SAN JOSÉ STATE
UNIVERSITY

Fall-2013

Launch Product and Feedback Analysis

CMPE-239 Project

Submitted to Prof. Magdalini Eirinaki

November 25, 2013

By

Bhavya Chauhan - 008678735
Harsh Shah - 008729149
Parin Shah - 008335561

Table of Contents

Introduction

Motivation

Use of social media has increased significantly since past 5 years and recent calculations before 4 months show that 25% of world's population use social media to express their views.[1] With such a large amount of people using social media, opinions or sentiments of different products is likely to be available on social media. So, our motivation for this project is to get an idea of sentiments of people across the world for a product before launch so that the release location of the product can be decided according to the sentiments in different continents. And further, we can get an idea of sentiments about a product after launch for further advancements. This business logic can help any company to launch the product as per demand.

Objective

Motive of this project is to stream/search the data on social media using specific set of keywords related to product and then segregate the data according to continents. Then further the data obtained according to continents is classified into three kinds of sentiments : Positive, Negative and Neutral. Then ultimately the result is shown continent wise using Google charts.

System Design

Architecture

We are using Twitter4J library published by twitter to fetch tweets containing keywords related to product. These tweets are fetched in excel sheet. The streaming runs independently from all other programs so that live analysis can be performed.

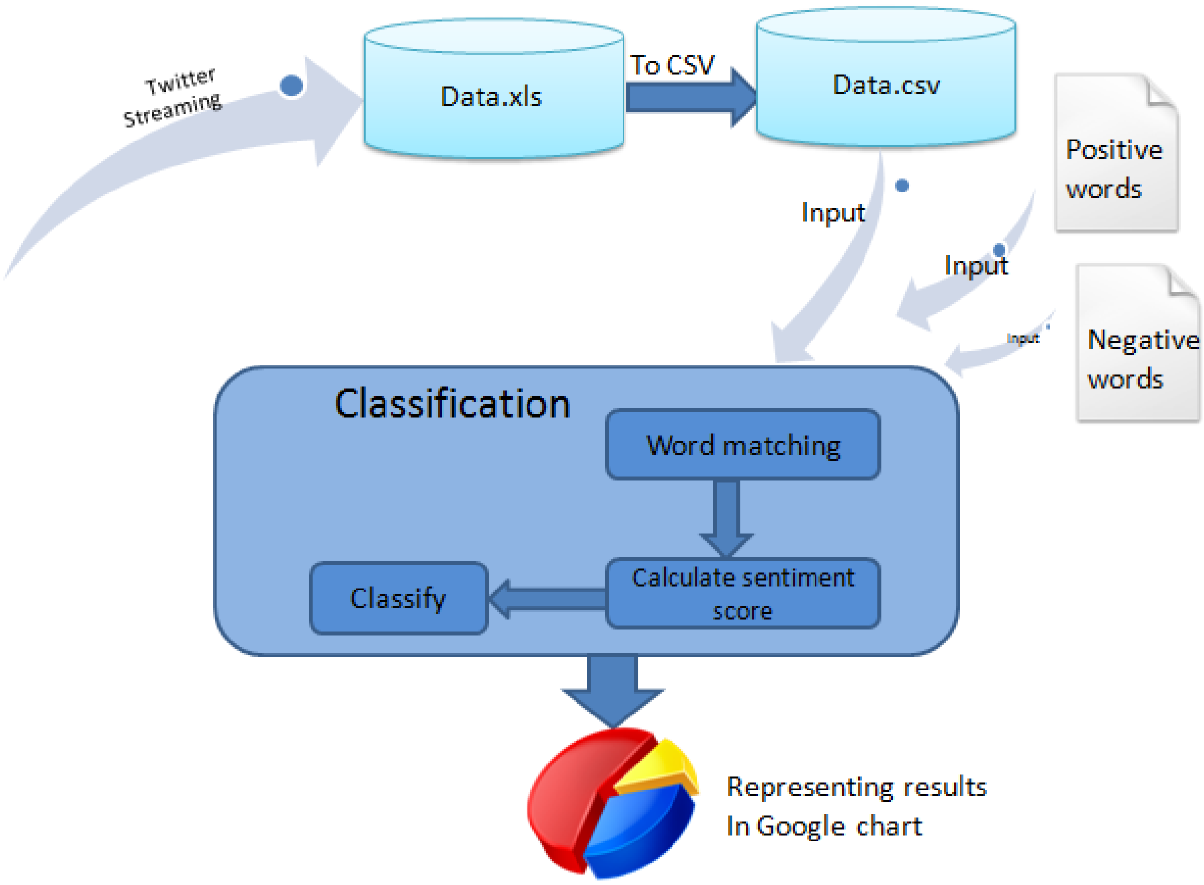


Figure-1 : System Architecture

Implementation details

Streaming from Twitter :

We are using Twitter4J library for streaming product related tweets from Twitter using Java. We needed to create a Twitter application on https://dev.twitter.com// We input four credentials : Consumer key, Consumer Secret Key, Access Token and Access Token Secret Key to access twitter APIs. The corresponding java code is written in Streaming.java file and Code snippet for the same is as follows

```
ConfigurationBuilder cb = new ConfigurationBuilder();
        cb.setDebugEnabled(true);
        cb.setOAuthConsumerKey("<Consumer Key");
        cb.setOAuthConsumerSecret("<Consumer Secret Key>");
        cb.setOAuthAccessToken("<Access token>");
        cb.setOAuthAccessTokenSecret("<Access token Secret>");
final TwitterStream twitterStream = new
TwitterStreamFactory(cb.build()).getInstance();
```

We are storing content of the tweet and Continent of the user who tweeted. We wanted to fetch exact location of the tweet but twitter APIs provide the location entered by the User which may have absurd values. So we decided to fetch geographic location of the tweet and then we derived its continent based on hard-coded values of latitudes and longitudes of all 7 continents.

Also note that we are pre-processing / cleaning tweets while streami**ng. i.e. we are removing emotico**ns or smiley contents like ":)" , ":(", etc.. We are providing product related keywords to stream related tweets only whose code snippet is as follows :

