was over 140 characters , which I assume comes from links included in tweet messages, but this number is almost negligible comparing to the overall lengths distribution. Apart from that, I implemented grouping by 5 in my mapper in order to avoid doing it manually in Excel. Finally, I added an if-else statement that enables a "no-care" mode for tweets that have over 140 characters and groups all of them together in the reducer – (>140, 1). I emitted each key-value pair in a form of (lower-bound length– upper-bound length, 1) for all tweets below 140 characters, which was done by dividing each tweet length by 5, rounding it to a higher of the 2 closest integers, and then multiplying it back by 5. I also calculated the lower bound which was done in relatively similar manner , but which less than the upper - bound by 4 . All these updates eliminated any need for manual work in Excel and the only thing that I needed to do was to do was to compose a graph from the output data,which is provided below. Even though there are significant changes in the graph, the overall shape of the curve (except for oversized tweets) still remains the same for this dataset, be it with or without special characters.

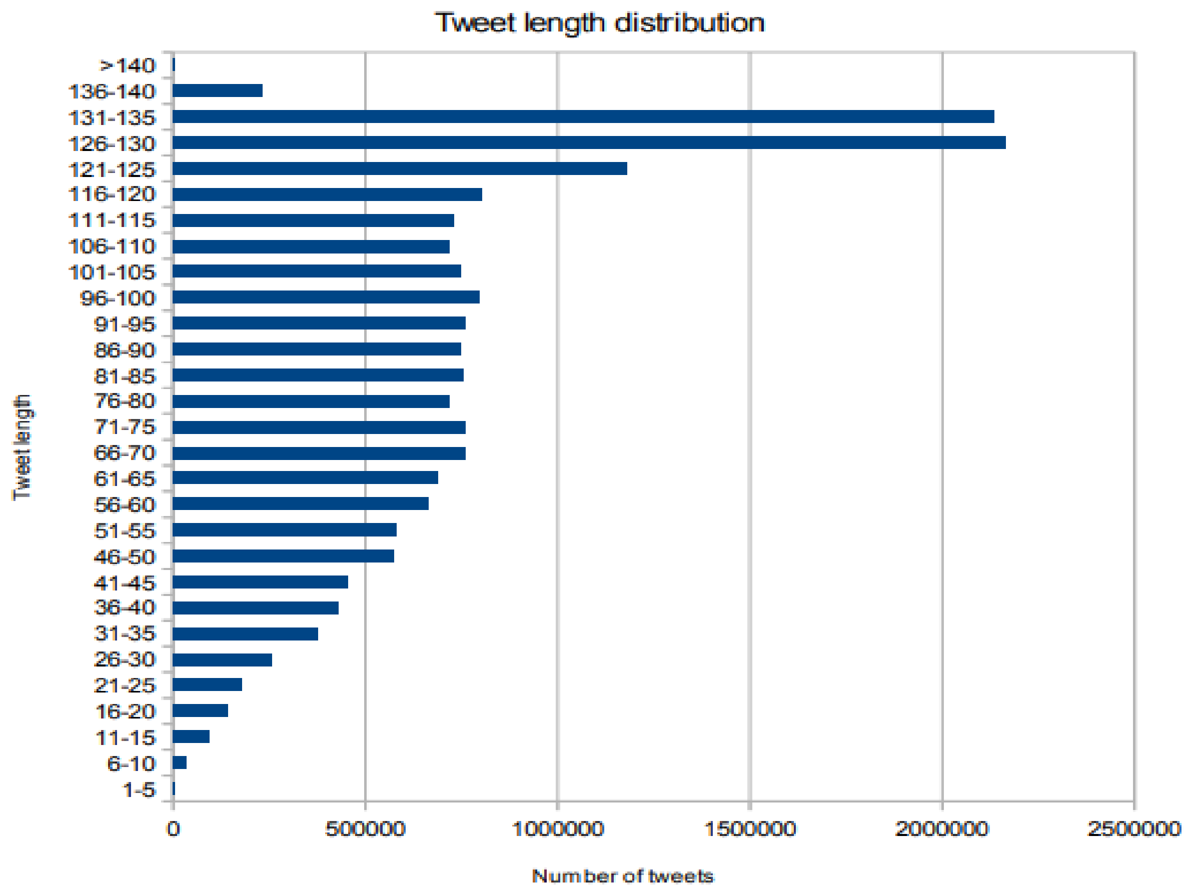


FIG 2.2

2.2 Text Analysis Avg

Within the mapper class – TAvgMapper I used Int Int Pair class to form aggregated values in order to emit key-value pairs in a form of (“avg”, (tweet message length, 1)). I specifically used a common “avg” key for all the values in order to combine them in the reducer class – TAvgReducer, in which sum of tweet message lengths was divided by sum of the occurrences of these lengths in order to achieve an average. My Hadoop job produced 100 as its result, which I then checked in Excel where the average tweet length was also roughly equal to 100 (100.45). I then altered my source code to get more accurate results by adding the same non-alphanumeric character replacement mechanism that I used in TextAnalysis, which resulted in the average of 92. This seems to be completely logical, as there were a lot of tweets with length over 140 characters, which are now much shorter due to filtering out special characters, and that is why the average gets more precise and even more balanced now.

2.3 Time Analysis

For part B as a mapper class I implemented TimeMapper and as a reducer class – TimeReducer. In TimeMapper first, I retrieved the tweet date string (using StringUtils). As for this task we only needed day, month and year, I partitioned the date string by commas and used only first part as a key value. From my mapper class I was emitting key-value pairs in a form of (tweet day-monthyear, 1). Then, in TimeReducer, I summed up all occurrences of each date and from the output of the Hadoop job I did compose a graph in Excel, which is presented below.

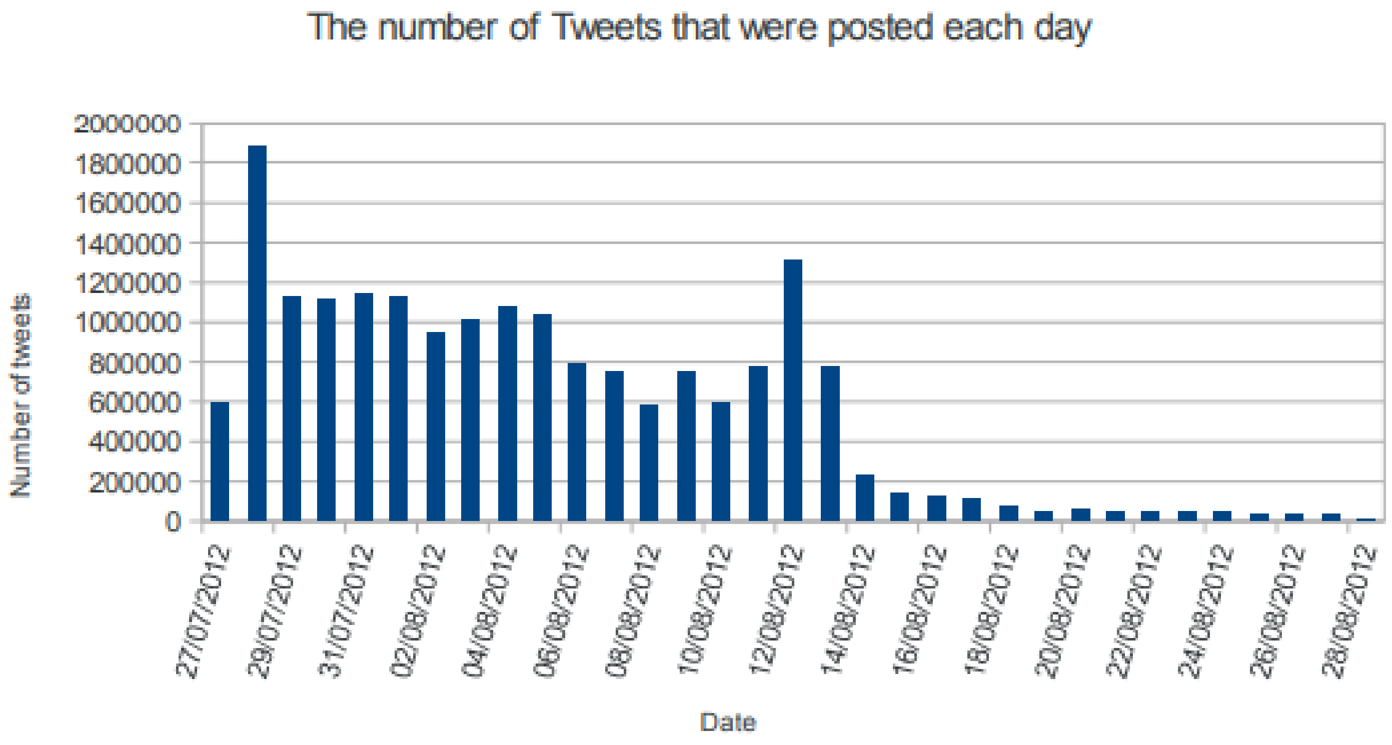


FIG 2.3

2.4 Hashtag Analysis

Most of the work for this part was done in the HMapper. We did pattern matching for hashtags (substrings starting with “#”) within the tweet messages, as when we tried to search among the tags themselves in a hashtag field, We encountered words containing “go” or “team” not only in the start of the tag, but also in the middle or in any part of the word. I specifically matched patterns starting with “#go” and “#team” by regular expressions in order to return all substrings of the tweet messages starting with “#go” or “#team” and ending with a whitespace. In order to group properly the teams, prefixes “# go” and “#team” were stripped off from the key values and key values were converted to lowercase strings, in order to obtain a cumulative number for hashtags for both of them. Moreover, in order to cover hashtags that look like”#goteam”, and to combine the values properly, we implemented a nested pattern matching against “#go” and then “team” in order to retrieve the actual name of the supported team. I decided to include the “#goteam” processing, as there was a significant number of tags in the output result starting with this prefix. Within the HReducer I simply added the number of hashtags encountered for each team. For some countries people misspelled country names, used slang names or added extra information in the end. For instance, for Australia's team people used “aus”, “aussie”, “aussies” and “australia”, whereas for Canada's team people tended to write “cana”, “canad”, “canadago” or the most popular – “canada”. As it seems too difficult to filter this automatically and accumulate the values in hadoop cluster, this was done manually in Excel. We also had to filter out the teams that did not resemble any country names (for example, “brijamada”). However, this was not that hard considering that the Even though there was an option to filter out the numbers of hashtags, outputting only those that are exceeding some specified minimum to make the output shorter, We decided not to take this approach, as We did not want to risk loosing some valuable data from the tweets dataset. Apart from that when We tried to limit hashtag numbers to start only from 100, it did not change significantly the output size. The table and graph presented below illustrate a list of countries against the number of supporting hashtags for each of them in the tweets dataset. It can be seen both through the graph and the table that the largest number of supporting tweets were received by gb team, followed by usa, whereas some countries have almost negligible number of supporting hashtags – for example, Zimbabwe.

