Twitter Data Retrieval and Analysis

Cohort1-Group 3:
Kirti Balagopal
Koundinya Nelanuthula
Meghna Sharma
Yeshwanth JA

Table of Contents

1.0 Business Case

Social media has become one of the most powerful tools in recent times for marketing and research that help organizations grow their business. To leverage the power of social media, we need to know how its users react when an event occurs (Presidential elections, Super Bowl, Apple product launch etc.). For this, we need to understand and analyze the data generated in the form of tweets, status updates etc.

Following the twitter hashtags on a particular trending issue can provide valuable business intelligence data such as mentioned below:

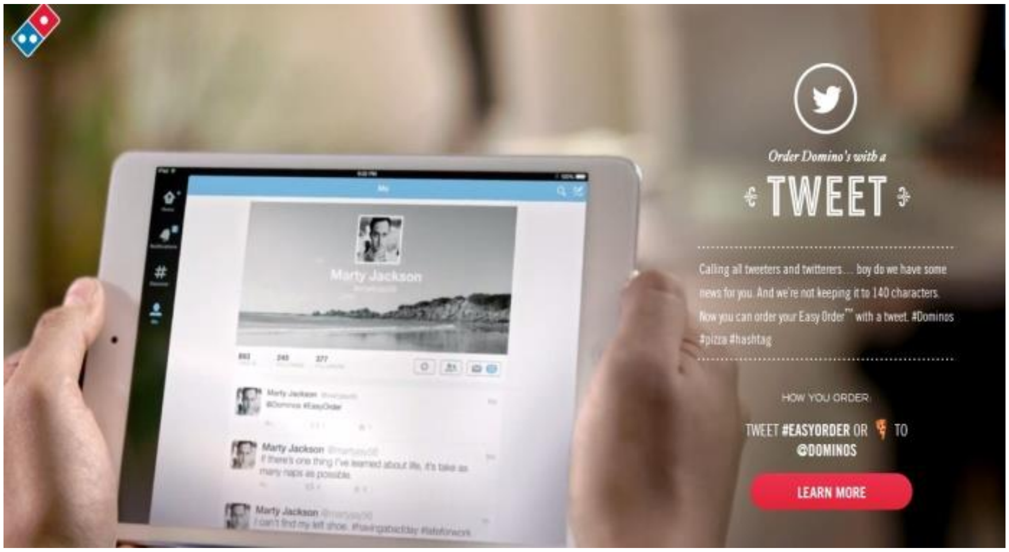
i. The public sentiments on an issue which in turn can help determine their attitude about the company. This aids the brand monitoring and in case of any negative tweets, the organization can take corrective measures to do immediate damage control.

ii. Demography of the community following the tweets and analysis of this data can provide valuable insights to outline target audience. Twitter marketing can thus help define the profiles of potential customers, their interests, locations etc.

iii. Performance of the hashtag content used in ad campaigns that can help find out what really resonates with the online community.

In this project, we plan to take twitter data samples with the requisite hashtags and provide useful results in the form of the number of users who tweeted using the hashtag, number of tweets containing media, number of tweets per users(celebrities or non-celebrities) etc. This data can very well be used in various data analysis models.

There are many instances in the business markets which have harnessed the power of the Twitter platform for marketing their products by reaching out to a global audience



Calling all tweeters and twitterers… boy do we have some news for you. And we're not keeping it to 140 characters. Now you can order your Easy Order™ with a tweet. #Dominos #pizza #hashtag

In May 2015, Domino's made use of one of the most innovative ways of technology for customers in the US to place order for their pizzas. This could be done by simply tweeting a pizza emoticon through a customer's twitter account or by tweeting #EasyOrder, provided the customer has registered their Twitter handle on their Domino's pizza profile. Once the customer confirms the message from Domino's, their order saved in their profile would automatically be delivered to their doorstep. This marketing strategy earned the brand a positive media coverage as well as the prestigious Cannes Lions award for creative campaigning.

Since 1997, Starbucks has been releasing various winter based designs on their iconic Christmas cups. But this year their holiday cup designs were kept minimalistic with a plain red color. People took it as an affront to the Christian faith. The term “red cup” was mentioned more than 61,000 times in a week. But deeper analysis proved that the outrage was more against publicizing a controversy against a beverage cup which impacted a whole nation. Thus it brought to light the need for a thorough clear sentimental analysis of trending issues to help organizations make strategic business decisions.

2.0 System Functionalities:

Our system will enable organizations to compare and contrast three different products or services from their own product line or from their competitors in the market. Some of the key advantages provided by the analysis carried out by our system are:

- Statistical data on a number of metrics as required by the organization which would aid in making critical decisions for business development and growth
- Potential to compare and contrast the data on any three products or services from their own product line or from their competitors in the market.
- Availability of the most trending news on one of the most influential social media platform which can be used for business intelligence.

Data Requirements

Since user profile data and hashtags from Twitter would be unstructured, data cannot be ordered in a relational or a hierarchical database. The data for our project will be extracted from Twitter using Apache Flume. The data would be in the JSON format and will be made available to the application logic. Data like username, twitter handle, location from where the user tweeted, number of followers etc. constitute our data.

Steps for Data Retrieval:

- Provide desired key words in the Flume configuration file and provide the Access Token and Secret Token provided by Twitter.
- Run the Flume agent using the open source jar provided in the readme manual.
- The data is then stored on HDFS in the form of JSON. This data is completely unstructured for the following reasons:

- All the users may not be verified users. So the verified field of that particular tweet would be empty.
- Some users may not have followers or may be following very few people.
- All this data cannot be stored in the form of relational data tables.

This raw data is converted into Hive Database, which is an eco-system of Hadoop and has in-built NoSQL implementation. Since major chunk of current days' data is schemaless and we wanted to make use of a database or a data warehouse that provides this functionality, we chose Hive.

3.0 Design Diagrams

3.1 Architecture Diagram

Hadoop Map-Reduce framework is built as Master-Slave architecture, with name node controlling the data nodes for data storage and job tracker controlling the task trackers for data processing purposes. Our system is a single node Hadoop cluster with a name node and a data node.

The application architecture of the system developed is Layered architecture. We have chosen this architecture as this correctly fits in with our business requirements. Also since we would like to expandour system later, this type of architecture is featured to be scalable. Since data is added to our system everyday, data can be added to the system without changing the application logic.

The components in the figure 3.1.1 are explained briefly below:

WebUsers: The users interact with the system with the front end provided. The user selects the operations of his choice to perform on the application, in this case – selecting the location wise tweet information or other metrics and submitting the options.

Presentation Layer: This layer provides the user interface that the web users interact with. Users provide the input and get the corresponding output, i.e the metric details the requested details.