```
FilterQuery fq = new FilterQuery();
String[] keywords = {"nexus 5", "nexus5"};
String st[]={"en","english"};
fq.language(st);
fq.track(keywords);
twitterStream.addListener(listener);
twitterStream.filter(fq);
```

The above mentioned Streaming program is independent to all the program and no other program has dependency on this program. Because, we are writing the streamed data to separate file and we are accessing that file further to analyze content in order to provide run-time and live analysis. For gathering previous tweets to use as a data-set, we used search API of Twitter. And its code snippet is as follows :

```java
for (int i = 0; i < tweets.size(); i++) {

      String content = "";
      String contn = "";

      Status status = (Status) tweets.get(i);

      HSSFRow row = sheet.createRow(rowcount);
      HSSFCell cell = row.createCell(0);
      HSSFCell cell1 = row.createCell(1);

      cell.setCellType(HSSFCell.CELL_TYPE_STRING);
      cell1.setCellType(HSSFCell.CELL_TYPE_STRING);

      try{
             content = status.getText();
             String tmpcontent = content;
             tmpcontent = tmpcontent.replaceAll("[^\\dA-Za-z ]",
"").replaceAll("\\s+", " ");
             content = tmpcontent.replace("\"", " ");
             double lon = status.getGeoLocation().getLongitude();
              double lat = status.getGeoLocation().getLatitude();
             double ASLAL = 11.62, ASLAU = 81.86, ASLOL = 27.33, ASLOU = 169.02;
             double AULAL = -48, AULAU = -4, AULOL = 112, AULOU = 180;
             double AFLAL = -35, AFLAU = 35, AFLOL = -15, AFLOU = 52;
             double NALAL = 5, NALAU = 85 , NALOL = -165, NALOU =-15;
             double SALAL = -55, SALAU =12  , SALOL =-84   , SALOU = -35 ;
             double ELAL = 35, ELAU = 80 , ELOL = -10  , ELOU = 65 ;

             if(status.getGeoLocation()!=null){

               if(lat <= ASLAU && lat >=ASLAL && lon <=ASLOU && lon >= ASLOL)
                    contn="Asia";
               else if(lat <= AULAU && lat >=AULAL && lon <=AULOU && lon >= AULOL)
                    contn="Australia";
               else if(lat <= AFLAU && lat >=AFLAL && lon <=AFLOU && lon >= AFLOL)
                    contn="Africa";
               else if(lat <= NALAU && lat >=NALAL && lon <=NALOU && lon >= NALOL)
                    contn="NorthAmerica";
               else if(lat <= SALAU && lat >=SALAL && lon <=SALOU && lon >= SALOL)
                    contn="SouthAmerica";
               else if(lat <= ELAU && lat >=ELAL && lon <=ELOU && lon >= ELOL){
                    contn="Europe";
               else{
                    contn = "NULL";
             }
             else{
                    contn = "NULL";
             }
```

```
        }catch(NullPointerException e){
                e.printStackTrace();
        }

        cell.setCellValue(content);
        cell1.setCellValue(contn);
        rowcount++;

        try {
                out = new FileOutputStream(f);
                workbook.write(out);
                out.flush();
                out.close();
        } catch (IOException e) {
                e.printStackTrace();
        }
}
```

