|  |  |
| --- | --- |
| Sentiment ANALYSIS In Twitter Messages and images-(challenge2) | CS590RA  Real Time big data analytics |

**By**

L V Sandeep Bathina Class-ID: 3

Rakesh Vistarakula Class-ID: 16

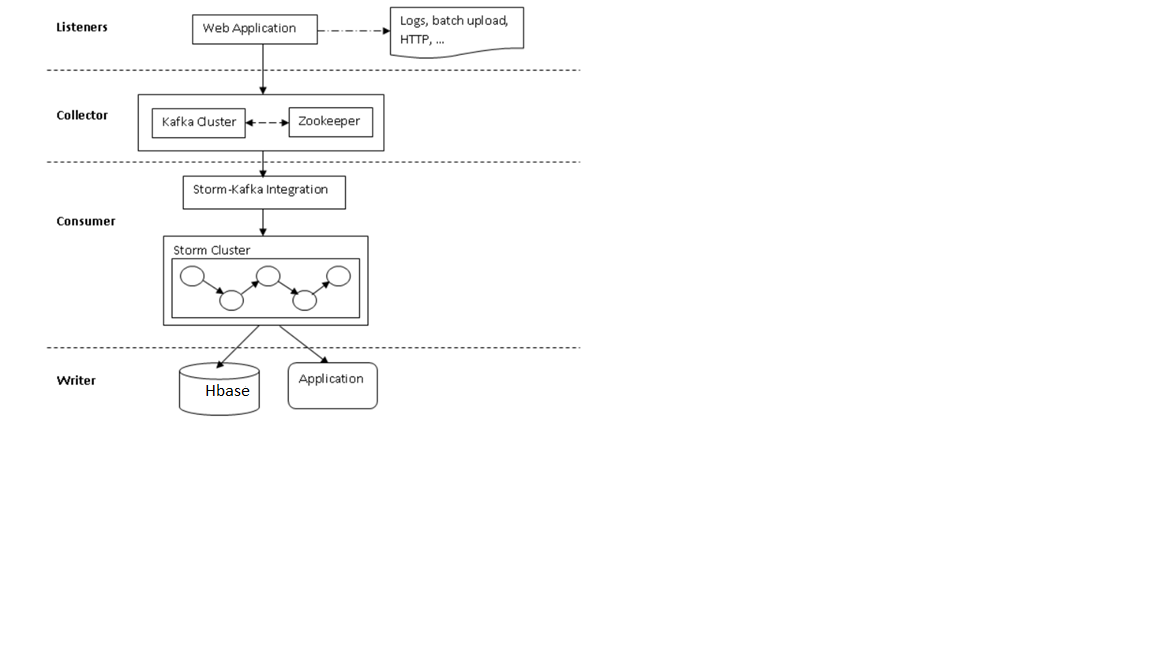
Lakshmana Dheeraj Polisetty Class-ID: 13

Balaji Gannavarapu Class-ID: 7

Table of Contents

|  |  |  |
| --- | --- | --- |
| S.No | Description | Page No |
| 1. | Design   1. Data model (features) 2. Sentiment analysis model and algorithm 3. Selection of datasets 4. Mobile App/web design | 3 |
| 2. | Features Implemented   1. Sentiment analysis algorithms 2. Predictive algorithms 3. Mobile User Interface | 9 |
| 3. | Outputs: | 13 |
| 4. | Website URL’s: | 17 |
| 5. | Github URL | 20 |
| 6. | Limitations | 24 |
| 7. | References | 25 |

1. Design
2. Data Model (features):



1. Listener:

The raw activity data has to be processed as a stream which is consumed by the storm.

1. Collector:

Kafka here is a messaging system which acts as pipeline between storm and data processing.