Business Layer: The business layer for this project consists of the application logic to fetch raw data from HDFS and store it in Hive understandable format, which is later used for querying the results. The output is provided to the presentation layer after the operations are performed on the Hive database.

Datastore Layer: The datastore layer has the HDFS data repository to store the raw data that is fetched from Twitter. We have created a configuration file that has important information about what data should be retrieved and where the data should be stored on HDFS. An open source Flume jar is tailored to the configuration file that fetches the data from Twitter based on the parameters given in the config file.

DataSource: Twitter Inc. is our data source. Twitter provides an option to its users to access the public data by creating an application. This process is explained in details in the subsequent section.

Layered Architecture for Twitter Data Retrieval and Analytics

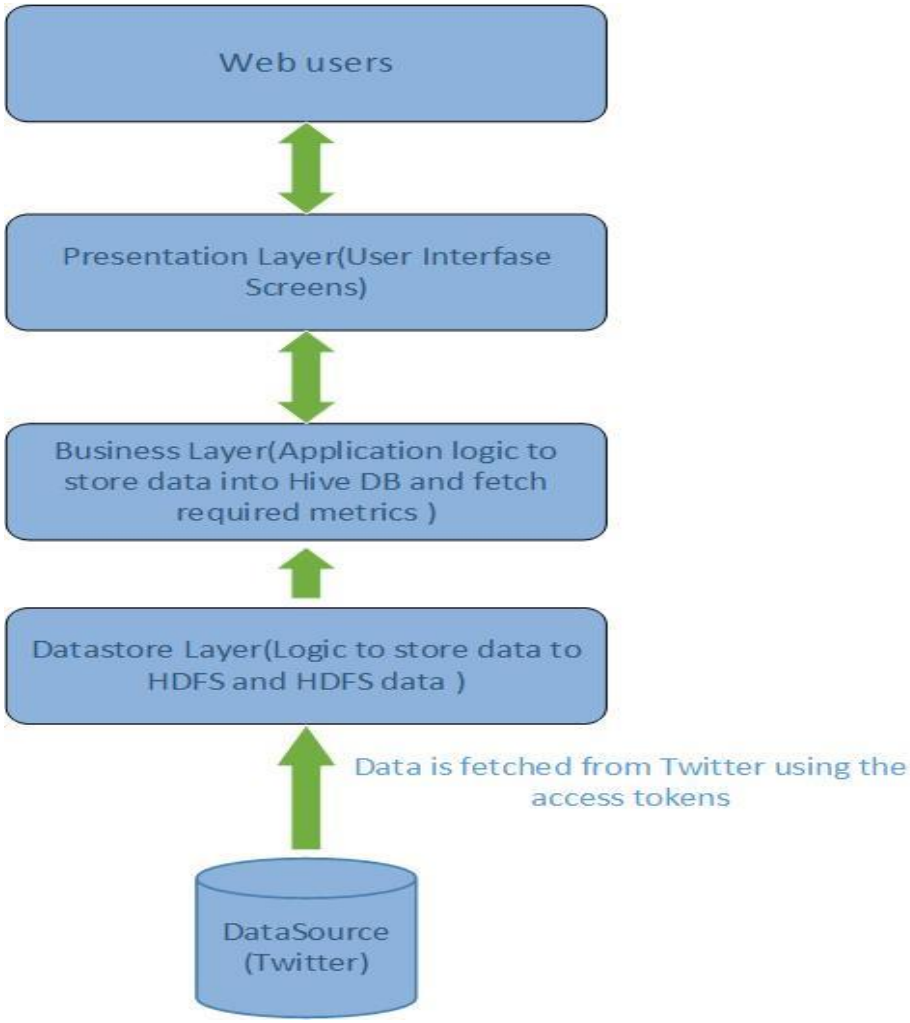

Web users
Presentation Layer(User Interfase Screens)
Business Layer(Application logic to store data into Hive DB and fetch required metrics)
Datastore Layer(Logic to store data to HDFS and HDFS data)
Data is fetched from Twitter using the access tokens
DataSource (Twitter)

Figure 3.1.1 Layered Architecture Diagram

3.2 Context Diagram

Twitter Data Analysis System:

As per the context diagram shown below, our system interacts with Twitter and gets the required data. The above shown boundary is considered because all the operations that are performed on the system begin after Twitter releases its data. Also, since Twitter releases data six hours after the tweet has been posted, our system does not have control over that time interval. Hence, Twitter API is external to our system.

Twitter API: All the information that the system gets is from Twitter, like the tweet text, the metadata of the tweet like tweet ID, user screen name, user geographic location, verified status, various counts related to the user etc. All the data that is fetched from Twitter API is mentioned as licensed data in its Developer Agreement and Policy as we have used one of our twitter accounts.