Converting data file to CSV

After streaming data in excel sheet, we convert the data to CSV file format. We are doing this conversation of file convert() in ToCSV.java.

Running R scripts from Java

After converting data file from excel to CSV, we run the runscript() from RunR.java which sets certain parameters executable files that can be used to run R scripts. the code snippet for the same is as follows:

```
public static void runscript() {
        try {
            RCaller caller = new RCaller();
            caller.setRscriptExecutable("path/bin/x64/Rscript.exe");
            caller.setRExecutable("path/bin/x64/Rscript.exe");
            caller.cleanRCode();
            Runtime.getRuntime().exec("path/bin/x64/Rscript.exe 
E:/Workspace/TwitterStreamPlot/src/Source/run.R");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
```

Calculating sentimental scores

Note that the last line in the try block of above mentioned code snippet indicates execution of run.R file which contains code to calculate sentiment score of each tweet. For this, a file containing positive sentimental words and negative sentimental words are given as input. Tweets from csv are read and converted to vector and each word in the tweet is compared with all the words in the words of positive and negative sentimental word files. So, basically, it's a word matching process of each word of tweet with that of positive and negative word-file. Each time the sentiment score is incremented if word is found in file of positive words and vice versa.

Classify tweet scores in positive, negative and neutral classes

After calculating sentimental scores for each tweet and storing them in results.csv file, we fetch these results and classify the sentiment of each tweet in three classes : Positive, Negative and Neutral. Simultaneously, we segregate them into six different continents and seventh as null (in case

```
f  = new File("src/Source/result.csv");
BufferedReader mycsv;
String cvsSplitBy = ",";
String line = "";
try {
        mycsv = new BufferedReader(new FileReader(f));
        while ((line = mycsv.readLine()) != null) {
                data = line.split(cvsSplitBy);
                if(data[3].equals("\"NorthAmerica\"")){
                        increment(0); // Increment sentiment scores in 0th
row of 2D array
                }
                else if(data[3].equals("\"SouthAmerica\"")){
                        increment(1);
                }
                else if(data[3].equals("\"Asia\"")){
                        increment(2);
                }
                else if(data[3].equals("\"Australia\"")){
                        increment(3);
                }
                else if(data[3].equals("\"Africa\"")){
                        increment(4);
                }
                else if(data[3].equals("\"Europe\"")){
                        increment(5);
                }
                else if(data[3].equals("\"NULL\"")){
                        increment(6);
                }
        }
        mycsv.close();

        } catch (IOException e) {
        e.printStackTrace();
        }

        return ana;}

private static void increment(int n){
                if(Integer.parseInt(data[1]) > 0)
                        ana[n][0]++;
                else if(Integer.parseInt(data[1]) < 0)
                        ana[n][2]++;
                else
                        ana[n][1]++;
        }
```

no continent is found due to lack of geolocation.) We perform this classification in fetchresult() in Score.java file. The Java code-snippet for the same is as follows:

The result of sentimental analysis is stored in 2D array with rows as continent and column as sentiment classes. This result is printed on console and then the program of drawing Pie-charts using Google-charts API starts its execution.