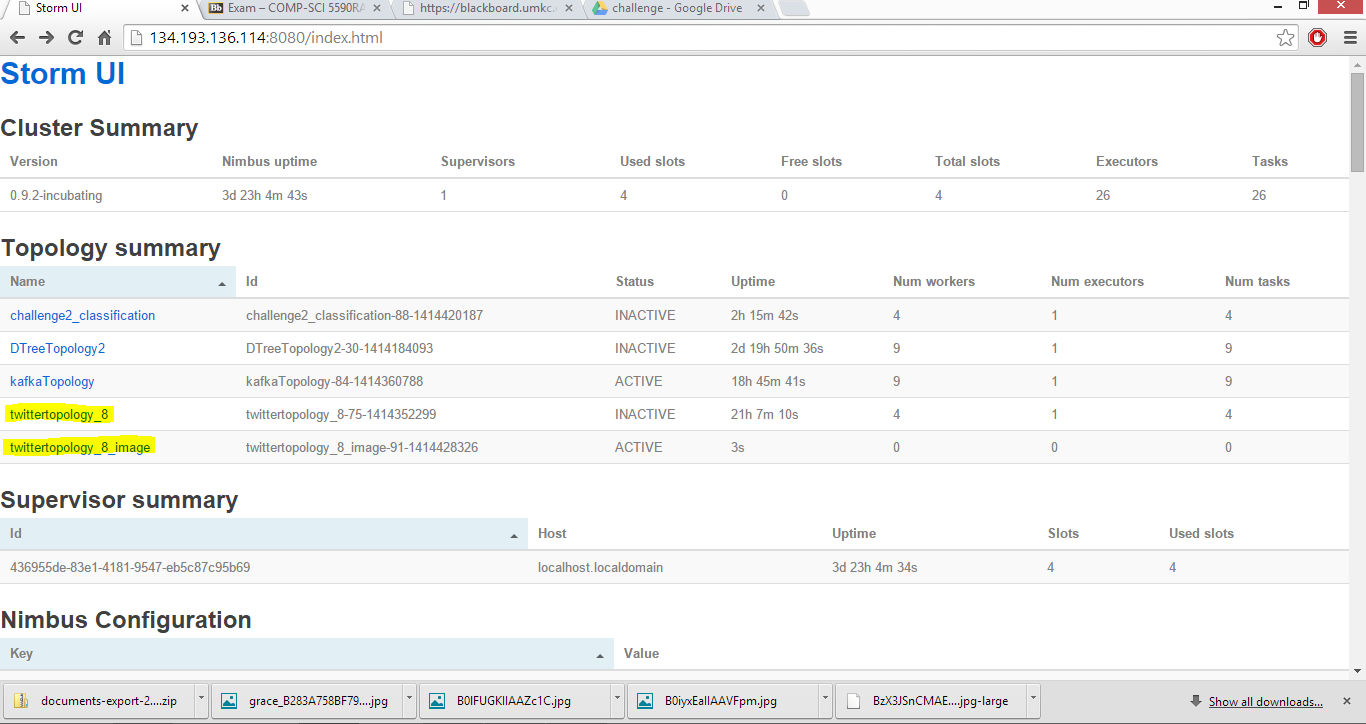
1. Consumer:

All data processed by collector consists of stream of data and some associated metadata.

1. Writer:

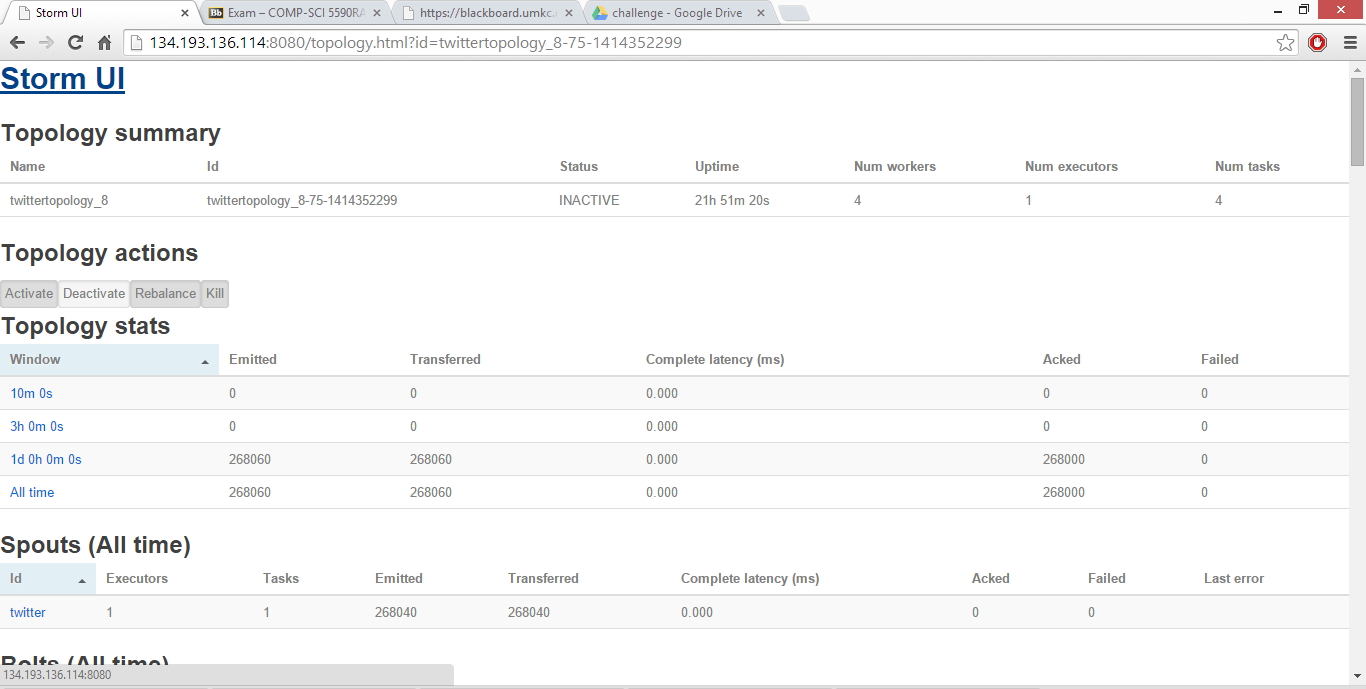
The standard data writer writes to applications and databases like hbase and file systems.