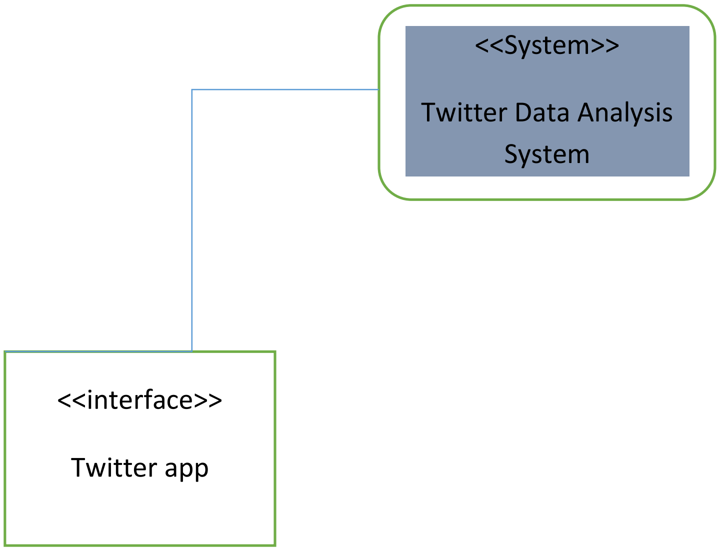


Figure 3.2.1 Context Diagram

3.3 Use Case Diagram

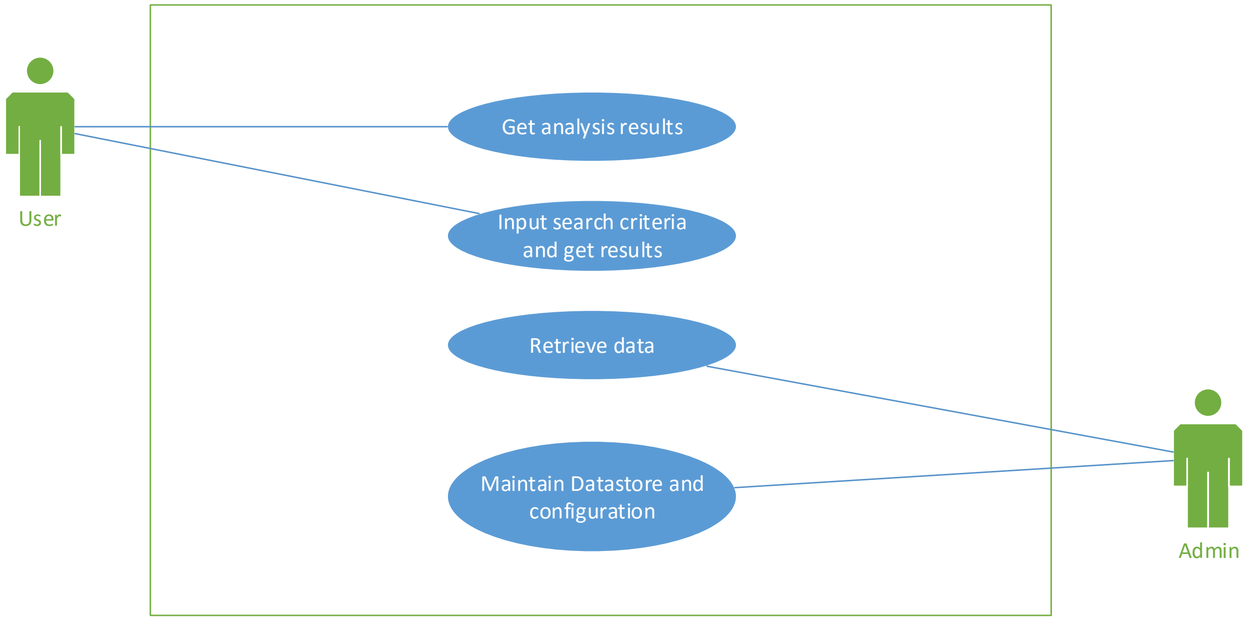

User
Get analysis results
Input search criteria and get results
Retrieve data
Maintain Datastore and configuration
Admin

Figure 3.3 Use Case Diagram

Use Case Description:

System	Twitter Data Retrieval and Analysis
Use case	**Retrieve data**
Actors	Admin
Description	The system Admin is responsible for retrieving data from Twitter by running the Flume agent. This data, which is unstructured, is stored on HDFS in the JSON format. Next, it is loaded into the HIVE database using HQL.
Data	Data retrieved from Twitter in JSON format

Stimulus	Admin runs the Flume agent for data extraction and HQL queries for storing in HIVE tables.
Response	Unstructured data retrieved from Twitter in JSON format is stored in the form of tables in the HIVE database.
Comments	The most recent data that can be extracted from Twitter is the one posted 6 hours before the retrieval time.

System	Twitter Data Retrieval and Analysis
Use case	**Maintain Datastore and configuration**
Actors	Admin
Description	The system Admin has to ensure the maintenance of data in the HIVE tables. This includes preserving data integrity, authenticity and
Data	Data available in HIVE tables
Stimulus	Admin runs the Flume agent for data extraction and HQL queries for storing in HIVE tables.
Response	Unstructured data retrieved from Twitter in JSON format is stored in the form of tables in the HIVE database.
Comments	The most recent data that can be extracted from Twitter is the one posted 6 hours before the retrieval time.

System	Twitter Data Retrieval and Analysis
Use case	**Input search Criteria and get analysis results**
Actors	User

Description	The system user can provide any three search keys of his choice on the system UI. Each of these would then be set in the corresponding configuration files that are used by the flume agent to retrieve the data from Twitter.
Data	Keywords to be used for search
Stimulus	User enters the keywords in the UI page
Response	Flume agent sets these keywords in the configuration files and uses them to retrieve data from twitter.
Comments	Although setting the keywords in the configuration file would be automated, running the flume agent would still remain a manual effort.

System	Twitter Data Retrieval and Analysis
Use case	Get analysis results
Actors	User
Description	The system user gets to download the report generated in the system. This is based on the data retrieved from Twitter for the keywords entered in the config file and the metrics chosen as the foundation for the data.
Data	Report based on the metrics provided to the user
Stimulus	User clicks on the button to generate the reports
Response	A report is provided to the user in a text file format which can be downloaded.
Comments	The report would be generated provide the user enters meaningful keywords to be searched

4.0 Workflow of the application

We have achieved the implementation of the application by using various technologies like Flume, Hive and Java Swings as mentioned in the previous section.

The high level overview (Figure 4.1) of the workflow is that firstly, we have created a Twitter App to post tweets and to get the access tokens and security tokens to download the tweets from Twitter. Then we have extracted the most recent twitter data using Flume and converted the unstructured format into a tables (structured format) using Hadoop Hive. Finally, we have presented the results of the analysis to the user by creating a front end using Java Swings.