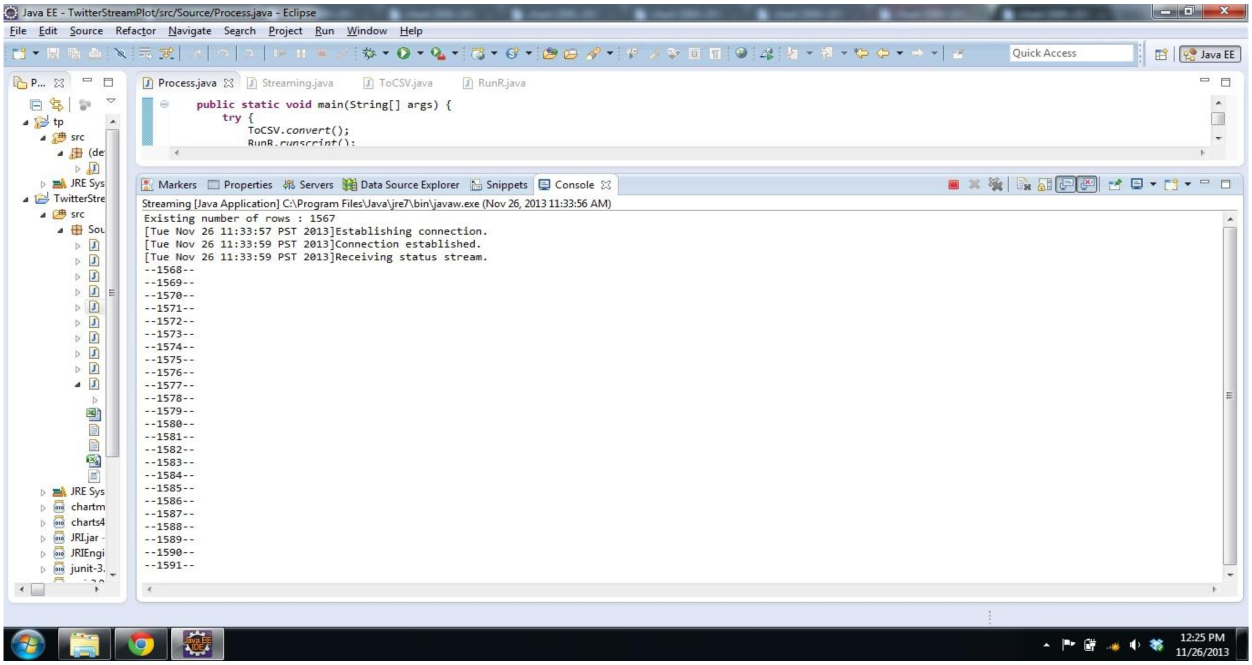
Using Google Chart to represent the result

