Sentiment Analysis of Twitter by using Apache Flume

Y.Sushmitha Reddy[1], **M.Padma**[2]

[1] Computer Science and Engineering Dept, GPREC,
Kurnool (District), Andhra Pradesh-518007, INDIA. reddy.sushmitha36@gmail.com

[2]Asst.Prof. at Computer Science and Engineering Dept, GPREC,
Kurnool (District), Andhra Pradesh-518007, INDIA. padma.gprec@gmail.com

Abstract

Now-a-days social media is playing a key role in all aspects of life. Identifying and classifying views expressed in source text are organized by twitter analysis. Organizations use twitter analysis to understand how the public feels about something at a particular moment in time, and also to track how those opinions change over time. In the form of tweets, status updates and blog posts a huge volume of twitter rich data is being created in social media. User generated data is very useful in knowing the attitude of the crowd. The presence of shortcut words and misspellings makes twitter analysis a bit difficult when compared to general sentiment analysis.

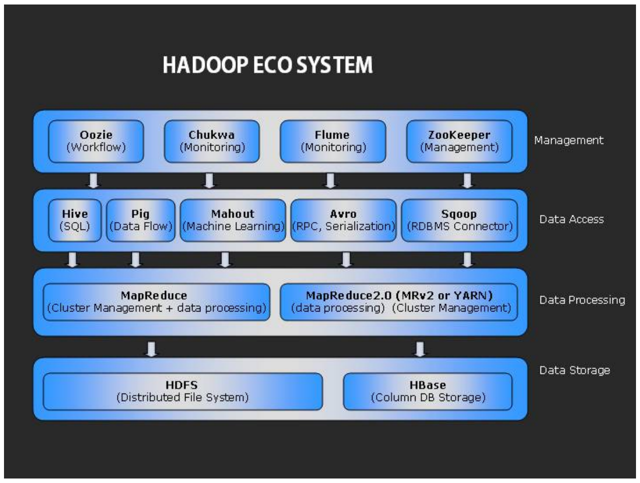
The two strategies used for analyzing twitter application from the text are Machine learning approach and Knowledge base approach. By using log analysis we are able to obtain opinions of a particular group or a particular person. Log files from web servers represent a treasure of data that can be used to gain a deep understanding of anything. Here in this paper, we analyze about any type of tweets in twitter on basis of timestamp (like seconds, minutes, hours, days etc.). Standford-corenlp (natural language processing) is used for classifying the tweets in providing twitter application by using Apache flume in Hadoop

Keywords:*Analysis, Hadoop,BIGDATA, Comment, Flume, Hive, HQL, Sentiment Analysis, Structured, Semi-Structured, Twitter, Tweets, Un-Structured.*

1. Introduction

Hadoop is a project from the Apache Software Foundation written in Java. It enables management of petabytes of data in thousands of machines. The motivation comes from Google's MapReduce and Google File System identification. Hadoop's biggest contributor has been the search giant Yahoo [1]. In recent years Hadoop has become a widely used platform and runtime surroundings for the deployment of Big Data applications.

Fig. 1: Describes clearly Apache Hadoop Ecosystem.


HADOOP ECO SYSTEM
Oozie (Workflow)
Chukwa (Monitoring)
Flume (Monitoring)
ZooKeeper (Management)
Management
Hive (SQL)
Pig (Data Flow)
Mahout (Machine Learning)
Avro (RPC, Serialization)
Sqoop (RDBMS Connector)
Data Access
MapReduce (Cluster Management + data processing)
MapReduce2.0 (MRv2 or YARN) (data processing) (Cluster Management)
Data Processing
HDFS (Distributed File System)
HBase (Column DB Storage)
Data Storage