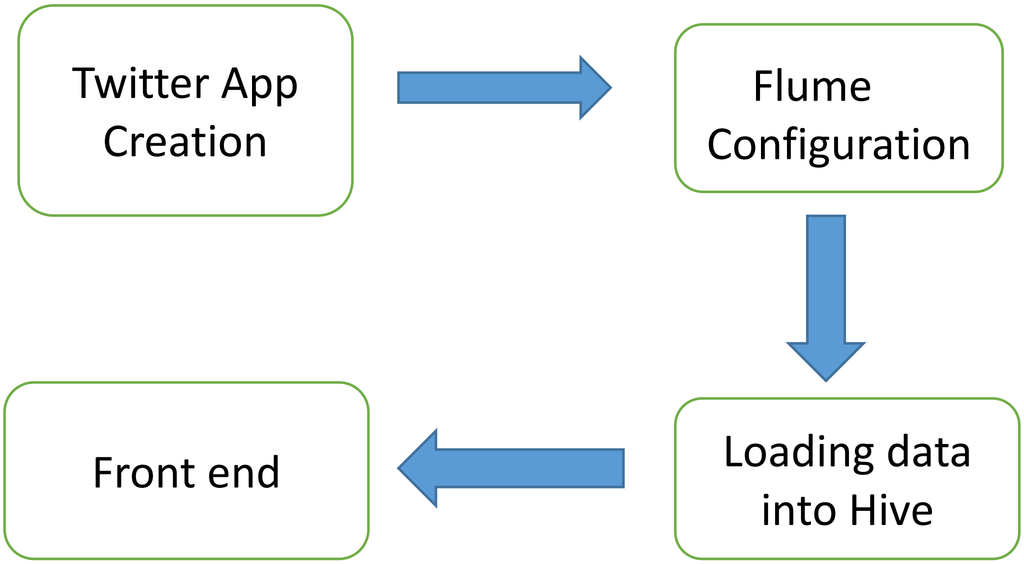
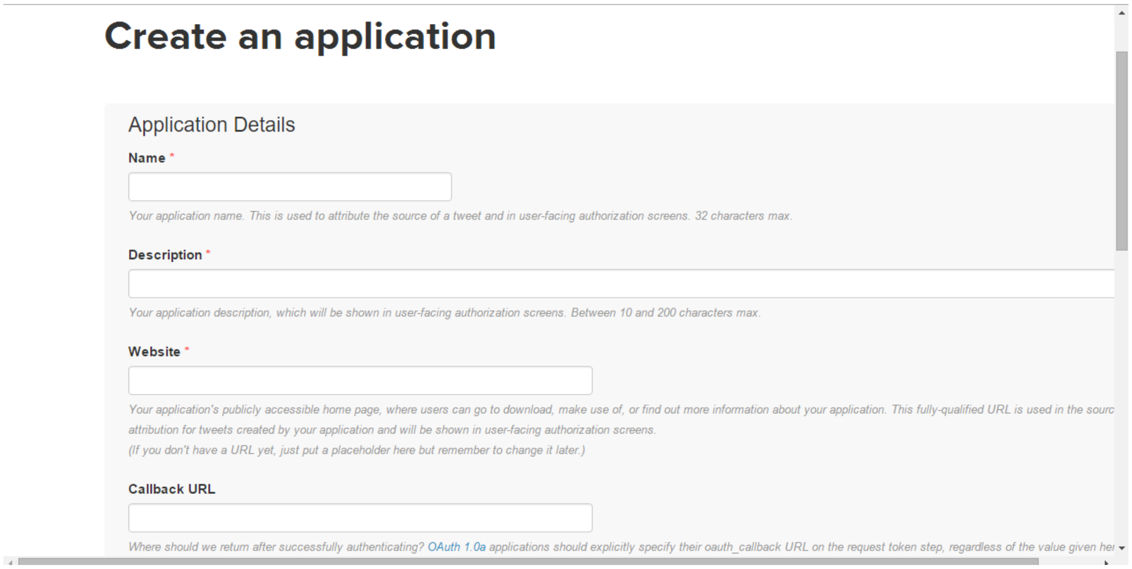


Figure 4.0.1 High Level Overview of the application workflow

4.1 Twitter App Creation

Each of the team members were successfully able to create the twitter account and post tweets so as to build on the data and get the most recent data. The limitation which twitter put recently on the use of its APIs is that the most recent data that can be extracted is the one posted 6 hours before the time of extraction.



Create an application

Application Details

Name *

Your application name. This is used to attribute the source of a tweet and in user-facing authorization screens. 32 characters max.

Description *

Your application description, which will be shown in user-facing authorization screens. Between 10 and 200 characters max.

Website *

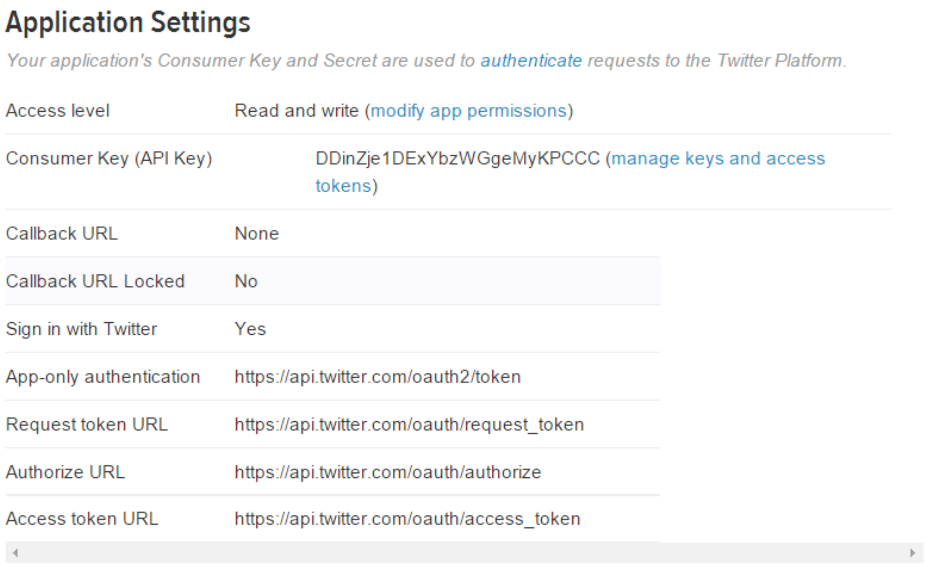
Your application's publicly accessible home page, where users can go to download, make use of, or find out more information about your application. This fully-qualified URL is used in the sourc attribution for tweets created by your application and will be shown in user-facing authorization screens.
(If you don't have a URL yet, just put a placeholder here but remember to change it later.)

Callback URL

Where should we return after successfully authenticating? OAuth 1.0a applications should explicitly specify their oauth_callback URL on the request token step, regardless of the value given her

Figure 4.1.1 Create a Twitter App

We created an app **(TwitDataRet)** on Twitter through which we could get the 'Customer Token', 'Customer Secret', 'Access Token' and 'Access Secret', thus receiving the permission to extract data from Twitter. These token are necessary to extract data from Twitter (Figure 4.1.2).



Application Settings

Your application's Consumer Key and Secret are used to authenticate requests to the Twitter Platform.

Access level	Read and write (modify app permissions)
Consumer Key (API Key)	DDinZje1DExYbzWGgeMyKPCCC (manage keys and access tokens)
Callback URL	None
Callback URL Locked	No
Sign in with Twitter	Yes
App-only authentication	https://api.twitter.com/oauth2/token
Request token URL	https://api.twitter.com/oauth/request_token
Authorize URL	https://api.twitter.com/oauth/authorize
Access token URL	https://api.twitter.com/oauth/access_token

Figure 4.1.2 Tokens and App settings

4.2 Flume Configuration

We used Apache Flume for the Twitter data extraction and loading the retrieved data into HDFS. We were able to successfully install and run Flume in the LINUX OS run on the virtual machine.

How does Flume work?

Apache Flume efficiently collects aggregates and moves large amounts of data from different sources to a centralized data store.