We are using Google chart APIs to represent the positive, negative and neutral sentiments for each continent. We are using com.googlecode.charts4j library for this purpose. Corresponding code snippets to display a chart in browser is as follows :

```
public static void makeChart(int[] global, String title) {

      int total = global[0] + global[1] + global[2];

      int[] perc = {(int)((global[0]*100)/total),
      (int)((global[1]*100)/total), (int)((global[2]*100)/total)};

      Slice s1 = Slice.newSlice(perc[0], Color.newColor("3366CC"),
Integer.toString(perc[0]) + "%", "Positive");
      Slice s2 = Slice.newSlice(perc[1], Color.newColor("990099"),
Integer.toString(perc[1]) + "%", "Neutral");
      Slice s3 = Slice.newSlice(perc[2], Color.newColor("DD4477"),
Integer.toString(perc[2]) + "%", "Negative");

      PieChart chart = GCharts.newPieChart(s1, s2, s3);
      chart.setTitle(title, BLACK, 16);
      chart.setSize(500, 200);
      chart.setThreeD(true);
      String myURL = chart.toURLString();

       BareBonesBrowserLaunch.openURL(myURL);
}
```

```java
public static void global(int[] global, String title){

      int total = global[0] + global[1] + global[2] + global[3] + global[4]
+ global[5] + global[6];

      int[] perc = {(int)((global[0]*100)/total),
      (int)((global[1]*100)/total), (int)((global[2]*100)/total),
      (int)((global[3]*100)/total), (int)((global[4]*100)/total),
      (int)((global[5]*100)/total), (int)((global[6]*100)/total)};

      String[] contnt = {"NorthAmerica", "SouthAmerica", "Asia",
"Australia", "Africa", "Europe", "Geolocation not specified"};

      Slice s1 = Slice.newSlice(perc[0], Color.newColor("3366CC"),
Integer.toString(perc[0]) + "%", contnt[0]);
      Slice s2 = Slice.newSlice(perc[1], Color.newColor("990099"),
Integer.toString(perc[1]) + "%", contnt[1]);
      Slice s3 = Slice.newSlice(perc[2], Color.newColor("DD4477"),
Integer.toString(perc[2]) + "%", contnt[2]);
      Slice s4 = Slice.newSlice(perc[3], Color.newColor("006400"),
Integer.toString(perc[3]) + "%", contnt[3]);
      Slice s5 = Slice.newSlice(perc[4], Color.newColor("2E0854"),
Integer.toString(perc[4]) + "%", contnt[4]);
      Slice s6 = Slice.newSlice(perc[5], Color.newColor("E6693E"),
Integer.toString(perc[5]) + "%", contnt[5]);
      Slice s7 = Slice.newSlice(perc[6], Color.newColor("F3B49F"),
Integer.toString(perc[6]) + "%", contnt[6]);

      PieChart chart = GCharts.newPieChart(s1, s2, s3, s4, s5, s6, s7);

      chart.setTitle(title + " Tweets over the Globe ", BLACK, 16);
      chart.setSize(500, 200);
      chart.setThreeD(true);
      String myURL = chart.toURLString();

      BareBonesBrowserLaunch.openURL(myURL);

    }
```

Screenshots

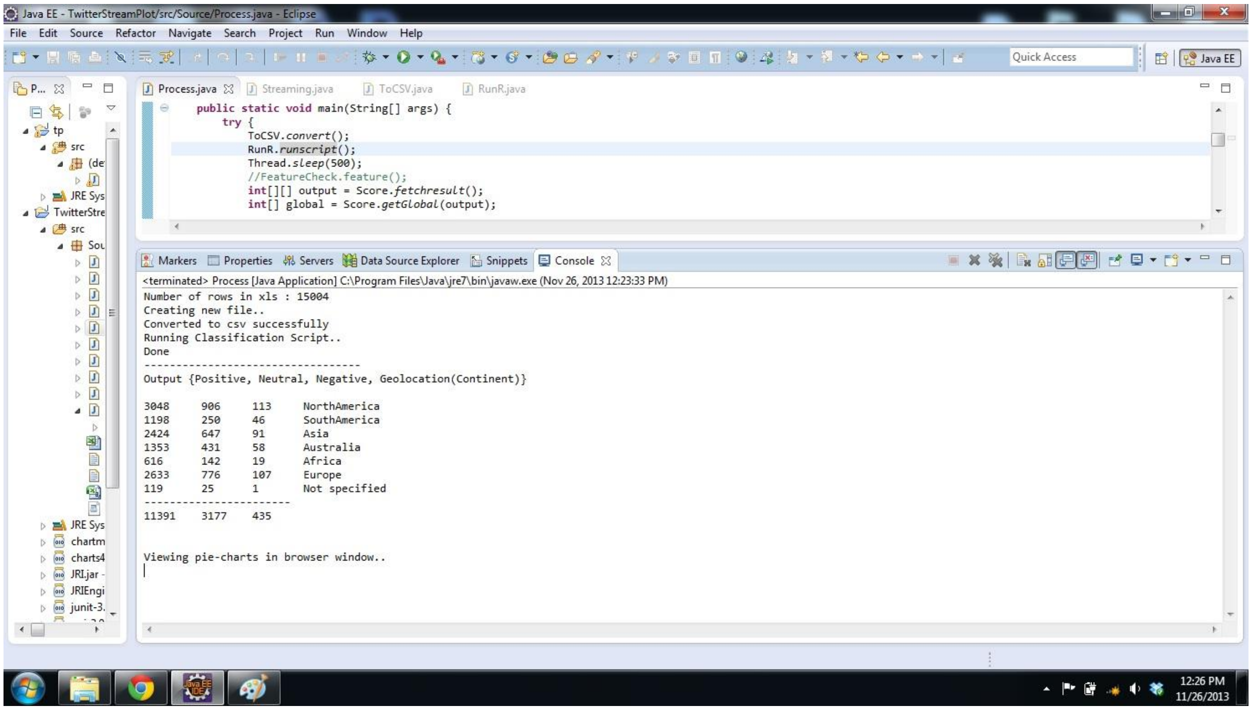


```
    public static void main(String[] args) {
        try {
            ToCSV.convert();
            RunR.runscript();
```