Big data is a collection of large data sets that include different types such as structured, unstructured and semistructured data. This data can be generated from different sources like social media (Facebook, Twitter, YouTube etc.), audios, images, log files, sensor data, stock market exchanges, transactional applications, sensor web etc. are the main contributors of these data [2]. These large data sets are called Big Data. In terms of relational databases, moving and modifying large volumes of unstructured data into the necessary form for Extraction, Transformation, Loading (ETL) can be both costly and timeconsuming. To process or analyze this huge amount of data or extracting meaningful information is a challenging task now a days. Big data exceeds the processing capability of traditional database to capture, manage, and process the voluminous amount of data. Falling storage costs and access to better compute power for less money have made the software more attractive as datasets continue to grow, As a result, many companies are rethinking their move toward to traditional enterprise storage and architecture to leverage big data [3]. Hadoop is best suited for Processing unstructured data, Complex parallel information processing, Large Data Sets/Files, Machine Learning serious fault tolerant data processing, Reports not needed in real time, Queries that cannot be expressed by SQL and Data processing Jobs needs to be faster. These are the key reasons why Hadoop platforms so attractive [4]. II. OVERVIEW OF HADOOP PLATFORM 1.0 Hadoop 1.0 popularized MapReduce programming for batch jobs and demonstrated the potential value of large scale, distributed processing. MapReduce, as implemented in Hadoop 1.0, can be I/O intensive, not suitable for interactive analysis [5]. In Hadoop 1.0, a single Namenode managed the entire namespace for a Hadoop cluster

2. Problem Statement

2.1 Existing System

Previously they used Lexicon-based which belong to unsupervised techniques, this method classify the data into two classes of positive or negative. This lexicon-based method uses the help of dictionary to classify the tweet into positive sentiment or negative sentiment. There are some steps of lexicon-based that is used in this research, such as determining the polarity of words, negation handling, and also giving score to every each entity in the tweet.

For those they are going to download the libraries that are provided by the twitter guys by using this they are crowded the data that we want predominantly. After getting raw data they will filter it and they will find out the constructive, negative and moderate words from the list of collected words in a text file. All these words should be collected by us to filter out or do some twitter analysis on the filtered data. These words can be called as a dictionary set by which they will perform twitter analysis.

Also, after performing all these things and they will store these in a database and coming to that they will use RDBMS. Where they are having limitations in creating tables and also accessing the tables effectively.

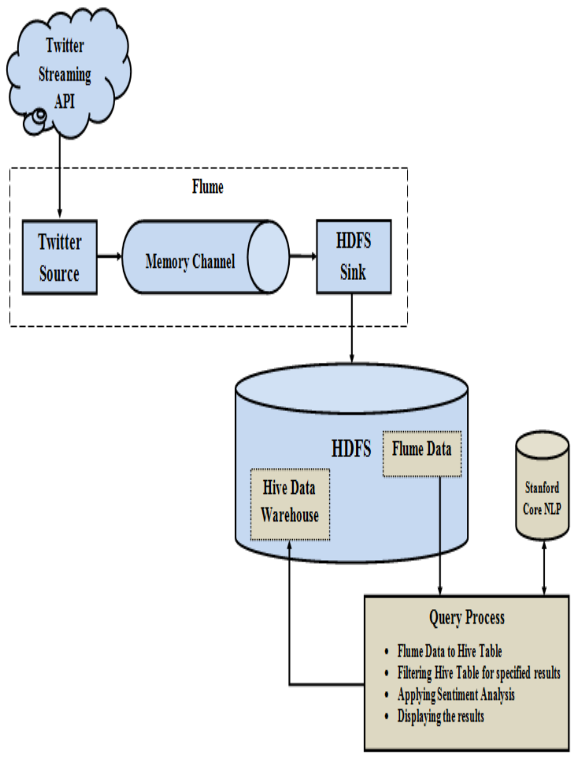
2.2 Proposed System

To overcome the drawbacks of existing system we are using Big Data problem statements. By using Hadoop and its Ecosystems, for getting raw data from the Twitter by using Hadoop online streaming tool called Apache Flume. By using this tool we are going to configure everything that we want to get data from the Twitter. For this we want to set the configuration and also want to define what information that we want to get from Twitter.