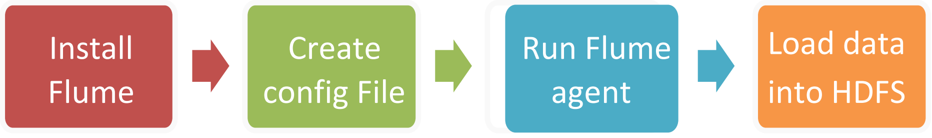


Figure 4.2.1 Flume Configuration

Firstly, we successfully installed and configured Flume on a Linux OS platform and resolved all the installation issues encountered. Then using the tokens received from Twitter, entering the source location, destination location and keywords, we created the configuration file (Figure 4.2.2). The Flume agent uses information present in the configuration file to get details about the source of the data (Twitter), the keywords (names of the movies) to extract data and the destination location where the data would be stored (HDFS). A Flume agent is a (JVM) process having components through which events flow from a source to the destination. Finally, Flume agent takes details from config file loads Twitter data into HDFS from Twitter app.

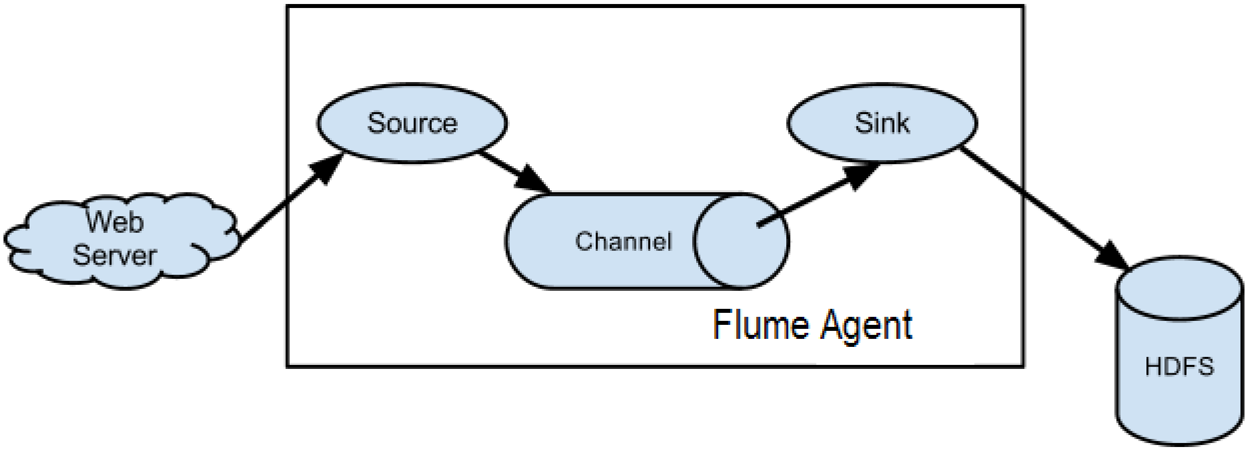
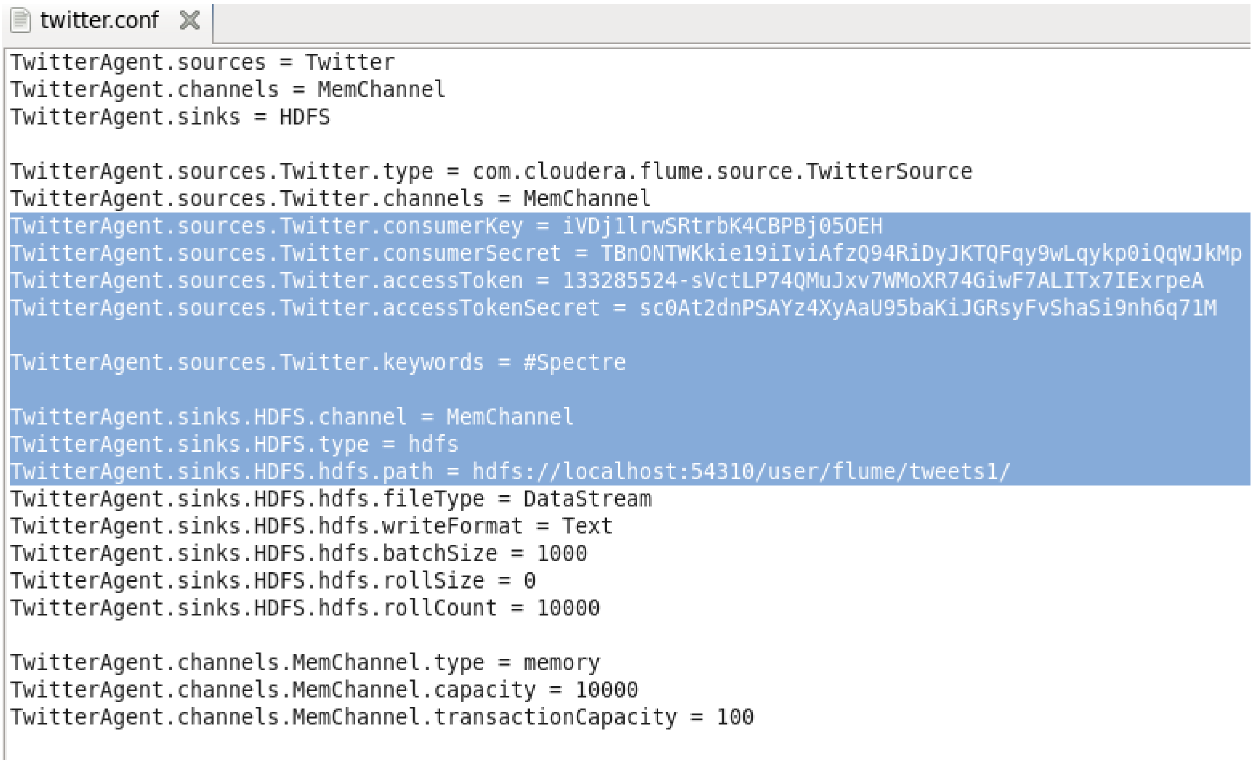