Teams	Hashtags #
australia	12609
bahamas	559
belgium	505
botswana	227
brazil	2190
canada	28719
caribbean	807
china	1177
croatia	431
egypt	1107
ethiopia	1651
france	1224
gb	401203
germany	532
ghana	426
grenada	945
haiti	740
holland	206
hungary	827
india	1434
indonesia	2646
iran	570
ireland	17798
israel	1048
italy	1052
jamaica	21033
japan	985
kenya	11152
korea	515
latvia	582
malaysia	9077
mexico	1355
mongolia	1673
nigeria	5716
pakistan	475
philippines	2159
poland	575
puertorico	753
qatar	1077
russia	2093
serbia	303
singapore	785
somalia	324
spain	550
sweden	206
turkey	391
uganda	728
usa	204269
zimbabwe	286

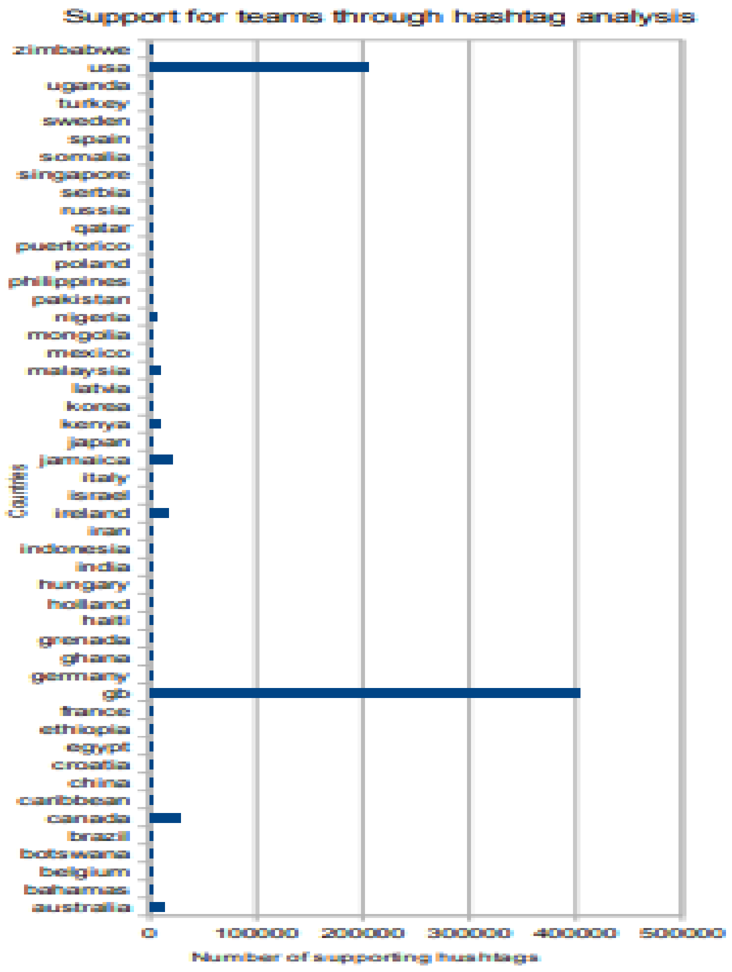


FIG 2.4

2.5 LIMITATIONS

1. In this project we have used map reduce technique to analyze provided dataset, results into a lot of lines of code and have a long compilation process.
2. We analyzed a provided dataset which has limited number of entries, we are not able to analyze live data.
3. We are not getting any valuable information from the tweet.

To overcome these limitations we are streaming live twitter data using FLUME instead of working on a limited amount of data and analyzing the dataset using PIG which reduce number of lines of code 20 times with respect to Map-reduce and to obtain some valuable information from analysis we are performing SENTIMENT ANALYSIS to obtain the emotions of people which will help a lot in advertising as well as media field.

CHAPTER 3

PROPOSED WORK

To overcome some limitations in the existing work till date, we are developing a model which:

- Connect to a live social media (twitter) data stream, extract and store this data on Hadoop. It capture the live streaming data which is generated, from different sources like twitter, weblogs and more into the HDFS cluster using Hadoop ecosystem component -Apache Flume.
- Process the data in Hadoop; restructure, filter and provide useful insights from it.
- Create tables in Hadoop and provide an interface to end users for simple querying.
- Do Sentiment Analysis on Tweets with Apache Pig Using AFINN Dictionary- It classify the tweets as positive, negative or neutral.
- Sentiment Analysis on Twitter – TimeZone wise analysis: - Sentiment Analysis will be performed on the tweets and the average of Sentiment Analysis will be measured based on the timezone of the people who tweeted them and thereby know the timezone wise views of a topic.

Components required to achieve proposed model :

1. We are streaming live data from twitter using Apache Flume.
2. We are using pig language with apache hadoop to analyse the extracted twitter data.
3. We will use Afinn dictionary to rate the sentiments of tweets.

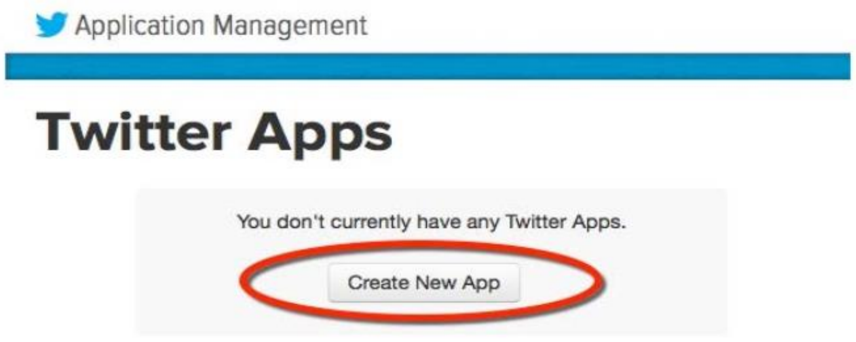
3.1 Streaming live data from twitter

To collect live data from twitter we need twitter API key, API secret, Access token, Access secret .

Step 1: Creating a Twitter Application

If you already have a Twitter account then login there or otherwise create a new twitter account on Twitter.com and login.

Next browse to https://dev.twitter.com/apps/and click the Create New App button.

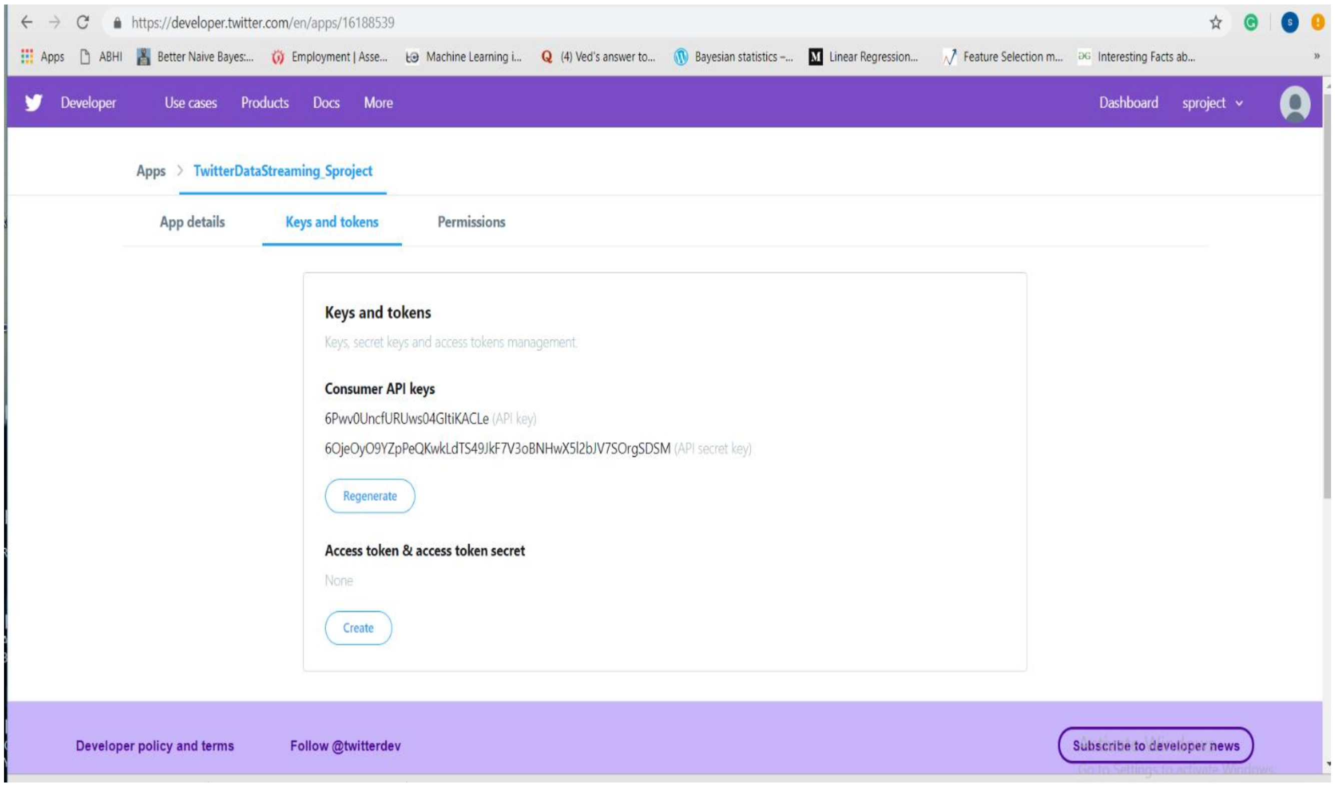


You don't currently have any Twitter Apps.

Fig 3.1

Step 2 : Fill in the basic app info form

The application "Name" must be globally unique across all Twitter apps for all users, so pick something unique. After filling the info , agree to the terms of use and press the "Create App" at the bottom of the form.You'll be redirected to the management page for your new app. Switch to theAPI Keystab, and click the "create my access token" button.



Keys and tokens

Keys, secret keys and access tokens management.

Consumer API keys

6Pwv0UncfURUws04GltiKACLe (API key)

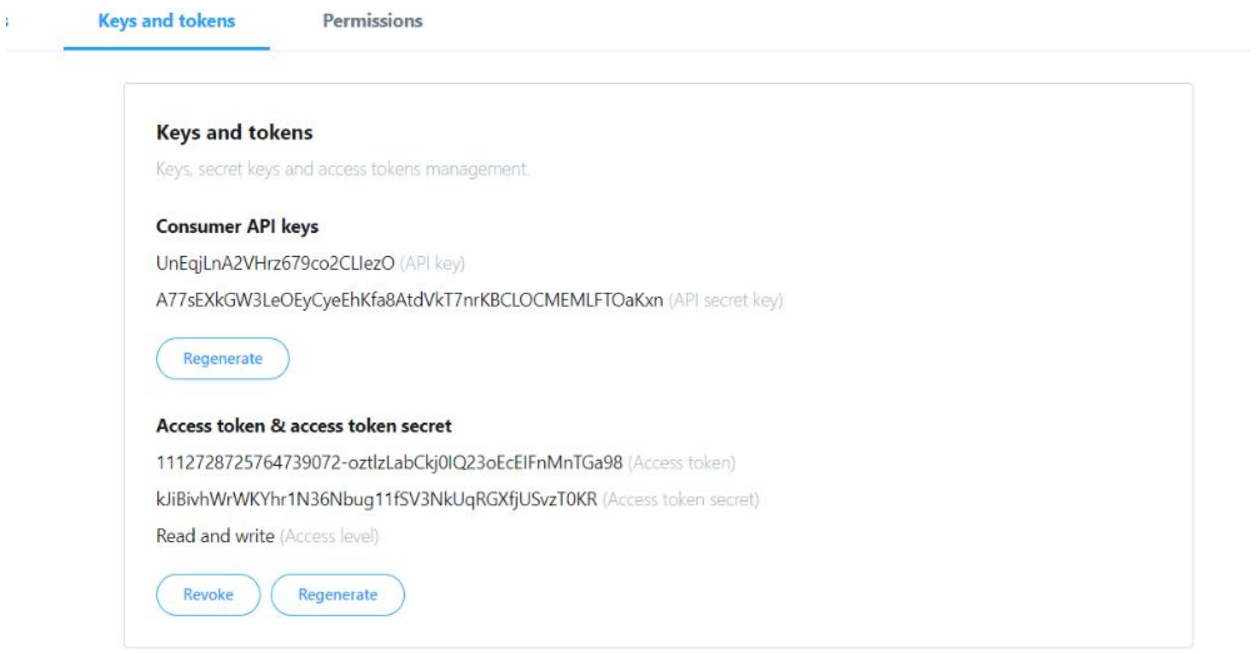
6OjeOyO9YZpPeQKwkLdTS49JkF7V3oBNHwX5l2bJV7SOrgSDSM (API secret key)

Access token & access token secret

None

FIG 3.2

Now create access token and access token secret:



Keys and tokens

Keys, secret keys and access tokens management.