**TOPOLOGY:**



**Topology for Twitter messages:**

[**http://134.193.136.114:8080/topology.html?id=twittertopology\_8-75-1414352299**](http://134.193.136.114:8080/topology.html?id=twittertopology_8-75-1414352299)



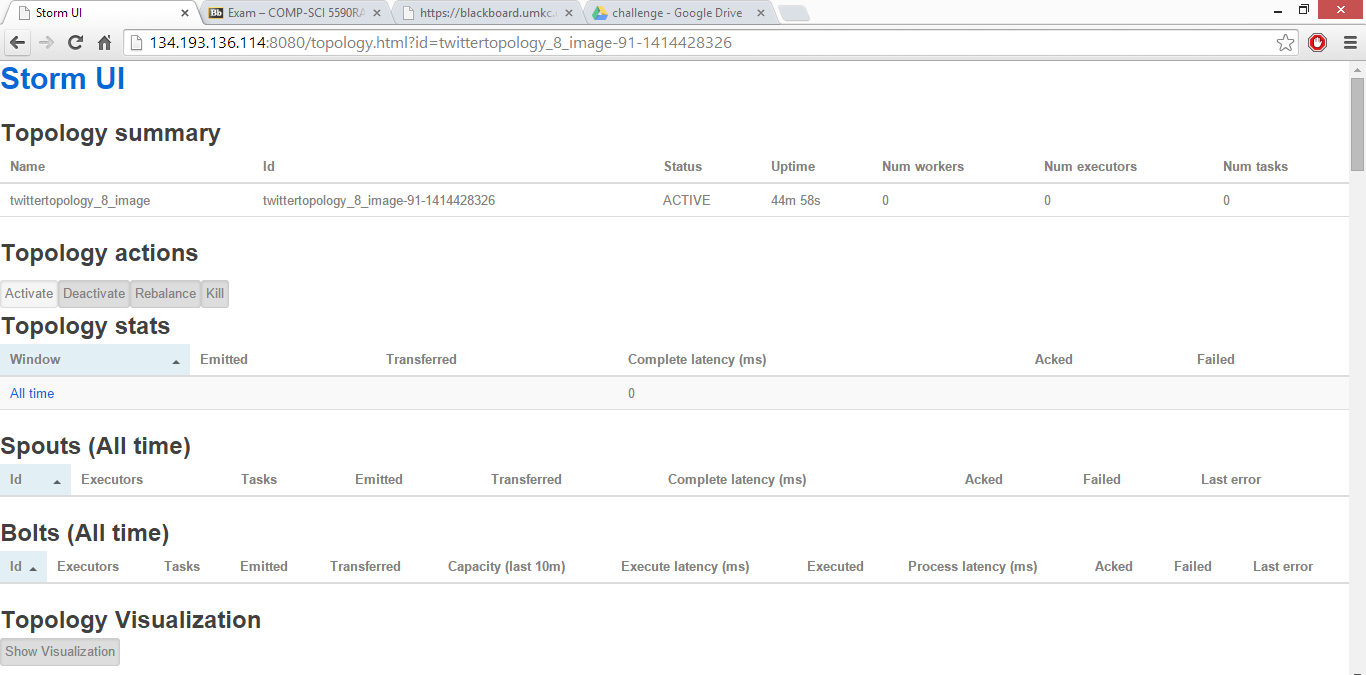
1. Initially we changed the multilang/resources/find\_all\_scriptures.py file by replacing all the bible data with movie collection data.
2. Next, the extracted tweets gets stored in the Twitter.txt file
3. We have also changed PrinterBolt file in bolt folder to save the tweets

Data.