Streaming [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (Nov 26, 2013 11:33:56 AM)

```
Existing number of rows : 1567
[Tue Nov 26 11:33:57 PST 2013]Establishing connection.
[Tue Nov 26 11:33:59 PST 2013]Connection established.
[Tue Nov 26 11:33:59 PST 2013]Receiving status stream.
--1568--
--1569--
--1570--
--1571--
--1572--
--1573--
--1574--
--1575--
--1576--
--1577--
--1578--
--1579--
--1580--
--1581--
--1582--
--1583--
--1584--
--1585--
--1586--
--1587--
--1588--
--1589--
--1590--
--1591--
```

Figure 2 - Streaming Tweets



```
    public static void main(String[] args) {
        try {
            ToCSV.convert();
            RunR.runscript();
            Thread.sleep(500);
            //FeatureCheck.feature();
            int[][] output = Score.fetchresult();
            int[] global = Score.getGlobal(output);
```

<terminated> Process [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (Nov 26, 2013 12:23:33 PM)

```
Number of rows in xls : 15004
Creating new file..
Converted to csv successfully
Running Classification Script..
Done
----------------------------------
Output {Positive, Neutral, Negative, Geolocation(Continent)}

3048    906     113     NorthAmerica
1198    250     46      SouthAmerica
2424    647     91      Asia
1353    431     58      Australia
616     142     19      Africa
2633    776     107     Europe
119     25      1       Not specified
-----------------------
11391   3177    435


Viewing pie-charts in browser window..
```