Consumer API keys

UnEqjLnA2VHrz679co2CLlezO (API key)

A77sEXkGW3LeOEyCyeEhKfa8AtdVkT7nrKBCLOCMEMLFTOaKxn (API secret key)

Access token & access token secret

1112728725764739072-oztlzLabCkj0IQ23oEcEIFnMnTGa98 (Access token)

kJiBivhWrWKYhr1N36Nbug11fSV3NkUqRGXfjUSvzT0KR (Access token secret)

Read and write (Access level)

FIG 3.3

There are four pieces of information we need to copy from the Form created, this will be used when we setup the Flume agent:

•API key

•API secret

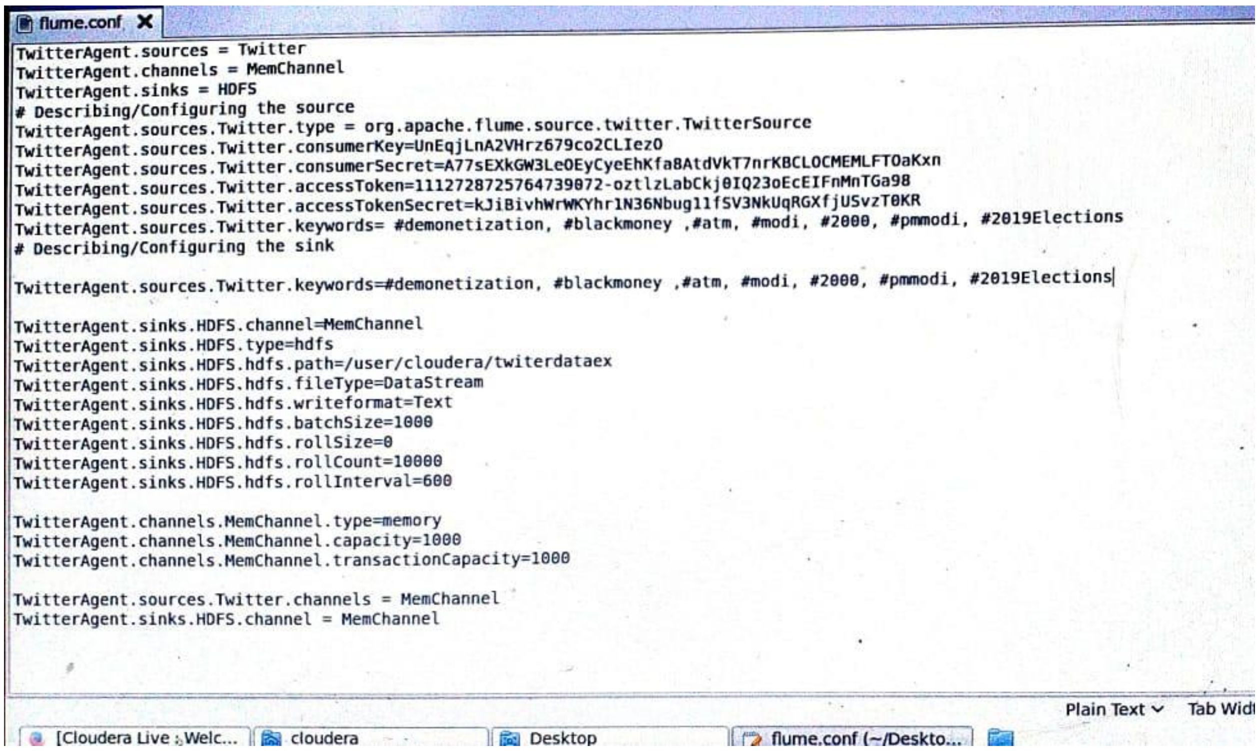
•Access token

•Access token secret

Step 3 : Install and configure Flume

Flume is a distributed, reliable, and available service for efficiently collecting, aggregating, and moving large amounts of log data. It has a simple and flexible architecture based on streaming data flows.

```
flume-ng agent -n TwitterAgent -c conf -f flume.conf
```

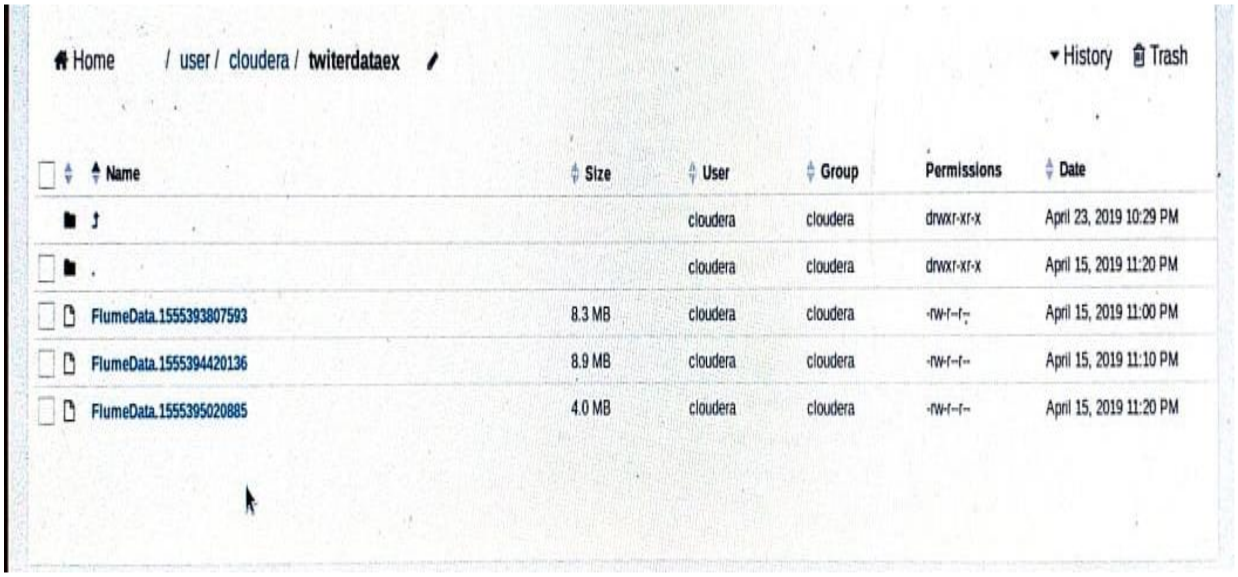


```
TwitterAgent.sources = Twitter
TwitterAgent.channels = MemChannel
TwitterAgent.sinks = HDFS
# Describing/Configuring the source
TwitterAgent.sources.Twitter.type = org.apache.flume.source.twitter.TwitterSource
TwitterAgent.sources.Twitter.consumerKey=UnEqjLnA2VHrz679co2CLIezO
TwitterAgent.sources.Twitter.consumerSecret=A77sEXkGW3LeOEyCyeEhKfa8AtdVkT7nrKBCLOCMEMLFTOaKxn
TwitterAgent.sources.Twitter.accessToken=1112728725764739072-oztlzLabCkj0IQ23oEcEIFnMnTGa98
TwitterAgent.sources.Twitter.accessTokenSecret=kJiBivhWrWKYhr1N36Nbug11fSV3NkUqRGXfjUSvzT0KR
TwitterAgent.sources.Twitter.keywords= #demonetization, #blackmoney ,#atm, #modi, #2000, #pmmodi, #2019Elections
# Describing/Configuring the sink

TwitterAgent.sources.Twitter.keywords=#demonetization, #blackmoney ,#atm, #modi, #2000, #pmmodi, #2019Elections

TwitterAgent.sinks.HDFS.channel=MemChannel
TwitterAgent.sinks.HDFS.type=hdfs
TwitterAgent.sinks.HDFS.hdfs.path=/user/cloudera/twiterdataex
TwitterAgent.sinks.HDFS.hdfs.fileType=DataStream
TwitterAgent.sinks.HDFS.hdfs.writeformat=Text
TwitterAgent.sinks.HDFS.hdfs.batchSize=1000
TwitterAgent.sinks.HDFS.hdfs.rollSize=0
TwitterAgent.sinks.HDFS.hdfs.rollCount=10000
TwitterAgent.sinks.HDFS.hdfs.rollInterval=600

TwitterAgent.channels.MemChannel.type=memory
TwitterAgent.channels.MemChannel.capacity=1000
TwitterAgent.channels.MemChannel.transactionCapacity=1000

TwitterAgent.sources.Twitter.channels = MemChannel
TwitterAgent.sinks.HDFS.channel = MemChannel
```

Fig 3.4(flume configuration)