All these will be saved into HDFS (Hadoop Distributed File System) in our prescribed format. From this raw data we are going to filter the information that is needed for us. And from that we are going to perform the twitter Analysis by using some UDF's (User Defined Functions) by which we can perform twitter analysis by taking Stanford Core NLP as the data dictionary so that by using that we can decide the list of words that coming under constructive, moderate and negligible.

The subsequent figure shows clearly the architecture view for the proposed system by this we can understand how our project is effective using the Hadoop ecosystems and how the data is going to store from the Flume, also how it is going to create tables using Hive also how the sentiment analysis is going to perform[8].

Fig. 2: Architecture diagram for proposed system.



3. Methodology

3.1. Extracting Twitter Data with Apache Flume

Apache Flume is one of the distribution frameworks which have the capability to use HDFS,HBase, Mongo DBetc., as Sink. As, the Twitter [4] Streaming API gives a constant stream of tweet data coming from the application it must reside in HDFS securely. The security can be ensured by the generation of keys at the time of creating an application in twitter.

3.2 Querying JSON Data with Hive

Hive will expect the input data in a delimited row format. But the twitter data will be in a JSON [28], format. So to handle this type of data hive will use Hive SerDe interface to interpret the data which comes through twitter.

SerDe stands for Serializer and Deserializer, these are the interfaces that make Hive to translate the data into the form that Hive can process. The Deserializer interface is used when we read data from the disk, and converts the data into format where Hive knows how to manipulate. The data has some structure and sometimes it don't even have a structure, but certain fields may or may not exist. This semi-structured nature of the data makes the data very hard to query in a traditional RDBMS. Hive can handle this data. So, eventually Hive can also handle the log files of web servers which may be in CSV, TSV or any unstructured, semi-structured formats.

3.3 PLAN OF WORK

This project involves the following modules:

- Collecting the required input data [Module I]
- Setting the cluster to store the data [Module II]
- Transfers the data into hive datawarehouse [Module III]

3.3.1 Collecting the required input data [Module I]

By application of twitter JSON format files are generated which have to be dumped into the hdfs file system of Hadoop.Tweeted, Re-tweeted and commented data regarding to particular issue, individual's opinion, trending issues, website popularity clicks from Twitter for Twitter. Sentiment analysis [21] and log file from webserver for log analysis.

3.3.2 Setting the cluster to store the data [Module II]

In this module, a cluster has been created, to store the metadata (namespace of all files) that will be stored in one node, backup of metadata in another node, to assign the jobs (job tracker) for execution will be in another node. To store the data and processing, remaining nodes have to be there. The architecture of the cluster [18] will be like the following

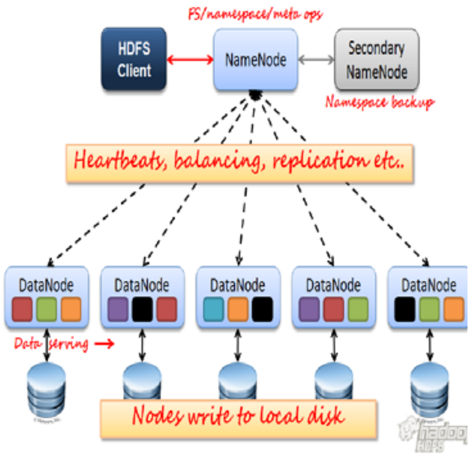
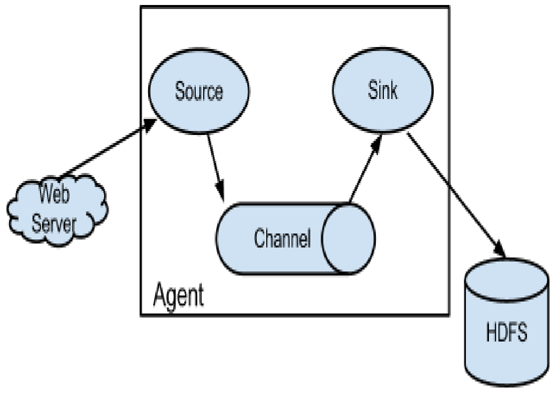