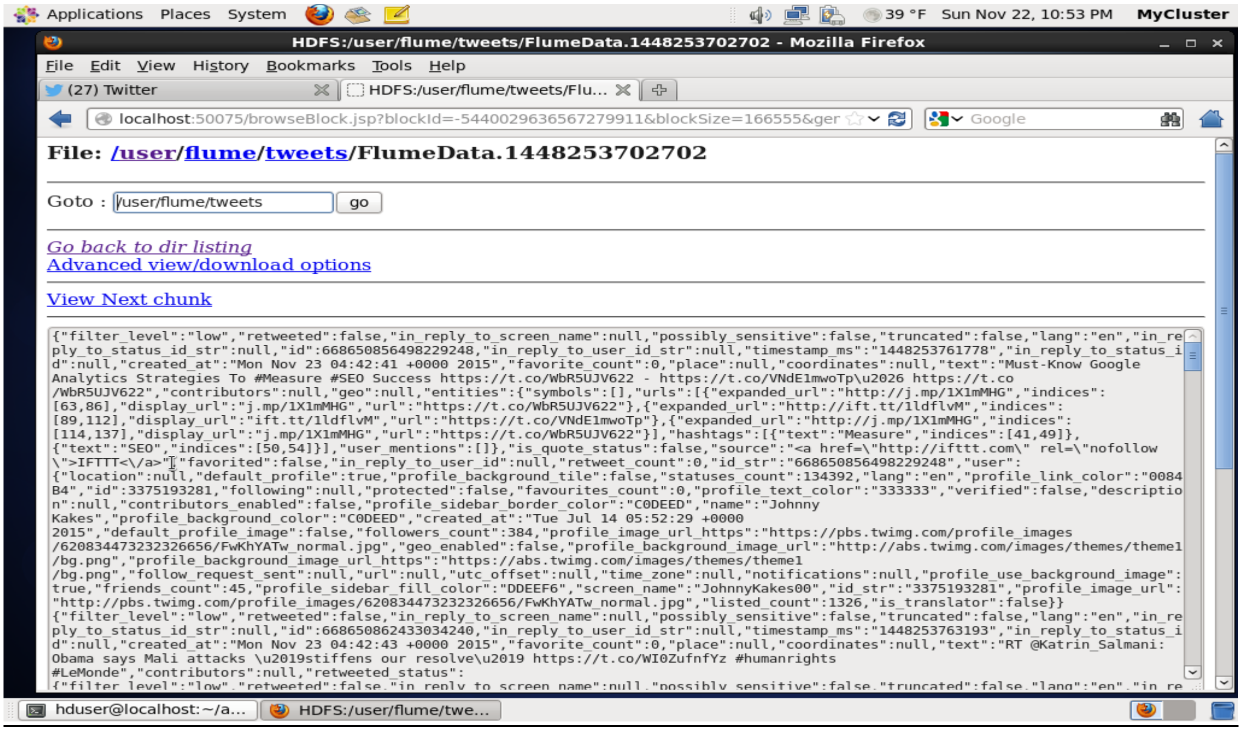
Figure 4.2.2 Flume Agent

The above figure depicts the process of how flume agent runs. The external source sends events to Flume in a format that is recognized by the target Flume source. The channel is a passive store that keeps the event until it's consumed by a Flume sink. The file channel is one example – it is backed by the local filesystem (HDFS is our case). The sink removes the event from the channel and puts it into an external repository like HDFS (via Flume HDFS sink) or forwards it to the Flume source of the next Flume agent (next hop) in the flow. The source and sink within the given agent run asynchronously with the events staged in the channel.



```
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 = iVDj1lrwSRtrbK4CBPBj05OEH
TwitterAgent.sources.Twitter.consumerSecret = TBnONTWKkie19iIviAfzQ94RiDyJKTQFqy9wLqykp0iQqWJkMp
TwitterAgent.sources.Twitter.accessToken = 133285524-sVctLP74QMuJxv7WMoXR74GiwF7ALITx7IExrpeA
TwitterAgent.sources.Twitter.accessTokenSecret = sc0At2dnPSAYz4XyAaU95baKiJGRsyFvShaSi9nh6q71M

TwitterAgent.sources.Twitter.keywords = #Spectre

TwitterAgent.sinks.HDFS.channel = MemChannel
TwitterAgent.sinks.HDFS.type = hdfs
TwitterAgent.sinks.HDFS.hdfs.path = hdfs://localhost:54310/user/flume/tweets1/
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
```

Figure 4.2.3 Flume Configuration File



localhost:50075/browseBlock.jsp?blockId=-5440029636567279911&blockSize=166555&ger