Figure 3 - Console Output

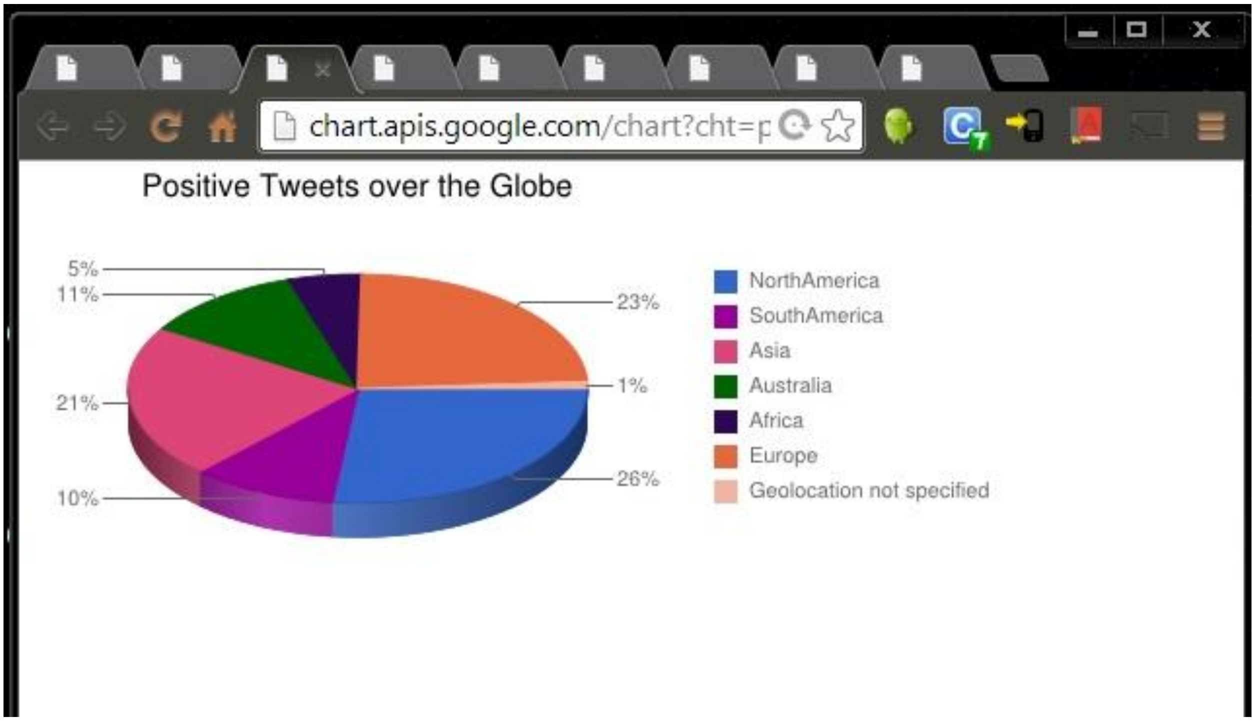


chart.apis.google.com/chart?cht=p
Positive Tweets over the Globe
5%
11%
23%
1%
21%
26%
10%
NorthAmerica
SouthAmerica
Asia
Australia
Africa
Europe
Geolocation not specified