Fig. 4: Describes clearly of HDFS

3.3.3 Transfers the data into cluster [Module III]

Dumping the chunks of data from module I to the cluster [18] file system in all data nodes (hdfs, Hadoop distributed file system) using flume from the name node by forming the data pipeline between them as shown below: Fig. 5: Describes clearly of data flow



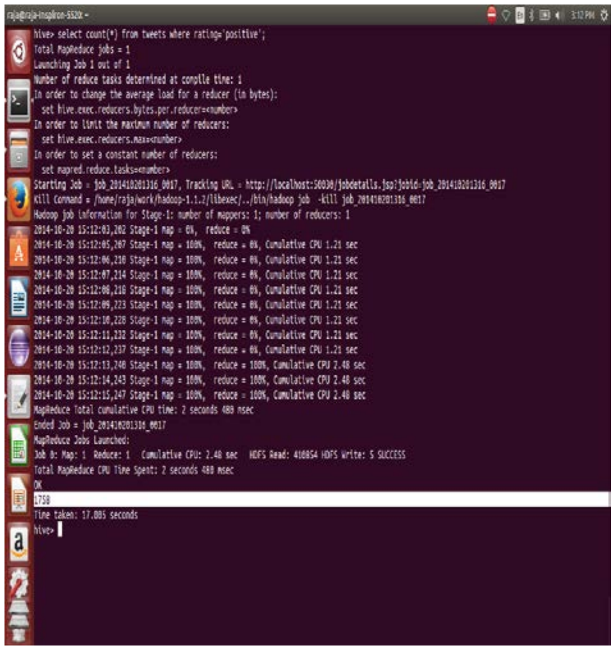
The Log file will also be dumped in hdfs through Hadoopfile system command.An external source like a web server delivers events to get consumed by Flume source. Format of events sent by the external source to the target flume source should be recognized by it. If we consider an example, Avro events can be received by an Avro Flume source from Avro clients or other Flume agents from an Avro sink in the flow that send events. Using a Thrift Flume Source a similar flow can be defined to receive events from a Thrift Sink or a Flume Thrift Rpc Client or Thrift clients written in any language generated from the Flume thrift protocol. Event is stored into one or more channels when a Flume source receives it. Until the event is consumed by a Flume sink it is kept by the passive store named channel. For example, file channel is one which is backed by the local file system.

Event is removed by the sink from the channel and kept into an external repository like HDFS (via Flume HDFS sink) or it can also be forwarded in the flow to the Flume source of the next Flume agent (i.e., next hop). With the events staged in the channel the source and sink within the given agent run asynchronously.

A critical-flow open flume with a constricted flow which causes a drop in the hydraulic grade line, creating a critical depth is a Venturi flume. For very large flow rates, usually given in millions of cubic units Venturi flume is used in flow measurement. Measurement by venture meter would normally measure in millimetres, but for venturi flume it is in metres.

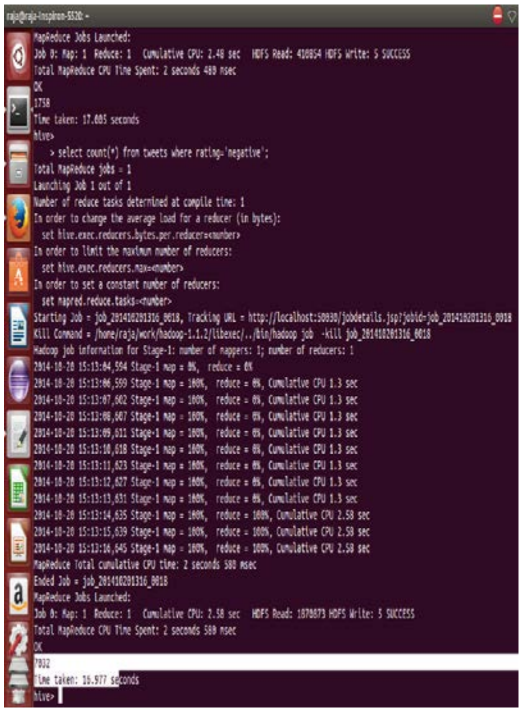
If the flow passes in a subcritical state through the flume, measurement of discharge with venturi flumes requires two measurements, one upstream and one at the throat (narrowest cross-section). A single measurement at the throat (which in this case becomes a critical section) is sufficient for computation of discharge, if the flumes are designed so as to pass the flow from sub critical to supercritical state while passing through the flume. The flumes are usually designed in such way as to form a hydraulic jump on the downstream side of the structure to ensure the occurrence of critical depth at the throat. These types of flumes are known as 'standing wave flumes' Sources

