TWITTER SENTIMENT ANALYSIS

Project Report

Raghul Somineni Raghupathy
Rsr379@nyu.edu

Table of Contents

1. Project Introduction

1.1 Introduction

This project is about extraction, storing and analyzing of live twitter data. Eventually, a sentiment analysis is done on the same to provide the nature of the tweets.

1.2 Project Objectives

- Setting up the environment for the data storage and process.
- Extraction of live twitter feed by establishing a connection with Twitter's API.
- Storage and processing of data on Hadoop (filtering and restructuring).
- Creation of tables on Hadoop to hold the twitter and also to provide an interface for simple querying for the users.
- Sentiment Analysis based on a keyword.
- Visualization of the sentiment analysis.
- Usage of Hadoop concepts.

1.3 Technologies Used

- **Hortonworks Hadoop on VM Workstation Player** – This has a CentOS bundled free with the Hortonworks Sandbox, which becomes the virtual guest system and the windows on which the VM runs is the host system.
- **PuTTY** – For remote access to Hadoop.
- **FileZilla (FTP Software)** – For any necessary file transfer b/w the host and the guest system.
- **Flume** – It is a distributed, reliable, and available service for efficiently collecting, aggregating, and moving large amounts of data.
- **Hadoop -** Open-Source software framework for storing data and running applications on clusters of commodity hardware. It provides massive storage.
- **Hadoop Distributed File System (HDFS)** – Distributed file system that provides high-performance access to data across Hadoop clusters.
- **Hive –** Data warehouse software project built on top of Hadoop for providing data summarization, query, and analysis.
- **MapReduce –** Programming model and an associated implementation for processing and generating big data sets with a parallel, distributed algorithm on a cluster.

2. System Architecture - Data Flow

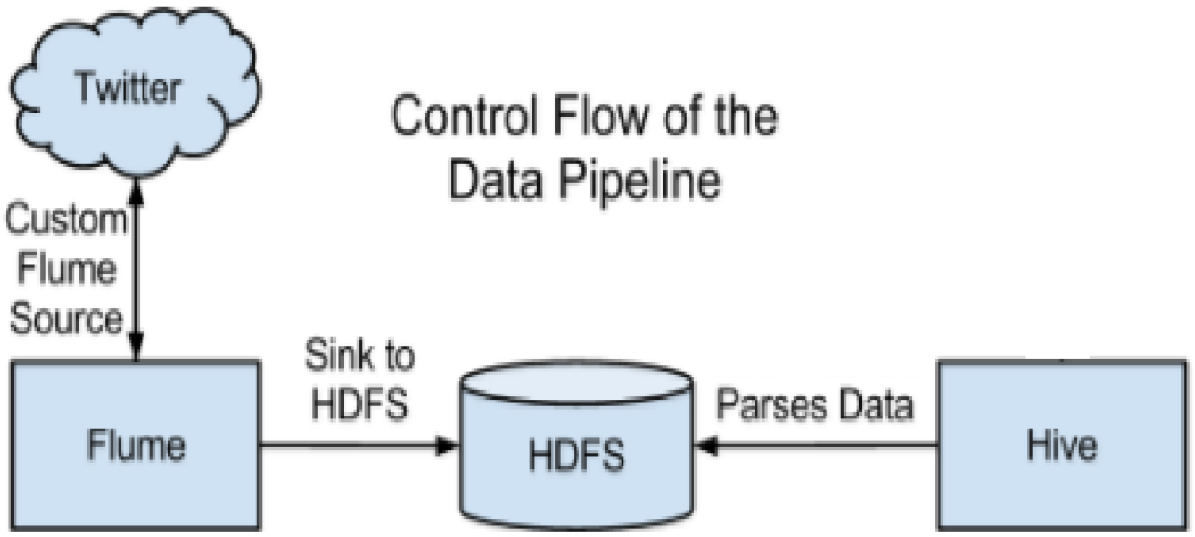


Figure 1: Control Flow of the Data Pipeline

As seen in the above figure the flow of data starts with flume accessing the live twitter feed using API access. Then, the data is stored in HDFS and the data is parsed by hive. Once this is done. We can go on to the sentiment analysis of the data.

3. Step by Step Implementation

3.1 Creating a Twitter Application

A new twitter application was created using my twitter account at https://dev.twitter.com. After filling a basic App info form, the application was created. Also, under the API keys tab the access token for the application was generated which would be used to provide authentication while accessing Twitter API and in turn its data.