Figure 4 - Tweet Analysis for Positive tweets

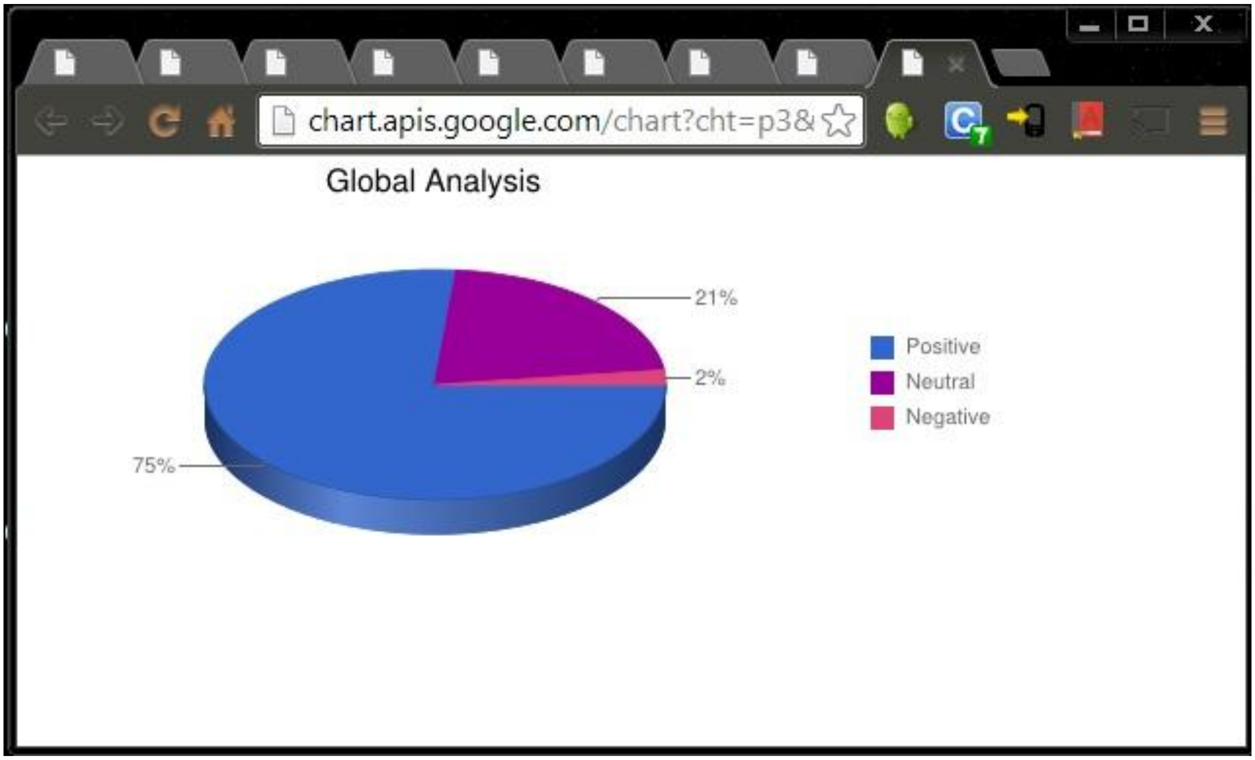


Figure 5 - Global Analysis

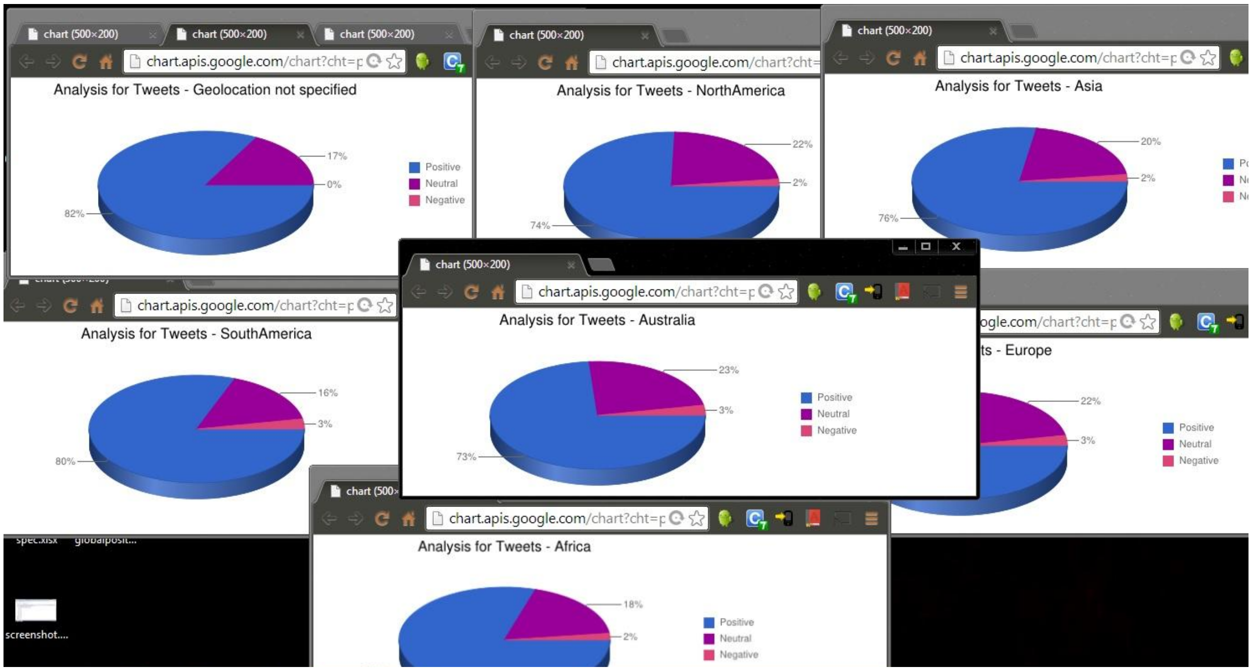


Figure 6 - Tweet Analysis for Each Continent

References

1. http://www.emarketer.com/Article/Social-Networking-Reaches-Nearly-One-Four-Around-World/1009976
2. Sentiment Analysis and Subjectivity By Bing Liu, Department of Computer Science, University of Illinois at Chicago
3. http://twitter4j.org/javadoc/index.html
4. https://code.google.com/p/charts4j/