It is a part of Flume that is connected to a source of data and which starts the data along its travel through a Flume [30] dataflow. By sending the events into a channel a source processes and moves them along. The sources function by collecting discrete pieces of data, transforming the data into single events, and then by using the channel processes events one item at a time, or as a group.



```
raja@raja-Inspiron-5520: ~
hive> select count(*) from tweets where rating='positive';
Total MapReduce jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapred.reduce.tasks=<number>
Starting Job = job_201410201316_0017, Tracking URL = http://localhost:50030/jobdetails.jsp?jobid=job_201410201316_0017
Kill Command = /home/raja/work/hadoop-1.1.2/libexec/../bin/hadoop job  -kill job_201410201316_0017
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2014-10-20 15:12:03,202 Stage-1 map = 0%,  reduce = 0%
2014-10-20 15:12:05,207 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 1.21 sec
2014-10-20 15:12:06,210 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 1.21 sec
2014-10-20 15:12:07,214 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 1.21 sec
2014-10-20 15:12:08,218 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 1.21 sec
2014-10-20 15:12:09,223 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 1.21 sec
2014-10-20 15:12:10,228 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 1.21 sec
2014-10-20 15:12:11,232 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 1.21 sec
2014-10-20 15:12:12,237 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 1.21 sec
2014-10-20 15:12:13,240 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 2.48 sec
2014-10-20 15:12:14,243 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 2.48 sec
2014-10-20 15:12:15,247 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 2.48 sec
MapReduce Total cumulative CPU time: 2 seconds 480 msec
Ended Job = job_201410201316_0017
MapReduce Jobs Launched:
Job 0: Map: 1  Reduce: 1   Cumulative CPU: 2.48 sec   HDFS Read: 410854 HDFS Write: 5 SUCCESS
Total MapReduce CPU Time Spent: 2 seconds 480 msec
OK
1758
Time taken: 17.085 seconds
hive>
```

Event-driven or poll able are two flavors of sources. The basic difference between event-driven and poll able sources is how events are generated and processed. Through mechanismslikecallbacks orRPC calls event-driven sources typically receive events. In contrast, poll able sources, operate by polling for events every so often in a loop. This differentiation can be framed in another way as a push-versus-pull model, where event-driven sources have events pushed to them, and pollable sources pull events from a generator.



```
raja@raja-Inspiron-5520: ~
MapReduce Jobs Launched:
Job 0: Map: 1  Reduce: 1   Cumulative CPU: 2.48 sec   HDFS Read: 410854 HDFS Write: 5 SUCCESS
Total MapReduce CPU Time Spent: 2 seconds 480 msec
OK
1758
Time taken: 17.085 seconds
hive>
    > select count(*) from tweets where rating='negative';
Total MapReduce jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapred.reduce.tasks=<number>
Starting Job = job_201410201316_0018, Tracking URL = http://localhost:50030/jobdetails.jsp?jobid=job_201410201316_0018
Kill Command = /home/raja/work/hadoop-1.1.2/libexec/../bin/hadoop job  -kill job_201410201316_0018
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2014-10-20 15:13:04,594 Stage-1 map = 0%,  reduce = 0%
2014-10-20 15:13:06,599 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 1.3 sec
2014-10-20 15:13:07,602 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 1.3 sec
2014-10-20 15:13:08,607 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 1.3 sec
2014-10-20 15:13:09,611 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 1.3 sec
2014-10-20 15:13:10,618 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 1.3 sec
2014-10-20 15:13:11,623 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 1.3 sec
2014-10-20 15:13:12,627 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 1.3 sec
2014-10-20 15:13:13,631 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 1.3 sec
2014-10-20 15:13:14,635 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 2.58 sec
2014-10-20 15:13:15,639 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 2.58 sec
2014-10-20 15:13:16,645 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 2.58 sec
MapReduce Total cumulative CPU time: 2 seconds 580 msec
Ended Job = job_201410201316_0018
MapReduce Jobs Launched:
Job 0: Map: 1  Reduce: 1   Cumulative CPU: 2.58 sec   HDFS Read: 1870873 HDFS Write: 5 SUCCESS
Total MapReduce CPU Time Spent: 2 seconds 580 msec
OK
7032
Time taken: 15.977 seconds
hive>
```