**Topology for Twitter Images:**

[**http://134.193.136.114:8080/topology.html?id=twittertopology\_8\_image-91-1414428326**](http://134.193.136.114:8080/topology.html?id=twittertopology_8_image-91-1414428326)



As storm is not running, we couldn’t show the bolts and spout above. But we have run this before and got our images.

* 1. The Twitter Images of the movie gets saved in the ImageFolder.
  2. The spout here is modified by specifying a particular movie.
  3. All the related images of that particular movie gets saved.



1. Features Implemented:
2. Sentiment analysis and algorithms:

Naive bayes analysis algorithm: The system is mainly based on naïve-bayes classifier for detecting polarity of the tweets. The best performance here is calculated based on the polarity categories positive and negative. Evaluation results shows good performance when it is used to detect four sentiment categories.

1. Predictive algorithms:

Not yet used.

1. Mobile user interface:

The interface we developed here is by using Android application.

1. Initially a button is created and when the button gets clicked, all the tweets get displayed over here with their tweet ids.
2. The next button click dispalys Images related to that movie which are positive or negative.
3. The analysis button is created and when the user clicks on the button the sentiment analysis part of each tweet is calculated and displayed over there.
4. The next button created over here is the visualization button which displays the piechart of analysis part and no.of tweets of those are displayed here.
5. Data Collection:

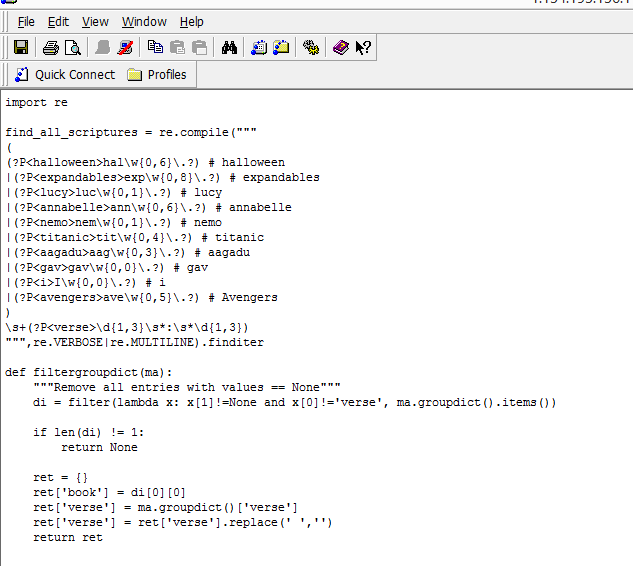
The real time data is collected from the twitter API using the Apache Storm technology.

To do this we are designing a Spout which will filter the data based on the Keyword and sends the data from twitter API to the Bolt. The Bolt will collect the data from the Twitter API and passes it to the Printer Bolt to write them to a file.

Tweets dataset:

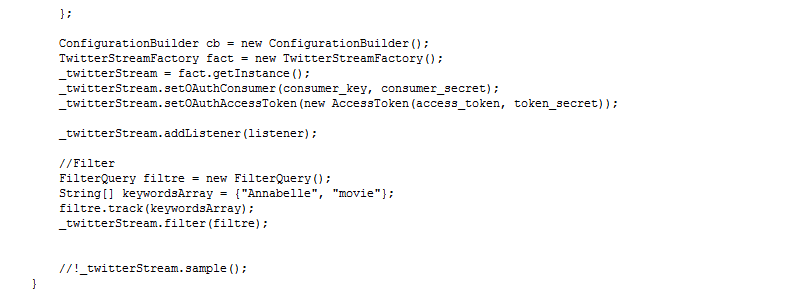
We have collected tweets data from twitter stream api,

The movie related tweets are collected by using below code:



Twitter Images:

Twiiter images are collected by using below code:



Feature Extraction and Classification:

We are analyzing the data collected in the above step using the Naive Bayes Algorithm.