Step 4: Run Twitter Extraction Program using Flume

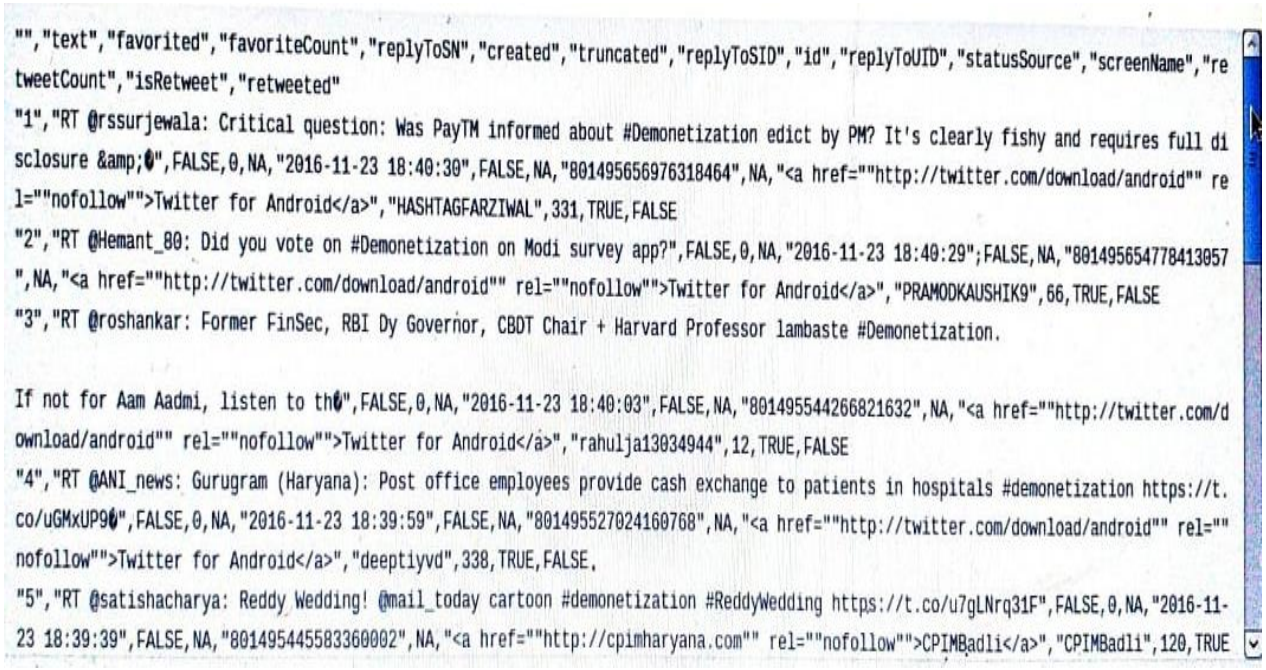
# hadoop fs –mkdir data/twiterdataex



Name	Size	User	Group	Permissions	Date
↑		cloudera	cloudera	drwxr-xr-x	April 23, 2019 10:29 PM
.		cloudera	cloudera	drwxr-xr-x	April 15, 2019 11:20 PM
FlumeData.1555393807593	8.3 MB	cloudera	cloudera	-rw-r--r--	April 15, 2019 11:00 PM
FlumeData.1555394420136	8.9 MB	cloudera	cloudera	-rw-r--r--	April 15, 2019 11:10 PM
FlumeData.1555395020885	4.0 MB	cloudera	cloudera	-rw-r--r--	April 15, 2019 11:20 PM

Fig 3.5(Flume data stored in hdfs)

Data format:



```
"","text","favorited","favoriteCount","replyToSN","created","truncated","replyToSID","id","replyToUID","statusSource","screenName","re
tweetCount","isRetweet","retweeted"
"1","RT @rssurjewala: Critical question: Was PayTM informed about #Demonetization edict by PM? It's clearly fishy and requires full di
sclosure &amp;�",FALSE,0,NA,"2016-11-23 18:40:30",FALSE,NA,"801495656976318464",NA,"<a href=""http://twitter.com/download/android"" re
l=""nofollow"">Twitter for Android</a>","HASHTAGFARZIWAL",331,TRUE,FALSE
"2","RT @Hemant_80: Did you vote on #Demonetization on Modi survey app?",FALSE,0,NA,"2016-11-23 18:40:29";FALSE,NA,"801495654778413057
",NA,"<a href=""http://twitter.com/download/android"" rel=""nofollow"">Twitter for Android</a>","PRAMODKAUSHIK9",66,TRUE,FALSE
"3","RT @roshankar: Former FinSec, RBI Dy Governor, CBDT Chair + Harvard Professor lambaste #Demonetization.

If not for Aam Aadmi, listen to th�",FALSE,0,NA,"2016-11-23 18:40:03",FALSE,NA,"801495544266821632",NA,"<a href=""http://twitter.com/d
ownload/android"" rel=""nofollow"">Twitter for Android</a>","rahulja13034944",12,TRUE,FALSE
"4","RT @ANI_news: Gurugram (Haryana): Post office employees provide cash exchange to patients in hospitals #demonetization https://t.
co/uGMxUP9�",FALSE,0,NA,"2016-11-23 18:39:59",FALSE,NA,"801495527024160768",NA,"<a href=""http://twitter.com/download/android"" rel=""
nofollow"">Twitter for Android</a>","deeptiyvd",338,TRUE,FALSE.
"5","RT @satishacharya: Reddy Wedding! @mail_today cartoon #demonetization #ReddyWedding https://t.co/u7gLNrq31F",FALSE,0,NA,"2016-11-
23 18:39:39",FALSE,NA,"801495445583360002",NA,"<a href=""http://cpimharyana.com"" rel=""nofollow"">CPIMBadli</a>","CPIMBadli",120,TRUE
```

` **Fig 3.6**

All tweets which we are going to extract will be stored in **twiterdataex** folder.

3.2 PIG

Apache Pig is an abstraction over Map-Reduce . It is a tool / platform which is used to analyze larger sets of data representing them as data flows . Pig is generally used with **Hadoop** ; we can perform all the data manipulation operations in Hadoop using Pig.

To write data analysis programs, Pig provides a high-level language known as **Pig Latin**. This language provides various operators using which programmers can develop their own functions for reading, writing, and processing data.

To analyze data using **Apache Pig**, programmers need to write scripts using Pig Latin language. All these scripts are internally converted to Map and Reduce tasks. Apache Pig has a component known as **Pig Engine** that accepts the Pig Latin scripts as input and converts those scripts into MapReduce jobs.

3.2.1 Why Do We Need Apache Pig

Programmers who are not so good at Java normally used to struggle working with Hadoop, especially while performing any MapReduce tasks. Apache Pig is a boon for all such programmers.

- Using **Pig Latin**, programmers can perform MapReduce tasks easily without having to type complex codes in Java.
- Apache Pig uses **multi-query approach**, thereby reducing the length of codes. For example, an operation that would require you to type 200 lines of code (LoC) in Java can be easily done by typing as less as just 10 LoC in Apache Pig. Ultimately Apache Pig reduces the development time by almost 16 times.
- Pig Latin is **SQL-like language** and it is easy to learn Apache Pig when you are familiar with SQL.
- Apache Pig provides many built-in operators to support data operations like joins, filters, ordering, etc. In addition, it also provides nested data types like tuples, bags, and maps that are missing from MapReduce.

The language used to analyze data in Hadoop using Pig is known as Pig Latin. It is a highlevel data processing language which provides a rich set of data types and operators to perform various operations on the data.

To perform a particular task Programmers using Pig, programmers need to write a Pig script using the Pig Latin language, and execute them using any of the execution mechanisms (Grunt Shell, UDFs, Embedded). After execution, these scripts will go through a series of transformations applied by the Pig Framework, to produce the desired output.

Internally, Apache Pig converts these scripts into a series of MapReduce jobs, and thus, it makes the programmer's job easy. The architecture of Apache Pig is shown below.

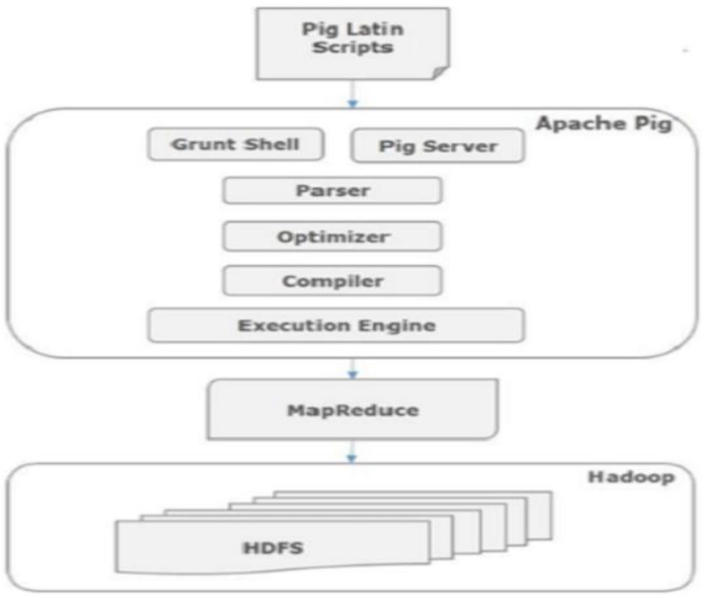


FIG 3.7

3.2.2 Apache Pig Components:

As shown in the figure, there are various components in the Apache Pig framework. Let us take a look at the major components.

- **Parser:** Initially the Pig Scripts are handled by the Parser. It checks the syntax of the script, does type checking, and other miscellaneous checks. The output of the parser will be a DAG (directed acyclic graph), which represents the Pig Latin statements and logical operators.In the DAG, the logical operators of the script are represented as the nodes and the data flows are represented as edges.
- **Optimizer:** The logical plan (DAG) is passed to the logical optimizer, which carries out the logical optimizations such as projection and pushdown.
- **Compiler:** The compiler compiles the optimized logical plan into a series of MapReduce jobs.
- **Execution engine:** Finally the MapReduce jobs are submitted to Hadoop in a sorted order. Finally, these MapReduce jobs are executed on Hadoop producing the desired results.

3.2.3 Apache Pig Execution Mechanisms

Apache Pig scripts can be executed in three ways, namely, interactive mode, batch mode, and embedded mode.

- **Interactive Mode (Grunt shell)** − You can run Apache Pig in interactive mode using the Grunt shell. In this shell, you can enter the Pig Latin statements and get the output (using Dump operator).
- **Batch Mode (Script)** − You can run Apache Pig in Batch mode by writing the Pig Latin script in a single file with .pig extension**.**
- **Embedded Mode (UDF)** − Apache Pig provides the provision of defining our own functions (User Defined Functions) in programming languages such as Java, and using them in our script.

3.2.4 Invoking the Grunt Shell

You can invoke the Grunt shell in a desired mode (local/MapReduce) using the - the **−x** option as shown below.

Local mode	MapReduce mode
Command − $ ./pig –x local	**Command** −$ ./pig -x mapreduce

Output –

```
15/09/28 10:13:03 INFO pig.Main:
Logging error messages to:
/home/Hadoop/pig_1443415383991.log
2015-09-28 10:13:04,838 [main]
INFO
org.apache.pig.backend.hadoop.execution
engine.HExecutionEngine - Connecting to
hadoop file system at: file:///

grunt>
```

Output –

```
15/09/28 10:28:46 INFO pig.Main:
Logging error messages to:
/home/Hadoop/pig_1443416326123.log
2015-09-28 10:28:46,427 [main] INFO
org.apache.pig.backend.hadoop.execution
engine.HExecutionEngine - Connecting to
hadoop file system at: file:///

grunt>
```

Either of these commands gives you the Grunt shell prompt as shown below.

- grunt >

3.2.5 Apache Pig instead of MapReduce. Why?

Apache Pig	**MapReduce**
Apache Pig is a data flow language.	MapReduce is a data processing paradigm.
It is a high level language.	MapReduce is low level and rigid.
Performing a Join operation in Apache Pig is pretty simple.	It is quite difficult in MapReduce to perform a Join operation between datasets.
Any novice programmer with a basic knowledge of SQL can work conveniently with Apache Pig.	Exposure to Java is must to work with MapReduce.
Apache Pig uses multi-query approach, thereby reducing the length of the codes to a great extent.	MapReduce will require almost 20 times more the number of lines to perform the same task.
There is no need for compilation. On execution, every Apache Pig operator is converted internally into a MapReduce job.	MapReduce jobs have a long compilation process.