On creation, we can see four pieces of information which would be used for authentication and they are:

- API Key
- API Secret

- Access Token
- Access Token Secret

3.2 Installation and Configuration of Flume

Initially I was trying to install flume without using the HortonWorks Platform. So, I had to install all the packages, set all the dependencies, class paths for all the packages that were needed for the flume to run. The initial steps are shown below:

- **Step 1:** Download latest Flume release from Apache Website.
- **Step 2**: Move/Copy it to the location you want to install, in my case it is "/usr/local/".

```
$cd Downloads/
$sudo cp apache-flume-1.6.0-bin.tar.gz /usr/local/
```

- **Step 3:** Extract the tar file. Go to the copied folder, in my case it is "/usr/local/" , run the below commands

```
$ cd /usr/local
$ sudo tar -xzvf apache-flume-1.6.0-bin.tar.gz
```

- **Step 4:** Rename folder from "**apache-flume-1.6.0-bin**" to "**flume**" for simplicity.

```
$ sudo mv apache-flume-1.6.0-bin flume
```

- **Step 5:** Update environments

```
$ gedit ~/.bashrc
```

- Add Below Lines:

```
export FLUME_HOME=/usr/local/flume
export FLUME_CONF_DIR=$FLUME_HOME/conf
export FLUME_CLASSPATH=$FLUME_CONF_DIR
export PATH=$PATH:$FLUME_HOME/bin
```

- **Step 6:** Change owner to user and group, in my case it is **user:hduser** and **group:hadoop**

```
$ sudo chown -R hduser:hadoop /usr/local/flume
```

- **Step 7:** Rename "**flume-env.sh.template**" to "**flume-env.sh**" and write the below values.

```
$ sudo mv /usr/local/flume/conf/flume-env.sh.template flume-env.sh
$ gedit /usr/local/flume/conf/flume-env.sh
```

- Add below line(please use your installed java version)

```
$JAVA_OPTS="-Xms500m -Xmx1000m -Dcom.sun.management.jmxremote"
export JAVA_HOME=/usr/lib/jvm/java-7-openjdk-amd64
```

- **Step 8:** Enter as hduser and run flume CLI:

```
$ flume-ng --help
```

This worked for a while and I had to come up with some other way to use flume once the VMware crashed. Then I decided to go with Hortonworks Platform.

We have already discussed why flume is being used in this project in section 1.3. Now for the installation of flume, all I had to run was a simple command:

```
yum install flume
```

After this command, flume was installed and was ready to be used.

The next step was to set the flume class path to the necessary .jar files.

Now that the agent code is in place, I needed to configure flume to create an agent using the class in the above set .jar. This was done by editing the flume.conf file.

In the flume.conf file, changes were made to accommodate the access tokens that were created in the twitter application. This is shown in the highlighted portion in the below snippet:

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

TwitterAgent.sources.Twitter.type = com.cloudera.flume.source.TwitterSource
TwitterAgent.sources.Twitter.channels = MemChannel
TwitterAgent.sources.Twitter.consumerKey = <consumerKey>
TwitterAgent.sources.Twitter.consumerSecret = <consumerSecret>
TwitterAgent.sources.Twitter.accessToken = <accessToken>
TwitterAgent.sources.Twitter.accessTokenSecret = <accessTokenSecret>

TwitterAgent.sources.Twitter.keywords = x-men, Interstellar

TwitterAgent.sinks.HDFS.channel = MemChannel
TwitterAgent.sinks.HDFS.type = hdfs
TwitterAgent.sinks.HDFS.hdfs.path =/user/root/data/tweets_raw
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.channels.MemChannel.type = memory
TwitterAgent.channels.MemChannel.capacity = 10000
TwitterAgent.channels.MemChannel.transactionCapacity = 100
```

TwitterAgent.sinks.HDFS.hdfs.path is the one that points to the NameNode and is the location in HDFS where the tweets will go. The keywords can be changed to get the tweets for any topic.

3.3 Twitter Data Extraction Using Flume

I have taken a few reference data files, which are listed below:

- **Dictionary** file has all the positive, negative and neutral words.
- **Time_zone_map** has mapping with time zones to countries. This will help us identify the country from a tweet.

The next step of the process was to use mkdir to create the directories necessary to hold the reference files and also the folders for holding the raw twitter data.

Now, the flume is started using the command:

```
/usr/lib/flume/bin/flume-ng agent –conf ./conf/ -f /etc/flume/conf/flume.conf -Dflume.root.logger=DEBUG,console -n TwitterAgent
```

This command starts loading the twitter data onto the sandbox.

3.4 Running Hive Scripts

Now that the data was loaded on to the sandbox, hive scripts were run to create tables and views on the extracted data files.

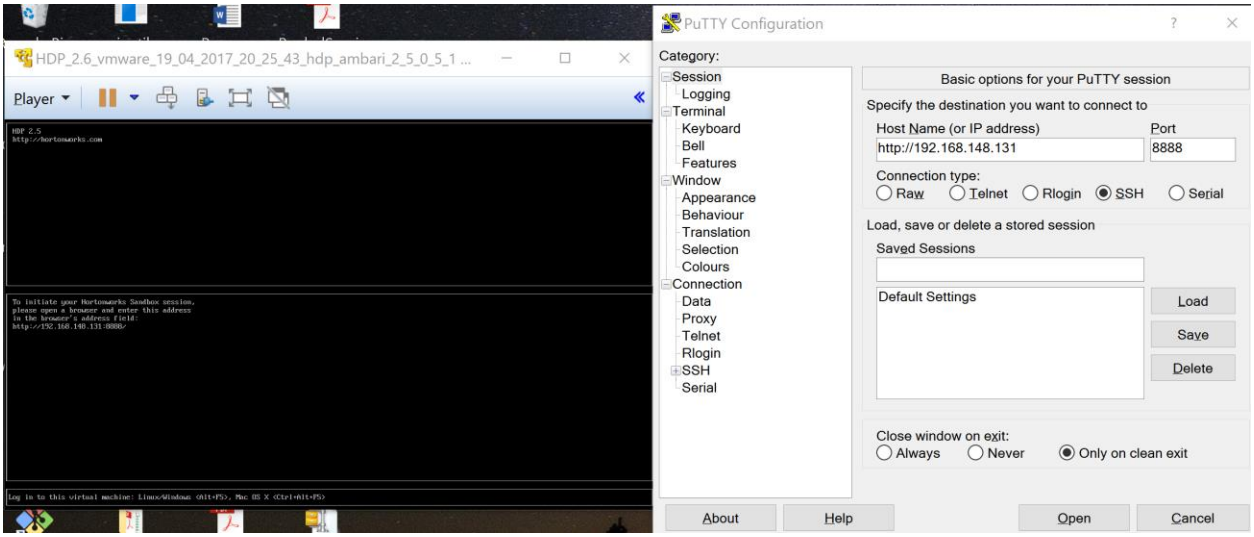
The scripts were written and run with the following objectives:

- Create table for the incoming data (Data Definition scripts)
- Conversion of raw twitter data into tabular format. (includes cleaning up the data)
- Use the dictionary to compute the sentiment of each tweet by analyzing the number of positive, negative and neutral words in a tweet. Based on this number, the sentiment of the tweet is decided.
- Creation of a new table that includes the sentiment for each tweet.

In order to run the hive scripts the following steps were followed:

- Start PuTTY with User Login:root\ Password: hadoop

- After the login the command prompt shows up with the prefix:
- The hive scripts are then run with the command: **hive -f tweets.sql**



Setting up PuTTY for remote access from windows host

Tweets.sql contains the scripts for the above mentioned objectives.

The snippet below shows the code for the computation of sentiment on tweets.

```
create view l1 as select id, words from Mytweets_raw lateral view explode(sentences(lower(text))) dummy as words;
create view l2 as select id, word from l1 lateral view explode( words ) dummy as word ;
```

```
create view l3 as select
    id,
    l2.word,
    case d.polarity
      when  'negative' then -1
      when 'positive' then 1
      else 0 end as polarity
 from l2 left outer join dictionary d on l2.word = d.word;
```

The above code snippets show that these scripts when run would compute the sentiment of the tweet by comparing the number of negative, positive and neutral words. Using this, the polarity of the tweets are decided.