This is done with the integration of Storm and Kafka. This is being done in two steps. First the data is being collected into the CSV file with the help of bolt named as CSV bolt. Then after that the data classification is done using the Weka bolt where we will use the Weka Tool for Training the data and analyze the sentiment.

Evaluation:

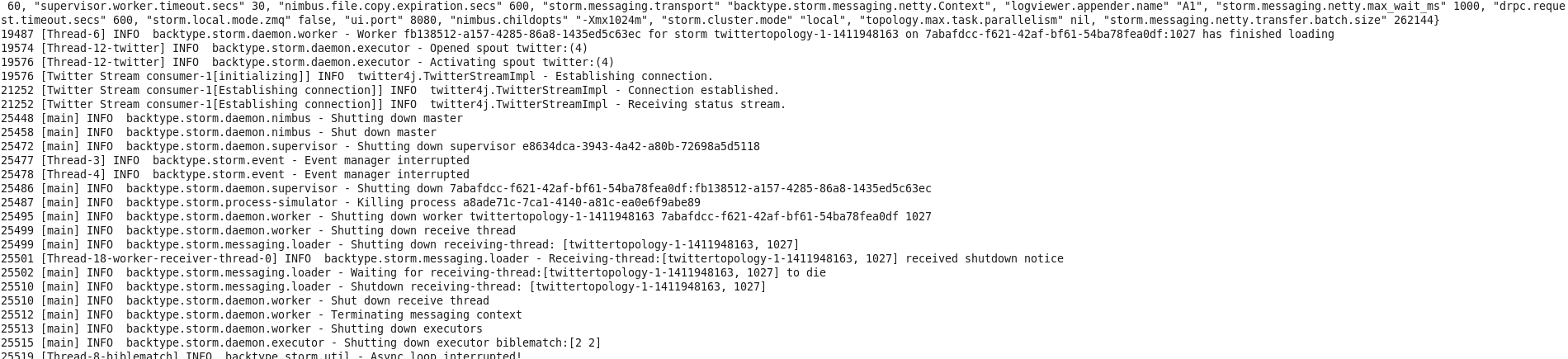
The evaluation of the data is being done using the SVM which uses the Weka Tool. Here the evaluation is done by converting the String data to numeric data and then applying algorithms on them.

1. Outputs:

Strom Remote:

1. Initially install storm , Kafka, zookeeper and hbase
2. Run the below command in storm
3. Before this, modify the habakkuk.properties file and provide your twitter id and keys. This is required to generate the live streaming twitter tweets.
4. The tweets gets displayed on the console.





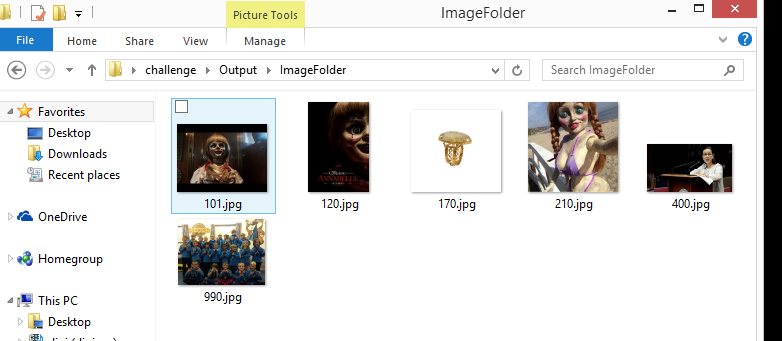
Generated real time streaming output is sent to a file

1. Modify the file printerbolt.java
2. Write the file writer code to extract the tuples of tweets to a file.
3. Provide the exact path in the code so that this can be used to analyze the tweets