File: /user/flume/tweets/FlumeData.1448253702702

Goto : /user/flume/tweets go

Go back to dir listing
Advanced view/download options

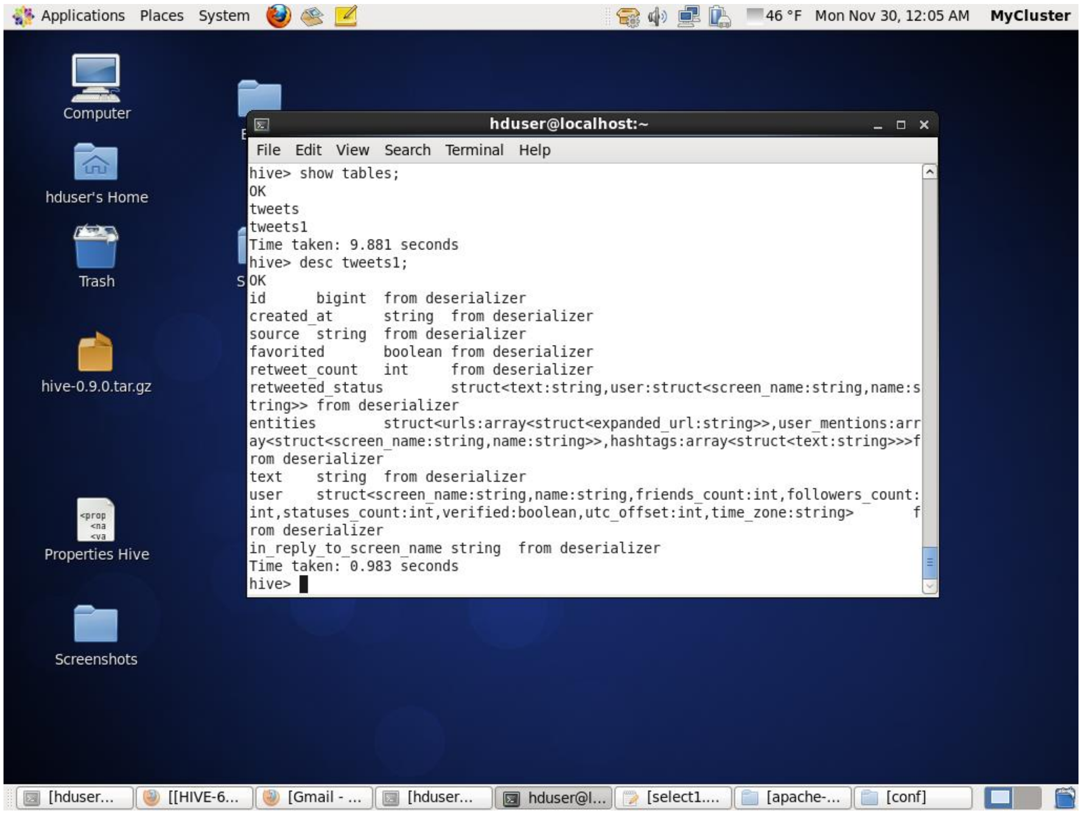
View Next chunk

```
{"filter_level":"low","retweeted":false,"in_reply_to_screen_name":null,"possibly_sensitive":false,"truncated":false,"lang":"en","in_re
ply_to_status_id_str":null,"id":668650856498229248,"in_reply_to_user_id_str":null,"timestamp_ms":"1448253761778","in_reply_to_status_i
d":null,"created_at":"Mon Nov 23 04:42:41 +0000 2015","favorite_count":0,"place":null,"coordinates":null,"text":"Must-Know Google
Analytics Strategies To #Measure #SEO Success https://t.co/WbR5UJV622 - https://t.co/VNdE1mwoTp\u2026 https://t.co
/WbR5UJV622","contributors":null,"geo":null,"entities":{"symbols":[],"urls":[{"expanded_url":"http://j.mp/1X1mMHG","indices":
[63,86],"display_url":"j.mp/1X1mMHG","url":"https://t.co/WbR5UJV622"},{"expanded_url":"http://ift.tt/1ldflvM","indices":
[89,112],"display_url":"ift.tt/1ldflvM","url":"https://t.co/VNdE1mwoTp"},{"expanded_url":"http://j.mp/1X1mMHG","indices":
[114,137],"display_url":"j.mp/1X1mMHG","url":"https://t.co/WbR5UJV622"}],"hashtags":[{"text":"Measure","indices":[41,49]},
{"text":"SEO","indices":[50,54]}],"user_mentions":[]},"is_quote_status":false,"source":"<a href=\"http://ifttt.com\" rel=\"nofollow
\">IFTTT<\/a>","favorited":false,"in_reply_to_user_id":null,"retweet_count":0,"id_str":"668650856498229248","user":
{"location":null,"default_profile":true,"profile_background_tile":false,"statuses_count":134392,"lang":"en","profile_link_color":"0084
B4","id":3375193281,"following":null,"protected":false,"favourites_count":0,"profile_text_color":"333333","verified":false,"descriptio
n":null,"contributors_enabled":false,"profile_sidebar_border_color":"C0DEED","name":"Johnny
Kakes","profile_background_color":"C0DEED","created_at":"Tue Jul 14 05:52:29 +0000
2015","default_profile_image":false,"followers_count":384,"profile_image_url_https":"https://pbs.twimg.com/profile_images
/620834473232326656/FwKhYATw_normal.jpg","geo_enabled":false,"profile_background_image_url":"http://abs.twimg.com/images/themes/theme1
/bg.png","profile_background_image_url_https":"https://abs.twimg.com/images/themes/theme1
/bg.png","follow_request_sent":null,"url":null,"utc_offset":null,"time_zone":null,"notifications":null,"profile_use_background_image":
true,"friends_count":45,"profile_sidebar_fill_color":"DDEEF6","screen_name":"JohnnyKakes00","id_str":"3375193281","profile_image_url":
"http://pbs.twimg.com/profile_images/620834473232326656/FwKhYATw_normal.jpg","listed_count":1326,"is_translator":false}}
{"filter_level":"low","retweeted":false,"in_reply_to_screen_name":null,"possibly_sensitive":false,"truncated":false,"lang":"en","in_re
ply_to_status_id_str":null,"id":668650862433034240,"in_reply_to_user_id_str":null,"timestamp_ms":"1448253763193","in_reply_to_status_i
d":null,"created_at":"Mon Nov 23 04:42:43 +0000 2015","favorite_count":0,"place":null,"coordinates":null,"text":"RT @Katrin_Salmani:
Obama says Mali attacks \u2019stiffens our resolve\u2019 https://t.co/WI0ZufnfYz #humanrights
#LeMonde","contributors":null,"retweeted_status":
{"filter_level":"low","retweeted":false,"in_reply_to_screen_name":null,"possibly_sensitive":false,"truncated":false,"lang":"en","in_re
```

Figure 4.2.3 JSON format for twitter data

4.3 Loading data into HIVE

Apache Hive is a data warehouse infrastructure built on top of Hadoop for providing data summarization, query, and analysis. We have used HiveQL to project structure (data in tables) onto the unstructured data stored in HDFS (Figure 4.3.1). We made use of an open source jar file hive-serdes-1.0-SNAPSHOT to structure data onto Hive repository. Of all the fields available in the JSON, only fields that are required for the metric calculation are considered.

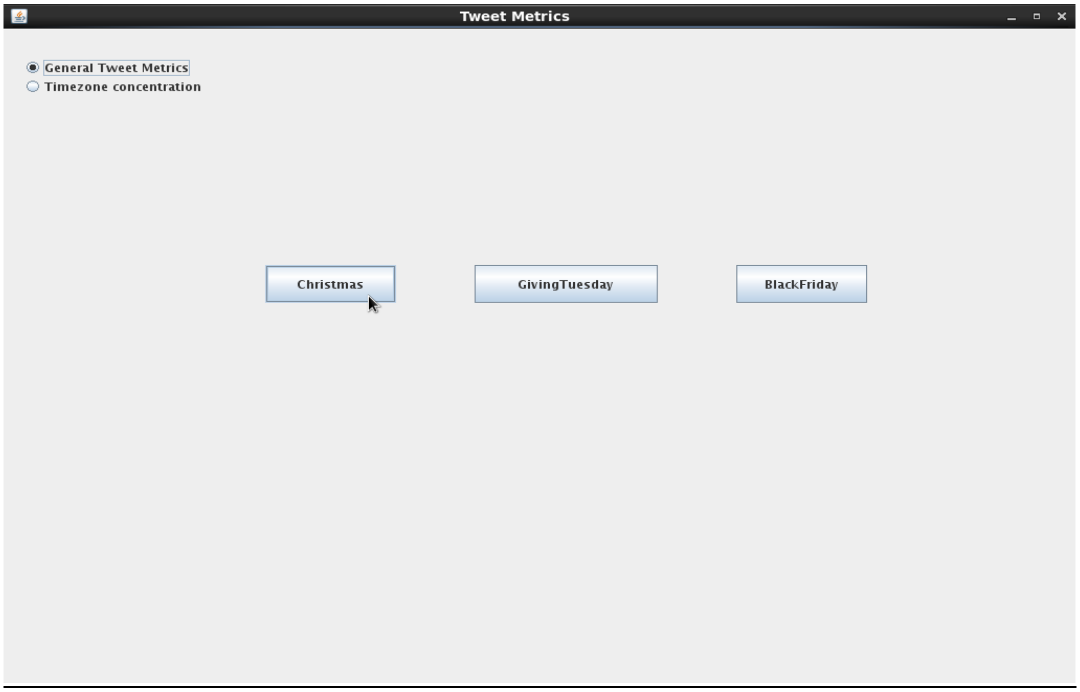


```
hive> show tables;
OK
tweets
tweets1
Time taken: 9.881 seconds
hive> desc tweets1;
OK
id      bigint  from deserializer
created_at      string  from deserializer
source  string  from deserializer
favorited       boolean from deserializer
retweet_count   int     from deserializer
retweeted_status        struct<text:string,user:struct<screen_name:string,name:s
tring>> from deserializer
entities        struct<urls:array<struct<expanded_url:string>>,user_mentions:arr
ay<struct<screen_name:string,name:string>>,hashtags:array<struct<text:string>>>f
rom deserializer
text    string  from deserializer
user    struct<screen_name:string,name:string,friends_count:int,followers_count:
int,statuses_count:int,verified:boolean,utc_offset:int,time_zone:string>       f
rom deserializer
in_reply_to_screen_name string  from deserializer
Time taken: 0.983 seconds
hive> 
```