Channels

Between the sources and sinks Channels act as a pathway. By sources events are added to channels and later by sinks removed from the channels. Multiple channels can actually support Flume data flows, which enables data flows, such as fanning out for replication purposes that are more complicated. For a Twitter example, we have defined a memory channel: TwitterAgent.channels.MemChannel.type = memory

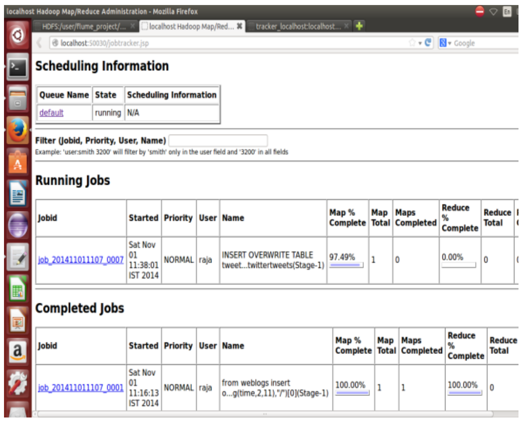
In-memory queue is used to store events until they're ready to be written to a sink by Memory channels. For data flows that have a high throughput Memory channels are useful. If the agent experiences a failure events may be lost, since they are stored in memory of the channel. Using a different type of channel, this situation can be remedied, if the risk of data loss is not tolerable.

Sinks

Sink is the final piece of the Flume dataflow. Sinks take Events are taken by sink and send to a resting location or forward them on to another agent. An HDFS sink, which writes events to a configured location in HDFS is utilized in Twitter example.

A number of things are done by the HDFS sink configuration we used: First, with the roll Count Parameter it defines the size of the files, so each file will end up containing 10,000 tweets. By setting the file Type to DataStream and setting write Format to Text it also retains the original data format. Instead of storing the data as a Sequence File or some other format this has been done. The most interesting piece, however, is the file path:

```
TwitterAgent.sinks.HDFS.hdfs.path =
hdfs://hadoop1:8020/user/flume/tweets/%Y/
%m/%d/%H/
```


localhost Hadoop Map/Reduce Administration - Mozilla Firefox

localhost:50030/jobtracker.jsp

Scheduling Information

Queue Name	State	Scheduling Information
default	running	N/A

Filter (Jobid, Priority, User, Name)
Example: 'user:smith 3200' will filter by 'smith' only in the user field and '3200' in all fields