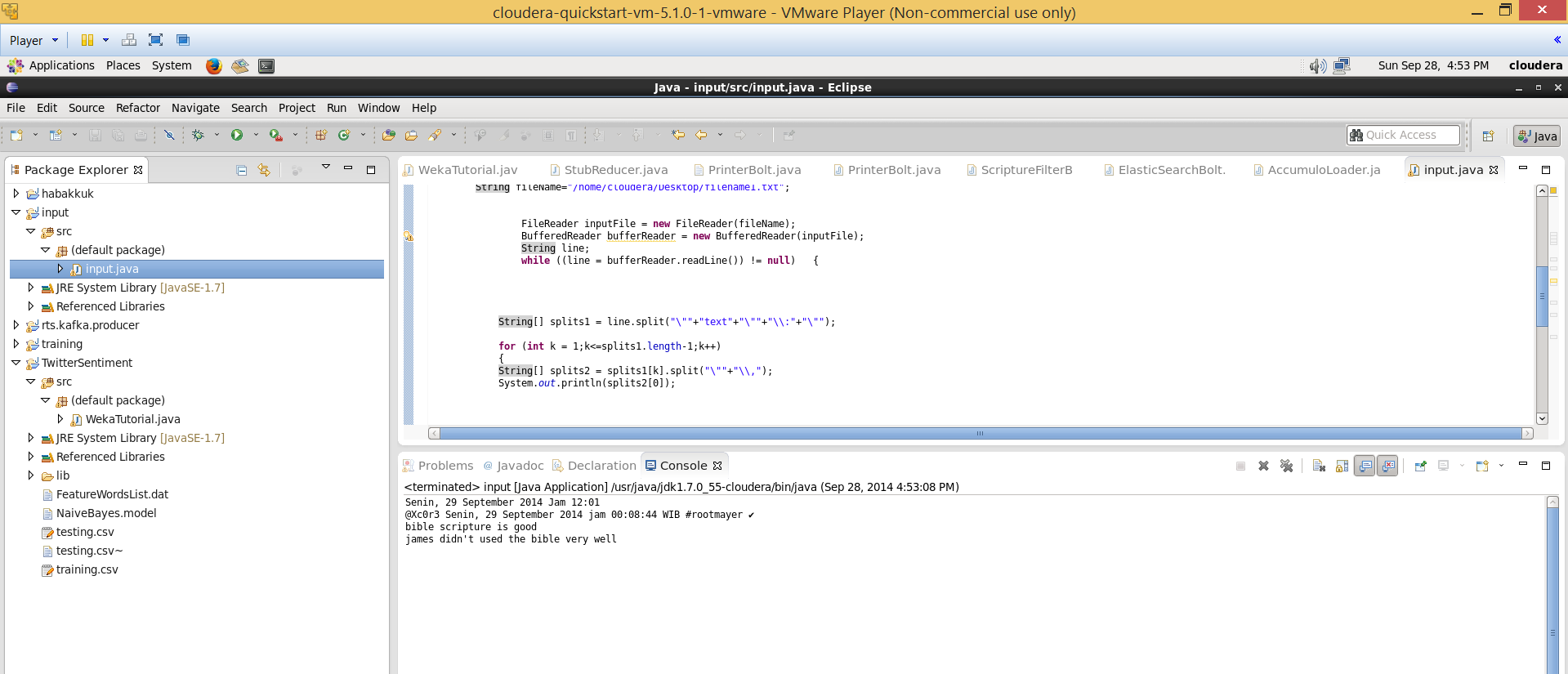
Generated real time streaming output IMAGES is sent to a folder

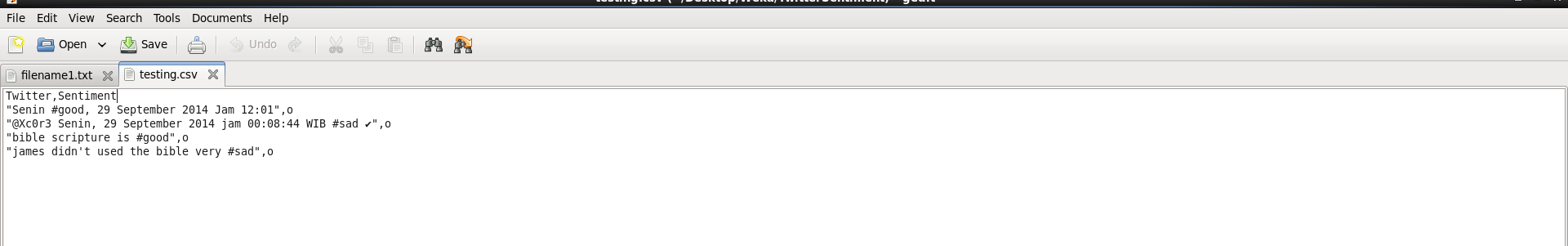
Below are the images of Annabelle movie tweets generated.