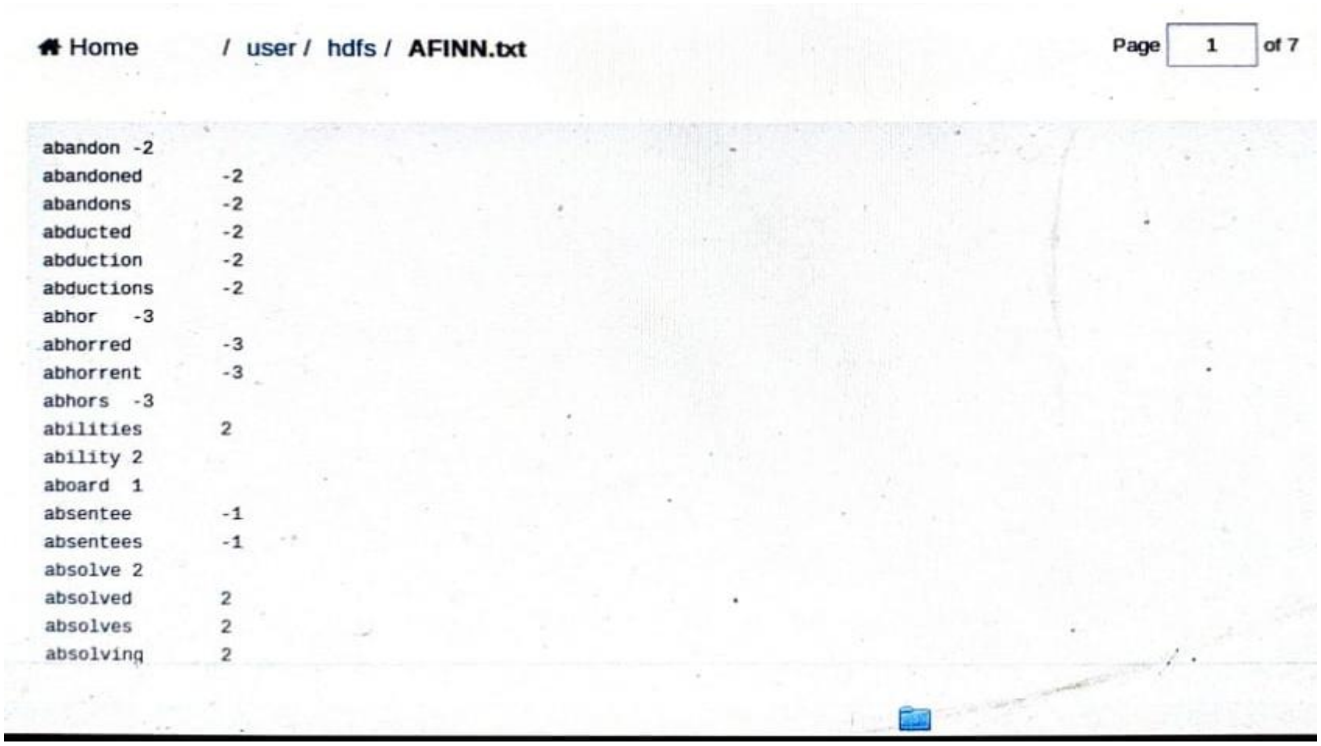
3.2.6 Access the Refined Sentiment Data with Excel

For performing Sentiment Analysis, we need the tweet_id and tweet_text, so we will create a csv file that will extract the id and tweet_text from the tweets using the JsonLoader**.**We need to load the tweets using JsonLoader which supports maps, so we are using **elephant bird JsonLoader** to load the tweets.

- The required dataset will be stored in demonetization.csv file.

Now we will use a dictionary called AFINN to calculate the sentiments. AFINN is a dictionary which consists of 2500 words rated from +5 to -5 depending on their meaning.

AFINN.txt has all the Positive, Negative and Neutral sentiment words list.



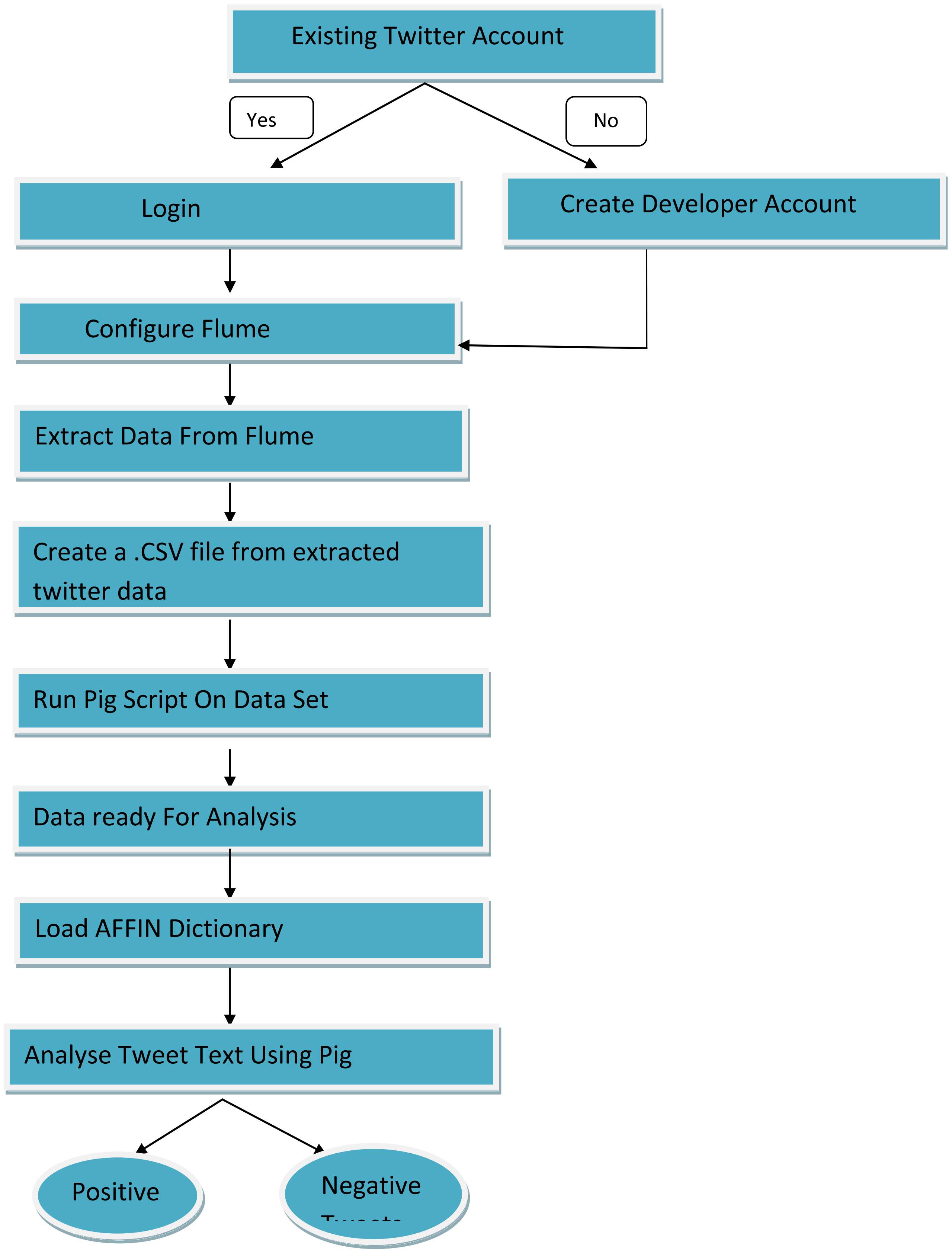
```
abandon -2
abandoned       -2
abandons        -2
abducted        -2
abduction       -2
abductions      -2
abhor   -3
abhorred        -3
abhorrent       -3
abhors  -3
abilities       2
ability 2
aboard  1
absentee        -1
absentees       -1
absolve 2
absolved        2
absolves        2
absolving       2
```

Fig 3.8

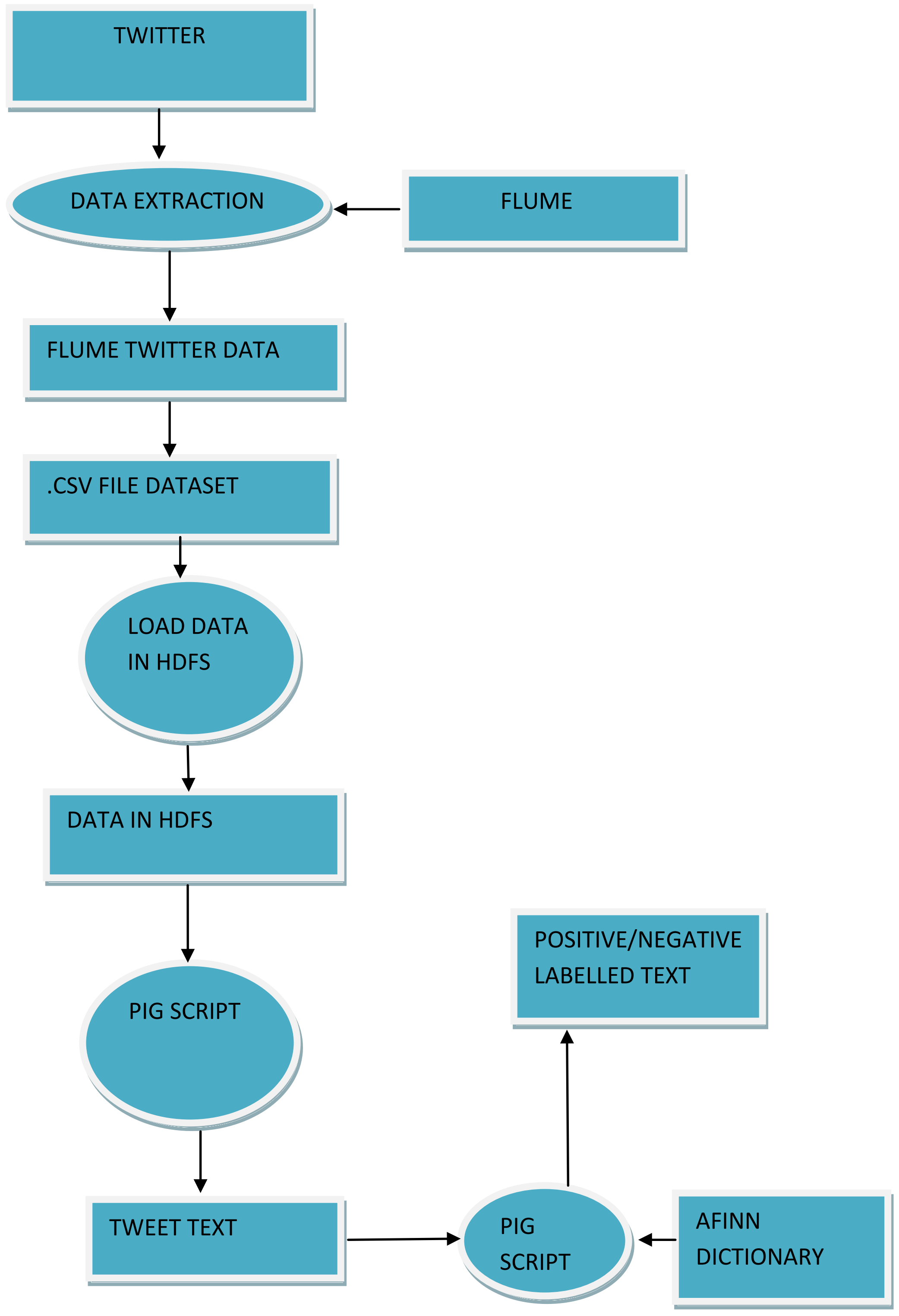
# hadoop fs –put /home/cloudera/Desktop/twitter/AFINN.txt /user/hdfs

- Perform sentiment analysis using pig with the help of AFINN dictionary

3.2.6 Flowchart



3.2.7 DFD (DATA FLOW DIAGRAM)



Chapter 4

Result and Discussion

4.1 Implementations of proposed work

Step 1:Create an HDFS directory & copy the input file“demonetization-tweets.csv”in that location

- hadoop fs -mkdir /user/hdfs

- hadoop fs –put /home/cloudera/Desktop/twitter/demonetization-tweets.csv /user/hdfs

Step 2: Analyze the tweet data from the file. The metadata of the file and a sample tweet is as given below.