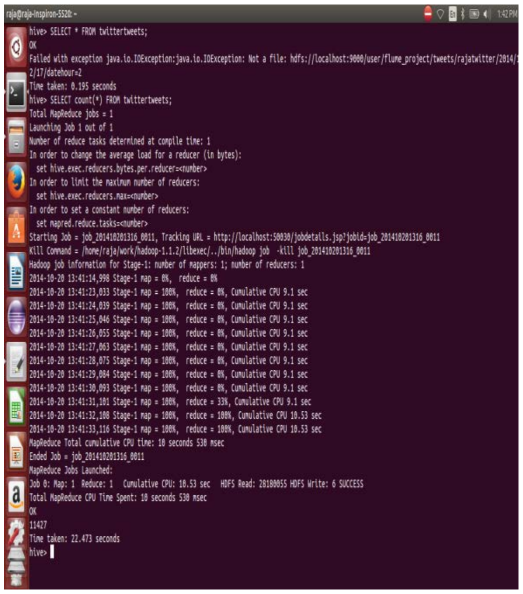
Running Jobs

Jobid	Started	Priority	User	Name	Map % Complete	Map Total	Maps Completed	Reduce % Complete	Reduce Total
job_201411011107_0007	Sat Nov 01 11:38:01 IST 2014	NORMAL	raja	INSERT OVERWRITE TABLE tweet...twittertweets(Stage-1)	97.49%	1	0	0.00%	0

Completed Jobs

Jobid	Started	Priority	User	Name	Map % Complete	Map Total	Maps Completed	Reduce % Complete	Reduce Total
job_201411011107_0001	Sat Nov 01 11:16:13 IST 2014	NORMAL	raja	from weblogs insert o...g(time,2,11),"/")[0](Stage-1)	100.00%	1	1	100.00%	0

The file path, as defined, uses some wildcards to specify that the files will end up in a series of directories for the year, month, day and hour during which the events occur. For example, an event that comes in at 8/20/2016 3:00PM will end up in HDFS at hdfs://hadoop1:8020/user/flume/tweets/2014/ 09/20/16/.Where does the timestamp information come from? If you'll recall, we added a header to each event in the Twitter [5] Source:


```
raja@raja-Inspiron-5520: ~
hive> SELECT * FROM twittertweets;
OK
Failed with exception java.io.IOException:java.io.IOException: Not a file: hdfs://localhost:9000/user/flume_project/tweets/rajatwitter/2014/1
2/17/datehour=2
Time taken: 0.195 seconds
hive> SELECT count(*) FROM twittertweets;
Total MapReduce jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapred.reduce.tasks=<number>
Starting Job = job_201410201316_0011, Tracking URL = http://localhost:50030/jobdetails.jsp?jobid=job_201410201316_0011
Kill Command = /home/raja/work/hadoop-1.1.2/libexec/../bin/hadoop job  -kill job_201410201316_0011
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2014-10-20 13:41:14,998 Stage-1 map = 0%,  reduce = 0%
2014-10-20 13:41:23,033 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 9.1 sec
2014-10-20 13:41:24,039 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 9.1 sec
2014-10-20 13:41:25,046 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 9.1 sec
2014-10-20 13:41:26,055 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 9.1 sec
2014-10-20 13:41:27,063 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 9.1 sec
2014-10-20 13:41:28,075 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 9.1 sec
2014-10-20 13:41:29,084 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 9.1 sec
2014-10-20 13:41:30,093 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 9.1 sec
2014-10-20 13:41:31,101 Stage-1 map = 100%,  reduce = 33%, Cumulative CPU 9.1 sec
2014-10-20 13:41:32,108 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 10.53 sec
2014-10-20 13:41:33,116 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 10.53 sec
MapReduce Total cumulative CPU time: 10 seconds 530 msec
Ended Job = job_201410201316_0011
MapReduce Jobs Launched:
Job 0: Map: 1  Reduce: 1   Cumulative CPU: 10.53 sec   HDFS Read: 28180055 HDFS Write: 6 SUCCESS
Total MapReduce CPU Time Spent: 10 seconds 530 msec
OK
11427
Time taken: 22.473 seconds
hive>
```

```
headers.put ("timestamp", String.valueOf (status.getCreatedAt ().getTime ()));
```

To determine the timestamp of the event this timestamp header is used by Flume, and the full path where the event should end up also resolved.