Now the tweet text messages are collected and sent to the csv file

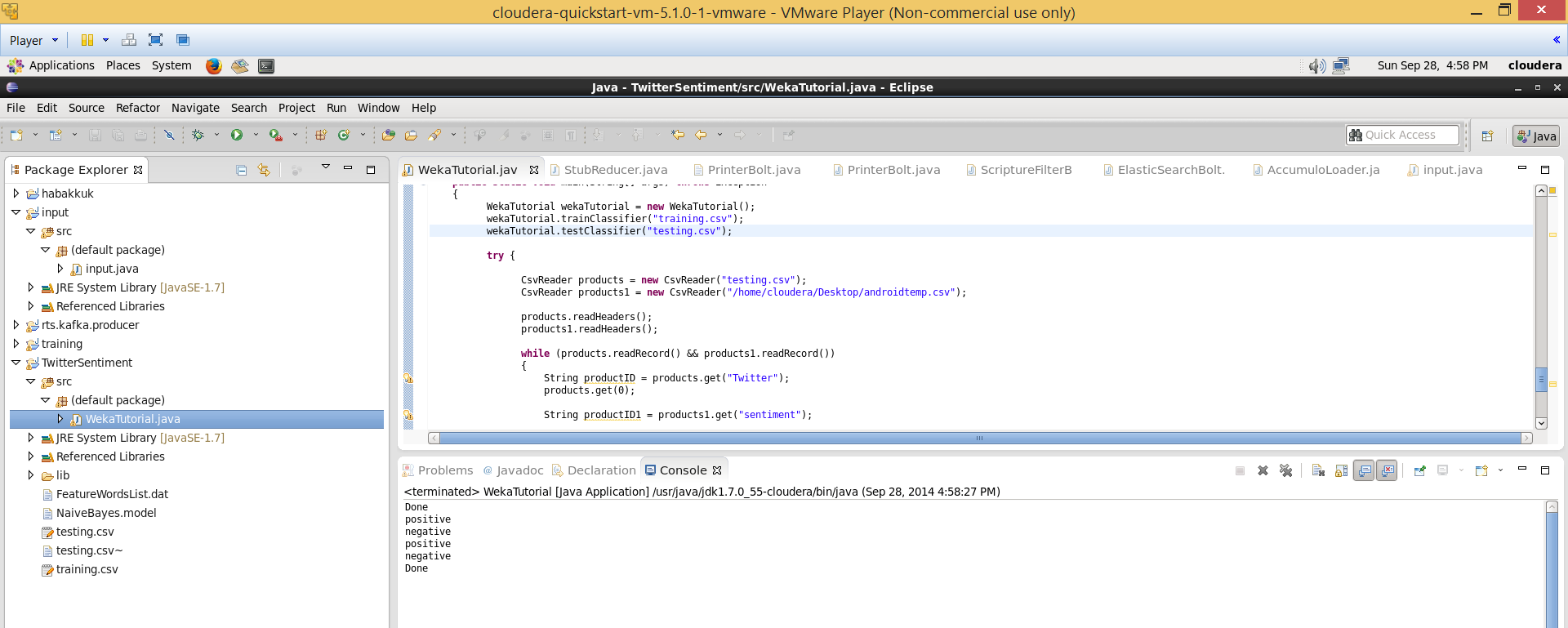
1. In order to analyze the tweets, we need to convert text file of tweets to csv format.
2. We need to append any special character at the end of each tweet.
3. This format must be followed in order to analyze data with training data.





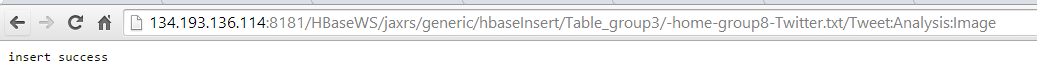
Now run the Analysis algorithm over this generated data

1. The generated csv file is fed to the twitter sentiment code
2. Here we uses weka jar to use the naïve bayes algorithm so that positive negative tweets can be identified over this.
3. Each tweet is analyzed and displayed on the console.



The tweets analysis is sent to hbase and retrieved.

1. Apart from this, the analyzed part is stored in to Hbase along with their corresponding tweets.
2. This file can be used for showing analysis and visualization to the user by using android application .