<u>Metadata</u>

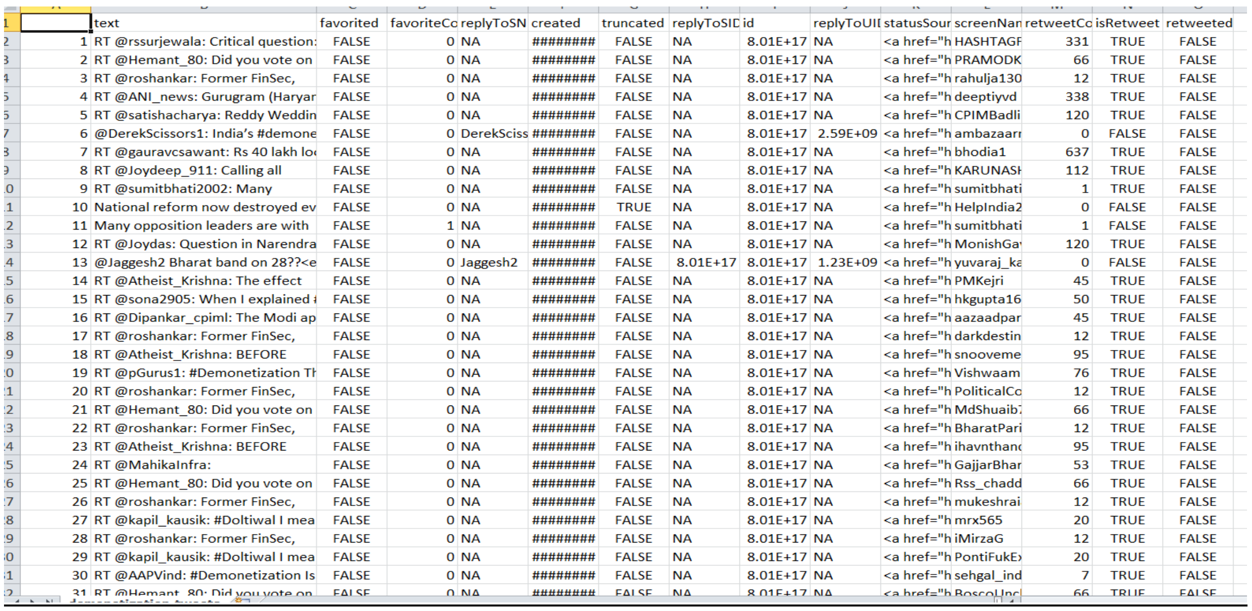
id

Text (Tweets)
favorited
favoriteCount
replyToSN
created
truncated
replyToSID
id
replyToUID
statusSource
screenName
retweetCount
isRetweet
retweeted

<u>Sample Tweet</u>

("1","RT @rssurjewala: Critical question: Was PayTM informed about #Demonetization edict by PM? It's clearly fishy and requires full disclosure &amp;�",FALSE,0,NA,"2016-11-23 18:40:30",FALSE,NA,"801495656976318464",NA,"<a href=""http://twitter.com/download/android"" rel=""nofollow"">Twitter for Android</a>","HASHTAGFARZIWAL",331,TRUE,FALSE)

Dataset in excel format:



		text	favorited	favoriteCo	replyToSN	created	truncated	replyToSID	id	replyToUID	statusSour	screenNan	retweetCo	isRetweet	retweeted
2	1	RT @rssurjewala: Critical question:	FALSE	0	NA	########	FALSE	NA	8.01E+17	NA	<a href="h	HASHTAGF	331	TRUE	FALSE
3	2	RT @Hemant_80: Did you vote on	FALSE	0	NA	########	FALSE	NA	8.01E+17	NA	<a href="h	PRAMODK	66	TRUE	FALSE
4	3	RT @roshankar: Former FinSec,	FALSE	0	NA	########	FALSE	NA	8.01E+17	NA	<a href="h	rahulja130	12	TRUE	FALSE
5	4	RT @ANI_news: Gurugram (Haryar	FALSE	0	NA	########	FALSE	NA	8.01E+17	NA	<a href="h	deeptiyvd	338	TRUE	FALSE
6	5	RT @satishacharya: Reddy Weddin	FALSE	0	NA	########	FALSE	NA	8.01E+17	NA	<a href="h	CPIMBadli	120	TRUE	FALSE
7	6	@DerekScissors1: India's #demone	FALSE	0	DerekSciss	########	FALSE	NA	8.01E+17	2.59E+09	<a href="h	ambazaarı	0	FALSE	FALSE
8	7	RT @gauravcsawant: Rs 40 lakh lo	FALSE	0	NA	########	FALSE	NA	8.01E+17	NA	<a href="h	bhodia1	637	TRUE	FALSE
9	8	RT @Joydeep_911: Calling all	FALSE	0	NA	########	FALSE	NA	8.01E+17	NA	<a href="h	KARUNASH	112	TRUE	FALSE
10	9	RT @sumitbhati2002: Many	FALSE	0	NA	########	FALSE	NA	8.01E+17	NA	<a href="h	sumitbhati	1	TRUE	FALSE
11	10	National reform now destroyed ev	FALSE	0	NA	########	TRUE	NA	8.01E+17	NA	<a href="h	HelpIndia2	0	FALSE	FALSE
12	11	Many opposition leaders are with	FALSE	1	NA	########	FALSE	NA	8.01E+17	NA	<a href="h	sumitbhati	1	FALSE	FALSE
13	12	RT @Joydas: Question in Narendra	FALSE	0	NA	########	FALSE	NA	8.01E+17	NA	<a href="h	MonishGa	120	TRUE	FALSE
14	13	@Jaggesh2 Bharat band on 28??<e	FALSE	0	Jaggesh2	########	FALSE	8.01E+17	8.01E+17	1.23E+09	<a href="h	yuvaraj_ka	0	FALSE	FALSE
15	14	RT @Atheist_Krishna: The effect	FALSE	0	NA	########	FALSE	NA	8.01E+17	NA	<a href="h	PMKejri	45	TRUE	FALSE
16	15	RT @sona2905: When I explained	FALSE	0	NA	########	FALSE	NA	8.01E+17	NA	<a href="h	hkgupta16	50	TRUE	FALSE
17	16	RT @Dipankar_cpiml: The Modi ap	FALSE	0	NA	########	FALSE	NA	8.01E+17	NA	<a href="h	aazaadpar	45	TRUE	FALSE
18	17	RT @roshankar: Former FinSec,	FALSE	0	NA	########	FALSE	NA	8.01E+17	NA	<a href="h	darkdestin	12	TRUE	FALSE
19	18	RT @Atheist_Krishna: BEFORE	FALSE	0	NA	########	FALSE	NA	8.01E+17	NA	<a href="h	snooveme	95	TRUE	FALSE
20	19	RT @pGurus1: #Demonetization Th	FALSE	0	NA	########	FALSE	NA	8.01E+17	NA	<a href="h	Vishwaam	76	TRUE	FALSE
21	20	RT @roshankar: Former FinSec,	FALSE	0	NA	########	FALSE	NA	8.01E+17	NA	<a href="h	PoliticalCo	12	TRUE	FALSE
22	21	RT @Hemant_80: Did you vote on	FALSE	0	NA	########	FALSE	NA	8.01E+17	NA	<a href="h	MdShuaib7	66	TRUE	FALSE
23	22	RT @roshankar: Former FinSec,	FALSE	0	NA	########	FALSE	NA	8.01E+17	NA	<a href="h	BharatPari	12	TRUE	FALSE
24	23	RT @Atheist_Krishna: BEFORE	FALSE	0	NA	########	FALSE	NA	8.01E+17	NA	<a href="h	ihavnthanc	95	TRUE	FALSE
25	24	RT @MahikaInfra:	FALSE	0	NA	########	FALSE	NA	8.01E+17	NA	<a href="h	GajjarBhar	53	TRUE	FALSE
26	25	RT @Hemant_80: Did you vote on	FALSE	0	NA	########	FALSE	NA	8.01E+17	NA	<a href="h	Rss_chadd	66	TRUE	FALSE
27	26	RT @roshankar: Former FinSec,	FALSE	0	NA	########	FALSE	NA	8.01E+17	NA	<a href="h	mukeshrai	12	TRUE	FALSE
28	27	RT @kapil_kausik: #Doltiwal I mea	FALSE	0	NA	########	FALSE	NA	8.01E+17	NA	<a href="h	mrx565	20	TRUE	FALSE
29	28	RT @roshankar: Former FinSec,	FALSE	0	NA	########	FALSE	NA	8.01E+17	NA	<a href="h	iMirzaG	12	TRUE	FALSE
30	29	RT @kapil_kausik: #Doltiwal I mea	FALSE	0	NA	########	FALSE	NA	8.01E+17	NA	<a href="h	PontiFukE	20	TRUE	FALSE
31	30	RT @AAPVind: #Demonetization Is	FALSE	0	NA	########	FALSE	NA	8.01E+17	NA	<a href="h	sehgal_ind	7	TRUE	FALSE
32	31	RT @Hemant_80: Did you vote on	FALSE	0	NA	########	FALSE	NA	8.01E+17	NA	<a href="h	BoscoUnc	66	TRUE	FALSE

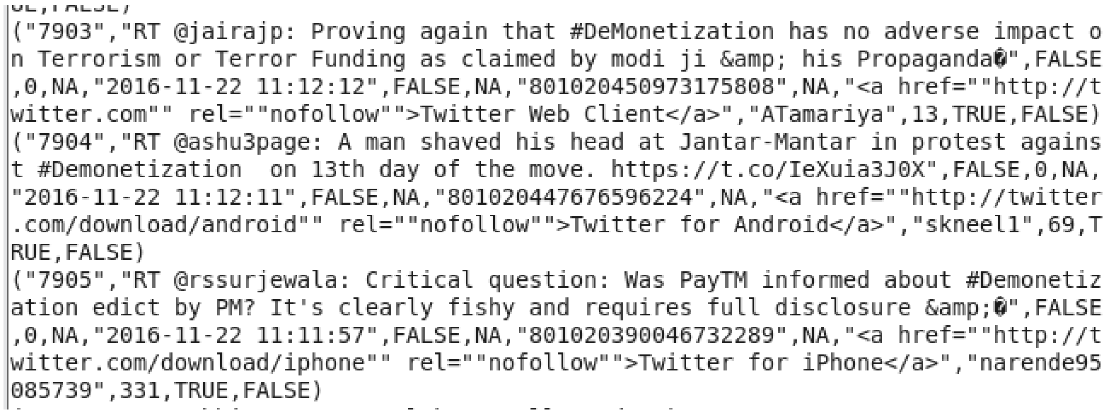
Fig 4.1

Step 3: First load the data into pig using PigStorage

- ➢ loadtweets = LOAD '/user/hdfs/demonetization-tweets.csv' USING PigStorage(',');

Now after loading successfully,we can see the tweets loaded successfully into pig by using the DUMP command.