```
create table tweets_sentiment as select
 id,
 case
   when sum( polarity ) > 0 then 'positive'
   when sum( polarity ) < 0 then 'negative'
   else 'neutral' end as sentiment
from l3 group by id;
```

Once the polarity has been set, the above script does the simple job of assigning the sentiments of the tweets as positive, negative or neutral. Finally, all the processed twitter data is exported as necessary.

3.5 Installation and Configuration of Hortonworks ODBC

The Hortonworks ODBC connector enables various BI tools and Excel to establish a connection and access the data in the Hortonworks Platform.

Once the HortonWorks ODBC driver was installed, the next thing to do was to configure it to enable a connection to the Hortonworks platform. For this, the configuration was done by opening the ODBC Data Source Administrator. Under the System DSN tab, the Hortonworks Hive DSN was selected by default and this was configured by providing the IP address of Hortonworks Sandbox.

The IP address of the sandbox is displayed in the command prompt window after the sandbox VM starts. With this, the ODBC connection was made between the windows host and the Hortonworks platform. Now data on the Hortonworks Platform can be exported to excel or other BI applications for visualization.

3.6 Accessing and Visualizing the Refined Sentiment Data with Excel

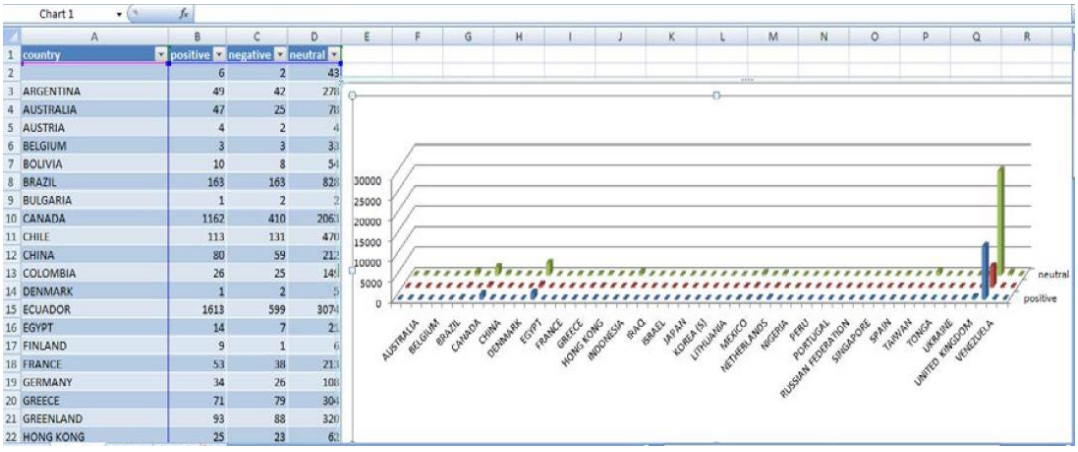
The last and the final step was to display the sentiment data in a presentable way. I decided to do this with MS Excel which worked out well. First, I had to set up the connection between HortonWorks platform and Excel. This is where the ODBC connectors that I installed came into use. Excel has the option of choosing the Data Source, which is found under "**Data -> From Other Sources -> From Microsoft Query".** After the data source has been selected as HortonWorks, we can view the tables that were created in section 3.4. The imported query data as appeared in the Excel Workbook:

	country	positive	negative	neutral
2		6	2	43
3	ARGENTINA	49	42	278
4	AUSTRALIA	47	25	78
5	AUSTRIA	4	2	4
6	BELGIUM	3	3	33
7	BOLIVIA	10	8	54
8	BRAZIL	163	163	828
9	BULGARIA	1	2	2
10	CANADA	1162	410	2063
11	CHILE	113	131	470
12	CHINA	80	59	212
13	COLOMBIA	26	25	149
14	DENMARK	1	2	5
15	ECUADOR	1613	599	3074
16	EGYPT	14	7	21
17	FINLAND	9	1	6
18	FRANCE	53	38	213
19	GERMANY	34	26	108
20	GREECE	71	79	304
21	GREENLAND	93	88	320
22	HONG KONG	25	23	62

Sheet1 Sheet2 Sheet3

Tweets categorized by Country and Sentiment

Now that the data has been imported successfully into Excel, we can use the Excel Column or Pie Chart to analyze and visualize the data. Some of the results obtained are shown below:



Number of Positive (Green), Negative (Red) and Neutral (Blue) Tweets by Country

3.7 Problems During the Project

- The major issue was the environment set-up which includes setting Hadoop, flume etc.

- Initially I had not used Hortonworks with its CentOS bundled Hadoop system, but had manually set up flume, Hadoop, java and other required packages. This worked for a while, I could retrieve the data and was working on processing it when my system gave up on me. I had to uninstall the whole VM Player and had to set up the whole thing again. Only this time it never worked.

- After that, I tried to change the tech being used. So, I considered and tried using HDInsight, which is a cloud distribution on Microsoft Azure. Even this failed me. I couldn't set up the environment properly.

- I also tried using spark and MLib, but I couldn't go through with it as I wanted to finish the one I put down on proposal first.

- For the short time that it did work, one of the major issues was cleaning up the raw twitter data for further analysis.

3.8 Conclusion and Next Steps

Even though this project is not complete, I believe that there is huge scope of development here. I plan to keep working on it and make it a better project. Also, I learnt a great deal about Big data and new technologies that are not in the curriculum.

In the future, I would like to create a proper web app with good user interface allowing them to analyze the data in any possible way they want. I also would like to compare the results of twitter analysis using different technologies. To be honest, I was trying to do that in the beginning stages which got down my work time on the proposed one.

I would like to thank the professor who gave us the freedom to choose any topic of our choice as our final project. Graduating this semester, I feel like I have learnt something about Big Data Analysis.

3.9 References

1. https://hortonworks.com/hadoop-tutorial/how-to-refine-and-visualize-sentiment-data/

2. http://blog.cloudera.com/blog/2012/09/analyzing-twitter-data-with-hadoop/

3. https://www.slideshare.net/OpenAnayticsMeetup/analyzing-twitter-data-with-hadoop-17718553

4. http://hadooptutorial.info/apache-flume-installation/