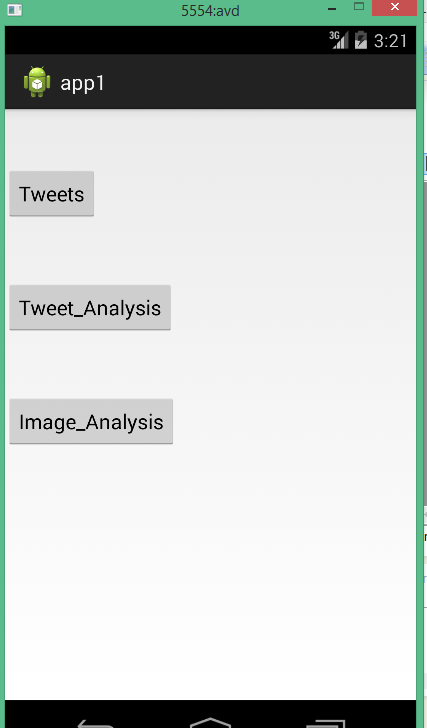


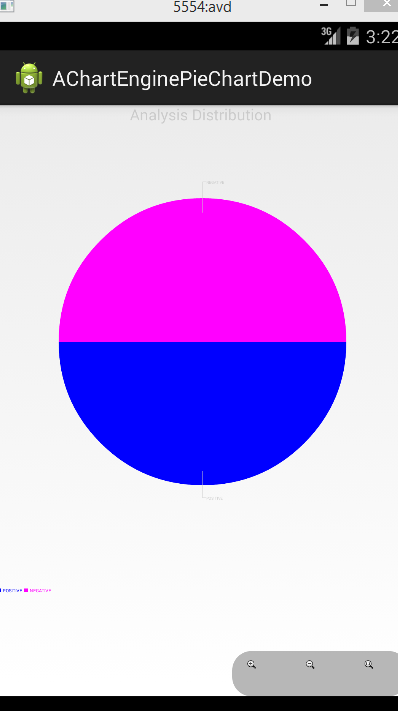




Android:

Now run the android app on the generated tweets





1. Website URL’s:

<http://blog.csdn.net/hljlzc2007/article/details/13275441>

<http://gramatica.usc.es/~gamallo/artigos-web/TASS2013.pdf>

<https://www.google.com/search?hl=en&site=imghp&tbm=isch&source=hp&biw=1536&bih=755&q=twitter+logo&oq=twitter+l&gs_l=img.3.0.0l10.1103.3765.0.4721.9.8.0.1.1.0.253.934.5j2j1.8.0....0...1ac.1.54.img..0.9.933.IpKE-td4Uzc#facrc=_&imgdii=_&imgrc=Z60PgD9iMKtm5M%253A%3BuY8NrGIitp5hnM%3Bhttp%253A%252F%252Floopele.com%252Fwp-content%252Fuploads%252F2014%252F07%252Ftwitter-logo.jpg%3Bhttp%253A%252F%252Floopele.com%252Ftwitter-logo%252F%3B1024%3B831>

<http://www.csvreader.com/java_csv_samples.php>

<http://stackoverflow.com/questions/2346713/csv-row-reader-question>

<https://blackboard.umkc.edu/webapps/blackboard/content/listContent.jsp?course_id=_114034_1&content_id=_1371866_1&mode=reset>

<http://www.java2s.com/Code/Jar/j/Downloadjavacsvjar.htm>

1. GitHub URL:

https://github.com/lbzv9/RA\_Challege2

1. Limitations:

We couldn’t able to judge the images and analyze them properly.

1. References:

<http://blog.csdn.net/hljlzc2007/article/details/13275441>

<http://gramatica.usc.es/~gamallo/artigos-web/TASS2013.pdf>

<https://www.google.com/search?hl=en&site=imghp&tbm=isch&source=hp&biw=1536&bih=755&q=twitter+logo&oq=twitter+l&gs_l=img.3.0.0l10.1103.3765.0.4721.9.8.0.1.1.0.253.934.5j2j1.8.0....0...1ac.1.54.img..0.9.933.IpKE-td4Uzc#facrc=_&imgdii=_&imgrc=Z60PgD9iMKtm5M%253A%3BuY8NrGIitp5hnM%3Bhttp%253A%252F%252Floopele.com%252Fwp-content%252Fuploads%252F2014%252F07%252Ftwitter-logo.jpg%3Bhttp%253A%252F%252Floopele.com%252Ftwitter-logo%252F%3B1024%3B831>

<http://www.csvreader.com/java_csv_samples.php>

<http://stackoverflow.com/questions/2346713/csv-row-reader-question>

<https://blackboard.umkc.edu/webapps/blackboard/content/listContent.jsp?course_id=_114034_1&content_id=_1371866_1&mode=reset>

<http://www.java2s.com/Code/Jar/j/Downloadjavacsvjar.htm>