4. Conclusions

For getting raw data from the Twitter, Hadoop online streaming tool using Apache Flume is considered. In this tool everything that is needed to get data from the Twitter is configured. The proposed system collects data from Twitter social network site and does NLP techniques to extract features from the tweets. Word Sense Disambiguation and Word Net signets are used to increase the accuracy of prediction. Then various assortment methods of classification are applied to classify the data as Positive, Negative and Neutral. Analyzed that in twitter data, the posted comments are either positive or negative and also have proved that hive is the best method suited for doing this. It has been observed that the ensemble method outperforms the traditional classification methods. Of the ensemble methods Extremely Randomized Trees classification performs better than others.

5. Future Work

In this paper it has shown the way for responsibility sentiment analysis for Twitter data. Also, we can do this by using re-tweeted by creating work flow so that it can give a time slang such that it will work based upon that time we allocated for performing a particular work. Also at last we can also visualize the word map i.e., the most frequent words that are used in positive, moderate and negative fields by using R language to visualize.

Acknowledgments

We are grateful to express sincere thanks to our faculties who gave support and special thanks to our department for providing facilities that were offered to us for carrying out this project.

References

[1] Go, A., Bhayani, R., & Huang, L. (2009). Twitter sentiment classification using distant supervision. CS224N Project Report, Stanford, 1-12.

[2] Tang, H., Tan, S., Cheng, X., A survey on sentiment detection of reviews, Expert Systems with Applications: An International Journal, v.36 n.7, p.10760-10773, September, 2009.

[3] A. Pak and P. Parouek, “Twitter as a corpus for sentiment analysis and opinion mining,” in Proceedings of LREC, vol. 2010.

[4] J. Dean and S. Ghemawat, “MapReduce: Simplified Data Processing on Large Clusters,” Communications of the ACM, Vol. 51, Iss. 1, pp. 107-113, January 2008.

[5] S. Ghemawat, H. Gobioff and S-T. Leung, “The Google File System,” ACM SIGOPS Operating System Review, Vol. 37, Iss. 5, pp. 29-43, December 2003.

[6] K. Shvachko, H. Kuang, S. Radia, and R. Chansler, “The Hadoop Distributed File System,” in the 26th IEEE Symposium on Mass Storage Systems and Technologies, pp. 1-10, May 2010.

[7] Bahrainian, S.A., Dengel, A., Sentiment Analysis using Sentiment Features, In the proceedings of WPRSM Workshop and the Proceedings of IEEE/WIC/ACM International Conference on Web Intelligence, Atlanta, USA, 2013.

[8]“Sentimental Analysis”, Inc. [Online]. Available: http://www.cs.uic.edu/~liub/FBS/sentiment-analysis [Accessed 23 March 2013].

[9] (Online Resource) Hive (Available on:http://hive.apache.org/).

[10] (Online Resource)http://jsonlint.com/

[11] (Online Resource)http://nlp.stanford.edu/software/corenlp.shtml.

[12]T. White, "The Hadoop Distributed Filesystem," Hadoop: The Definitive Guide, pp. 41-73, Gravenstein Highwa North, Sebastopol: O’Reilly Media, Inc., 2010.

[13] (Online Resource) http://flume.apache.org/

[14]S. W. Ambler. Relational databases 101: Looking at the whole picture.www.AgileData.org, 2009.

[15] J. Manyika, M. Chui, B. Brown, J. Bughin, R. Dobbs, C. Roxburgh, and A. H. Byers, “Big Data: The Next Frontier For Innovation, Competition, And Productivity”, May 2011.

First Author: Sushmitha Reddy was born in Andhra Pradesh, India. She received the B.Tech Degree in Computer Science and Engineering from Jawaharlal Nehru Technological University Anantapur branch, India in 2014 and M.Tech Degree in also same branch and University. Her research interests are in the area of Semantic Web and Big Data Analytics.

Second Author: Smt.M. Padma M.E. was born in Andhra Pradesh, India. She is working as Asst.Prof. in Computer Science & Engineering Dept, GPREC Kurnool(district),A.P.518007,INDIA.Her research interest s are in the area of cloud computing and big data analyst.