Figure 4.3.1 HQL Query to convert unstructured data into tables

5.0 User Interface

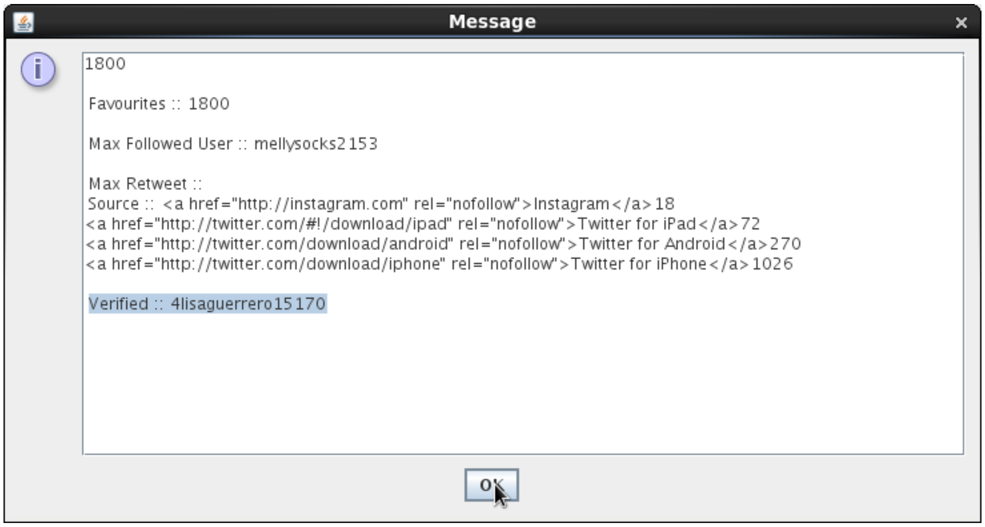
The user interface has been developed using Java Swings. On the interface (Figure 5.1), there are 3 buttons which represent the names of the movies. These are nothing but the 3 keywords which have been inserted in the configuration file on which the analysis is performed using the twitter data. These keywords have been assessed using a list of metrics that are mentioned in front of the checkboxes. User would be able to select various combinations of the metrics to be used to analyze the movies from twitter data.



● General Tweet Metrics
○ Timezone concentration

Figure 5.1 User Interface

On the click of any the buttons, the result of the selected metrics for that particular movie would be displayed to the user (Figure 5.2).



```
1800

 Favourites :: 1800

 Max Followed User :: mellysocks2153

 Max Retweet ::
 Source ::  <a href="http://instagram.com" rel="nofollow">Instagram</a>18
<a href="http://twitter.com/#!/download/ipad" rel="nofollow">Twitter for iPad</a>72
<a href="http://twitter.com/download/android" rel="nofollow">Twitter for Android</a>270
<a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a>1026

 Verified :: 4lisaguerrero15170
```

Figure 5.2 User Interface Results

6.0 User Guide

The interface is extremely user friendly and the user would be able to easily navigate through the GUI following the below steps:

i. After launching the GUI, the user will see option to click on one of the keywords.
ii. The user can either select the location-wise metric or other metrics option.
iii. Location-wise metric shows info about how the tweets are spread across different geographic locations.
iv. Other metrics gives the total number of tweets, number of verified users, total number of favorite tweets, user with maximum number of followers and device from which the tweet is posted.

The list of metrics used in the application will help the user to understand how to tweet more effectively and how to market and publicize their products more effectively by knowing the trending locations, topics and the format and content of the most popular tweets. The metrics used are described as follows:

i. **Location**–This metric will allow the user to understand in which languages the tweets on this particular movie is more popular.
ii. **User with maximum followers**– This gives the user's screen name along with the followers count for that batch of tweets.
iii. **Number of the Favorite tweets** - This metric will indicate the number of favorited tweets for that particular topic.
iv. **Max retweeted** - This metric will give the max retweeted tweet.
v. **Number of Verified users** – This metric tells the user how many of the users and the authenticate celebrity/ famous personalities.
vi. **Source of the Tweets** – This indicates the sources like the iPhone, Android, PC, Laptops have been used to tweet.

7.0 Limitations of the Existing System

Our system has a few limitations and these can be improved upon in future.

i. The current system doesn't perform the sentimental analysis of the tweets. Future scope can include the sentimental analysis which can indicate the positive and negative tweets.

ii. The Twitter allows downloading the data that is 6 hours old. So the application can get the data that was posted 6 hours ago , that will be the most recent data for our application.

8.0 Estimated Risks

As with any new implementation, there are a few risks associated with our application as well and we have noted them as below:

i. The system is currently being run on Linux platform. There may be installation and configuration issues which may hinder the functioning of the system on other platforms.

ii. The user interface screen that takes the keywords as entered by the user does not have any data quality or validation checks associated with it. The user may enter invalid keywords which would not be captured as an error by the system.

iii. It may be possible that Twitter stops giving out feeds for free to users, in which case cost of maintaining the project increases.

9.0 Bibliography and References

The books that we referred to complete this project successfully are as follows:

1. Hadoop in Action by Chuck Lam
2. NoSQL Distilled by Pramod J. Sadalage and Martin Fowler

Websites for reference:

1. http://hortonworks.com/hadoop-tutorial/how-to-process-data-with-apache-hive/
2. https://www-01.ibm.com/software/data/infosphere/hadoop/hive/
3. http://www.cloudera.com/content/www/en-us/resources/training/introduction-to-apache-hive.html
4. https://business.twitter.com/basics
5. http://www.cnbc.com/2015/11/09/starbucks-holiday-red-cup-brews-controversy-on-social-media.html
6. http://www.digitaltrends.com/social-media/dominos-tweet-to-order/
7. https://flume.apache.org/FlumeUserGuide.html