- ➢ DUMP loadtweets;



```
UE,FALSE)
("7903","RT @jairajp: Proving again that #DeMonetization has no adverse impact o
n Terrorism or Terror Funding as claimed by modi ji &amp; his Propaganda�",FALSE
,0,NA,"2016-11-22 11:12:12",FALSE,NA,"801020450973175808",NA,"<a href=""http://t
witter.com"" rel=""nofollow"">Twitter Web Client</a>","ATamariya",13,TRUE,FALSE)
("7904","RT @ashu3page: A man shaved his head at Jantar-Mantar in protest agains
t #Demonetization  on 13th day of the move. https://t.co/IeXuia3J0X",FALSE,0,NA,
"2016-11-22 11:12:11",FALSE,NA,"801020447676596224",NA,"<a href=""http://twitter
.com/download/android"" rel=""nofollow"">Twitter for Android</a>","skneel1",69,T
RUE,FALSE)
("7905","RT @rssurjewala: Critical question: Was PayTM informed about #Demonetiz
ation edict by PM? It's clearly fishy and requires full disclosure &amp;�",FALSE
,0,NA,"2016-11-22 11:11:57",FALSE,NA,"801020390046732289",NA,"<a href=""http://t
witter.com/download/iphone"" rel=""nofollow"">Twitter for iPhone</a>","narende95
085739",331,TRUE,FALSE)
```

Our intension is to extract the id and the tweet_text as follows

- ➢ extract_details = FOREACH loadtweets GENERATE $0 as id,$1 as text;

Step 4: DUMP the “extract_details”, and the following sample result will be produced.

Output:

```
("1","RT @rssurjewala: Critical question: Was PayTM informed about #Demonetization edict by PM? It's clearly fishy and requires full disclosure &amp;�")
```

Step 5: divide the tweet_text into words to calculate the sentiment of the whole tweet.

- ➢ tokens = foreach extract_details generate id,text, FLATTEN(TOKENIZE(text)) As word;

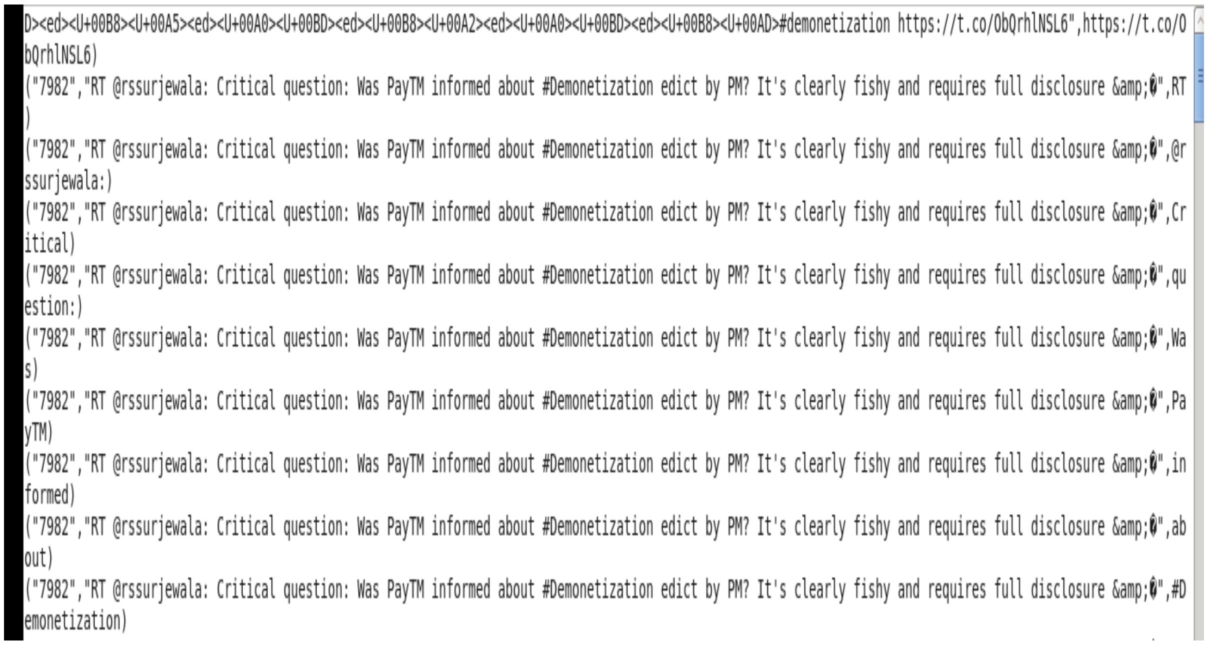
For every word in the tweet_text, each word will be taken and created as a new row. We can use the DUMP command to check the same.

For example, In the below record, we can see that at the last RT word has been taken and created a new record for that.

- ➢ DUMP tokens;

Output:

```
("1","RT @rssurjewala: Critical question: Was PayTM informed about #Demonetization edict by PM? It's clearly fishy and requires full disclosure &amp;�",RT)
```



```
D><ed><U+00B8><U+00A5><ed><U+00A0><U+00BD><ed><U+00B8><U+00A2><ed><U+00A0><U+00BD><ed><U+00B8><U+00AD>#demonetization https://t.co/ObQrhlNSL6",https://t.co/O
bQrhlNSL6)
("7982","RT @rssurjewala: Critical question: Was PayTM informed about #Demonetization edict by PM? It's clearly fishy and requires full disclosure &amp;�",RT
)
("7982","RT @rssurjewala: Critical question: Was PayTM informed about #Demonetization edict by PM? It's clearly fishy and requires full disclosure &amp;�",@r
ssurjewala:)
("7982","RT @rssurjewala: Critical question: Was PayTM informed about #Demonetization edict by PM? It's clearly fishy and requires full disclosure &amp;�",Cr
itical)
("7982","RT @rssurjewala: Critical question: Was PayTM informed about #Demonetization edict by PM? It's clearly fishy and requires full disclosure &amp;�",qu
estion:)
("7982","RT @rssurjewala: Critical question: Was PayTM informed about #Demonetization edict by PM? It's clearly fishy and requires full disclosure &amp;�",Wa
s)
("7982","RT @rssurjewala: Critical question: Was PayTM informed about #Demonetization edict by PM? It's clearly fishy and requires full disclosure &amp;�",Pa
yTM)
("7982","RT @rssurjewala: Critical question: Was PayTM informed about #Demonetization edict by PM? It's clearly fishy and requires full disclosure &amp;�",in
formed)
("7982","RT @rssurjewala: Critical question: Was PayTM informed about #Demonetization edict by PM? It's clearly fishy and requires full disclosure &amp;�",ab
out)
("7982","RT @rssurjewala: Critical question: Was PayTM informed about #Demonetization edict by PM? It's clearly fishy and requires full disclosure &amp;�",#D
emonetization)
```

Fig 4.2

Step 6:Checking the schema: We use the describe tokens command to check the schema of that relation and is as follows:

- ➢ describe tokens;

Output:

```
tokens: {id: bytearray,text: bytearray,word: chararray}
```

4.2 Analysis of Sentiments

Now, we have to analyse the Sentiment for the tweet by using the words in the text. We will rate the word as per its meaning from +5 to -5 using the dictionary AFINN. The AFINN is a dictionary which consists of 2500 words which are rated from +5 to -5 depending on their meaning.

Step 1: Copy the "AFFIN.txt" dictionary to the HDFS directory that we have created & then load the dictionary using the below statement.

```
➢ hadoop fs –put /home/cloudera/Desktop/twitter/AFINN.txt /user/hdfs
➢ dictionary =LOAD '/user/hdfs/AFINN.txt' using PigStorage('\t')
  AS(word:chararray,rating:int);
```

Step 2: We perform a map side join by joining the tokens statement and the dictionary contents using this relation:

```
➢ word_rating = join tokens by word left outer, dictionary by word using 'replicated';
```

Step 3: Check the schema of the statement after performing join operation by using the below command:

- ➢ describe word_rating;

Output:

```
word_rating: {tokens::id: bytearray,tokens::text: bytearray,tokens::word:
chararray,dictionary::word: chararray,dictionary::rating: int}
```

In the above statement describe word_rating , we can see that the word_rating has joined the tokens (consists of id, tweet text, word) statement and the dictionary(consists of word, rating).

Step 4: Extract the id, tweet text and word rating(from the dictionary) by using the below relation.

- rating = foreach word_rating generate tokens::id as id,tokens::text as text, dictionary::rating as rate;

Step 5:Check the schema of the relation rating by using the below command

- describe rating;

Output:

```
rating: {id: bytearray,text: bytearray,rate: int}
```

In the above statement describe rating we can see that our relation now consists of id , tweet text and rate(for each word).

Step 6: Group the rating of all the words in a tweet by using the below relation:

- word_group = group rating by (id,text);

It is grouped by two constraints, id and tweet text.

Step 7: Perform the Average operation on the rating of the words per each tweet.

- avg_rate = foreach word_group generate group, AVG(rating.rate) as tweet_rating;

The Average rating of the tweet using the rating of each word have been calculated.

From the above relation, we will get all the tweets i.e., both positive and negative.

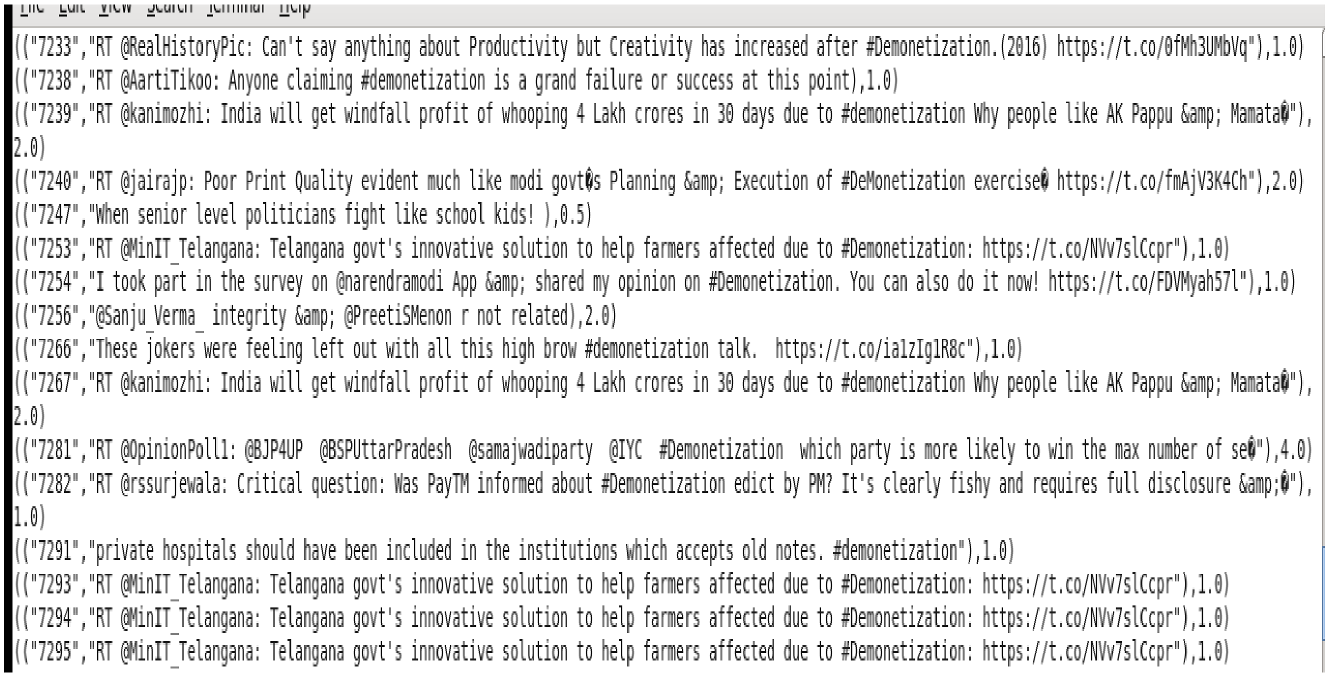
Step 8: The positive tweets can be classified by taking the rating of the tweet which can be from 0-5. Whereas, the negative tweets can be classified by taking the rating of the tweet from -5 to -1.

The Sentiment Analysis on Twitter data using Pig have been performed successfully.

Step 9: Operation to filter out the positive tweets

The positive tweets are filtered using the below statement:

- ➢ positive_tweets = filter avg_rate by tweet_rating>=0;

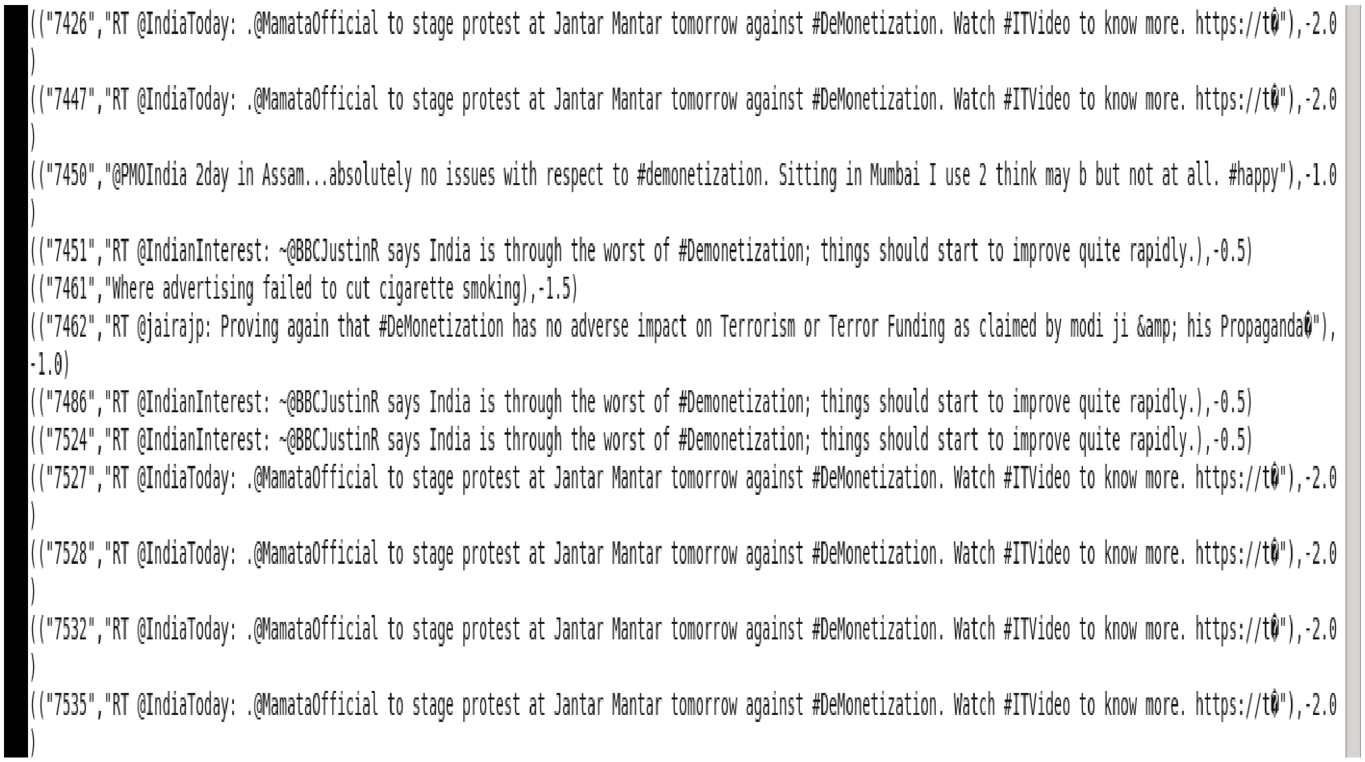


```
(("7233","RT @RealHistoryPic: Can't say anything about Productivity but Creativity has increased after #Demonetization.(2016) https://t.co/0fMh3UMbVq"),1.0)
(("7238","RT @AartiTikoo: Anyone claiming #demonetization is a grand failure or success at this point),1.0)
(("7239","RT @kanimozhi: India will get windfall profit of whooping 4 Lakh crores in 30 days due to #demonetization Why people like AK Pappu &amp; Mamata�"),
2.0)
(("7240","RT @jairajp: Poor Print Quality evident much like modi govt�s Planning &amp; Execution of #DeMonetization exercise� https://t.co/fmAjV3K4Ch"),2.0)
(("7247","When senior level politicians fight like school kids! ),0.5)
(("7253","RT @MinIT_Telangana: Telangana govt's innovative solution to help farmers affected due to #Demonetization: https://t.co/NVv7slCcpr"),1.0)
(("7254","I took part in the survey on @narendramodi App &amp; shared my opinion on #Demonetization. You can also do it now! https://t.co/FDVMyah57l"),1.0)
(("7256","@Sanju_Verma_ integrity &amp; @PreetiSMenon r not related),2.0)
(("7266","These jokers were feeling left out with all this high brow #demonetization talk.  https://t.co/ia1zIg1R8c"),1.0)
(("7267","RT @kanimozhi: India will get windfall profit of whooping 4 Lakh crores in 30 days due to #demonetization Why people like AK Pappu &amp; Mamata�"),
2.0)
(("7281","RT @OpinionPoll1: @BJP4UP  @BSPUttarPradesh  @samajwadiparty  @IYC  #Demonetization  which party is more likely to win the max number of se�"),4.0)
(("7282","RT @rssurjewala: Critical question: Was PayTM informed about #Demonetization edict by PM? It's clearly fishy and requires full disclosure &amp;�"),
1.0)
(("7291","private hospitals should have been included in the institutions which accepts old notes. #demonetization"),1.0)
(("7293","RT @MinIT_Telangana: Telangana govt's innovative solution to help farmers affected due to #Demonetization: https://t.co/NVv7slCcpr"),1.0)
(("7294","RT @MinIT_Telangana: Telangana govt's innovative solution to help farmers affected due to #Demonetization: https://t.co/NVv7slCcpr"),1.0)
(("7295","RT @MinIT_Telangana: Telangana govt's innovative solution to help farmers affected due to #Demonetization: https://t.co/NVv7slCcpr"),1.0)
```

Fig 4.3 Positive Tweets

The negative tweets are filtered as follows:

- negative_tweets = filter avg_rate by tweet_rating<0;



```
(("7426","RT @IndiaToday: .@MamataOfficial to stage protest at Jantar Mantar tomorrow against #DeMonetization. Watch #ITVideo to know more. https://t�"),-2.0
)
(("7447","RT @IndiaToday: .@MamataOfficial to stage protest at Jantar Mantar tomorrow against #DeMonetization. Watch #ITVideo to know more. https://t�"),-2.0
)
(("7450","@PMOIndia 2day in Assam...absolutely no issues with respect to #demonetization. Sitting in Mumbai I use 2 think may b but not at all. #happy"),-1.0
)
(("7451","RT @IndianInterest: ~@BBCJustinR says India is through the worst of #Demonetization; things should start to improve quite rapidly.),-0.5)
(("7461","Where advertising failed to cut cigarette smoking),-1.5)
(("7462","RT @jairajp: Proving again that #DeMonetization has no adverse impact on Terrorism or Terror Funding as claimed by modi ji &amp; his Propaganda�"),
-1.0)
(("7486","RT @IndianInterest: ~@BBCJustinR says India is through the worst of #Demonetization; things should start to improve quite rapidly.),-0.5)
(("7524","RT @IndianInterest: ~@BBCJustinR says India is through the worst of #Demonetization; things should start to improve quite rapidly.),-0.5)
(("7527","RT @IndiaToday: .@MamataOfficial to stage protest at Jantar Mantar tomorrow against #DeMonetization. Watch #ITVideo to know more. https://t�"),-2.0
)
(("7528","RT @IndiaToday: .@MamataOfficial to stage protest at Jantar Mantar tomorrow against #DeMonetization. Watch #ITVideo to know more. https://t�"),-2.0
)
(("7532","RT @IndiaToday: .@MamataOfficial to stage protest at Jantar Mantar tomorrow against #DeMonetization. Watch #ITVideo to know more. https://t�"),-2.0
)
(("7535","RT @IndiaToday: .@MamataOfficial to stage protest at Jantar Mantar tomorrow against #DeMonetization. Watch #ITVideo to know more. https://t�"),-2.0
)
```

Fig 4.4 Negative Tweets

CHAPTER 5

CONCLUSION

In this work, a real-time model was designed to identify the sentiment polarity(positive/negative) in twitter using Afinn dictionary. Twitter is a large source of data, which make it more attractive to perform sentiment analysis. Here, we performed analysis on tweets extracted by flume which collect and aggregate data and store them in hdfs. The patterns of public opinions can be understood and a review can be given based on people's opinion.

The limitation of this project is that processing time of performing analysis on data live is minimum 5 minutes, here we are first collecting the tweets in HDFS then after some time when data is fully fetched only then we can perform analysis. Therefore, for performing live data analysis we have to use spark which is a real time data processing engine. It is used to process the streaming data in real time. Moreover,the ram capacity of minimum 16 GB is required to for spark which is not easily available.

The future scope of the project lies in the aspect that the sentiment analysis module will extract real time data which will serve as an important model in various fields such as security field, entertainment field as well as advertisement field. In security field this model can help reduce online harassement and trolls. In entertainment field, we can anlayse the behavior of a group of person and recommend movies, book, music etc according to our real time analysis. Sentiment analysis can prove fruitful to the Advertising brands as well. They can see the response of public about a specific product through their tweets and hashtags and develop their marketing strategies based on these analysis. In future we can also build an emotionally aware chatbot with the use of Natural Language Processing(NLP). Undoubtedly, Sentiment Analysis will benefit our Digital as well as non-digital society at large.

CHAPTER 6

REFERENCES

1. **https://github.com/Zhanelya/Big-Data-Project**
2. **https://en.wikipedia.org/wiki/Sentiment_analysis**
3. **https://ieeexplore.ieee.org/document/6812968?tp=&arnumber=6812968**
4. **https://www.tutorialspoint.com/hadoop/hadoop_mapreduce.htm**
5. **http://darenr.github.io/afinn/**
6. **https://en.wikipedia.org/wiki/Natural_language_processing**