INFO 7250: BIG DATA ENGINEERING

PROJECT REPORT

Universal Rating Analysis using Social Media

Amulya Aankul

Disha Akarte

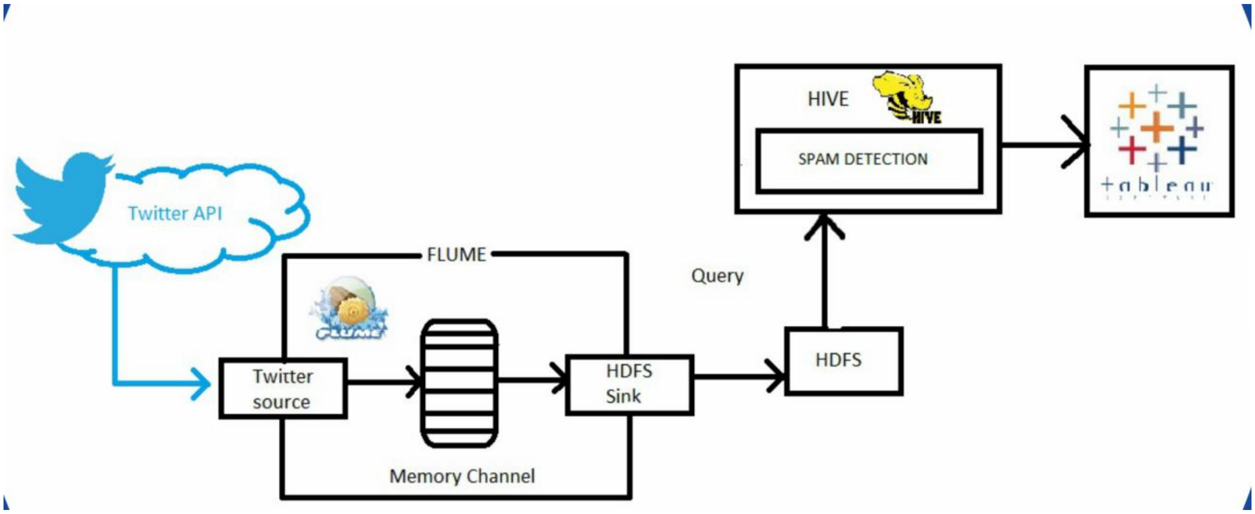
Vaishnavi Nampally

Universal Rating Analysis using Social Media

I. Introduction:

Microblogging websites have evolved to become a source of varied kind of information. This is due to nature of microblogs on which people post real time messages about their opinions on a variety of topics, discuss current issues, complain, and express positive sentiment for products they use in daily life. In fact, companies manufacturing such products have started to poll these microblogs to get a sense of general sentiment for their product. Many times these companies study user reactions and reply to users on microblogs. One challenge is to build technology to detect and summarize an overall sentiment to create a Universal Rating System. We first filter tweets according to the hashtags and then compare each word in the tweet with the list of positive and negative words in the dictionary look up. We then summarize the output to create a universal rating of that particular hashtags.

Performing Sentiment Analysis on Twitter is trickier than doing it for large reviews. This is because the tweets are very short (only about 140 characters) and usually contain slangs, emoticons, hash tags and other twitter specific jargon. For the development purpose twitter provides streaming API which allows the developer an access to 1% of tweets tweeted at that time bases on the particular keyword. The object about which we want to perform sentiment analysis is submitted to the twitter API's which does further mining and provides the tweets related to only that object. Twitter data is generally unstructured i.e use of abbreviations is very high. Also it allows the use of emoticons which are direct indicators of the author's view on the subject. Tweet messages also consist of a timestamp and the user name. This timestamp is useful for guessing the future trend application of our project. For doing twitter data analysis first data is collected using FLUME in local HDFS. Tweets are preprocesses for removing noise, meaningless symbols and spams.

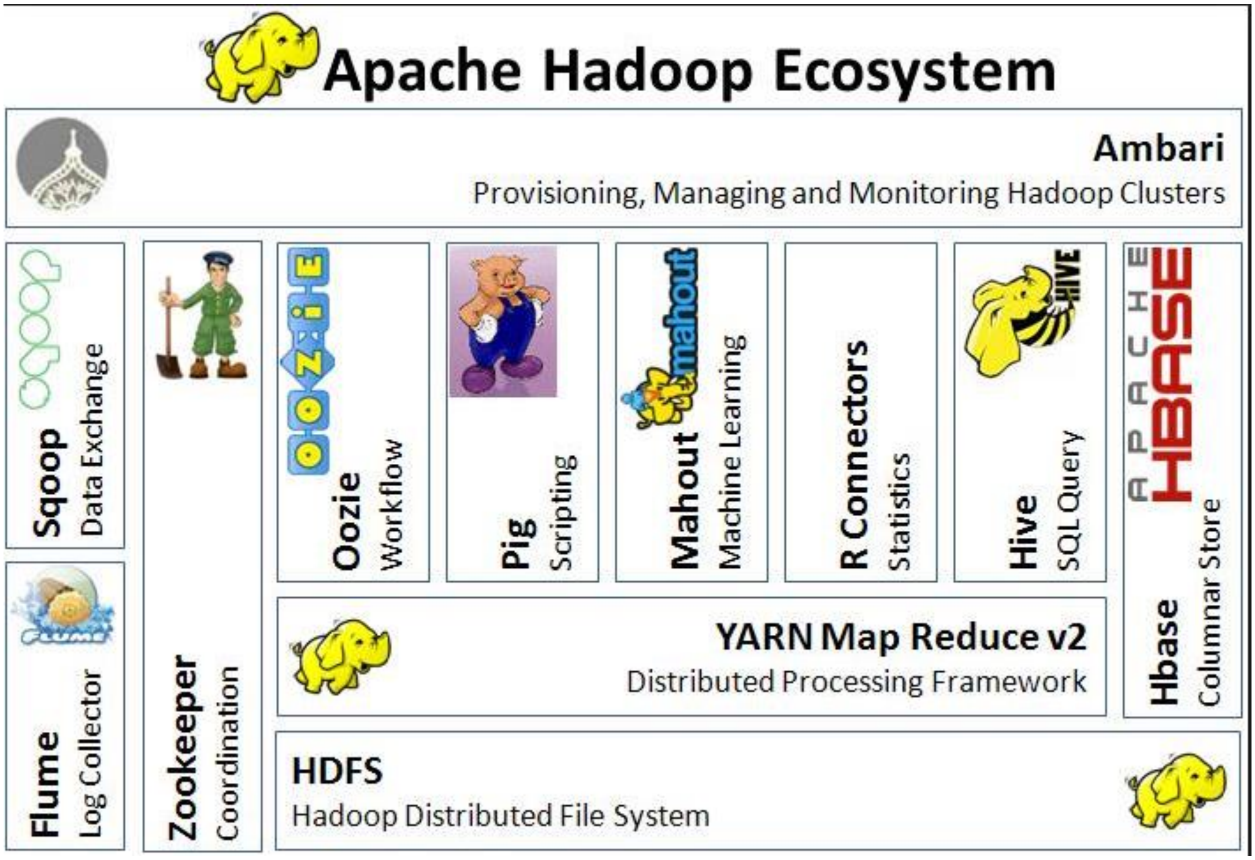


Installation and workflow

II. HADOOP

Apache Hadoop is good choice for twitter analysis as it works for distributed big data. Apache Hadoop is an open source software framework for distributed storage and large scale distributed processing of data-sets on clusters. Hadoop runs applications using the MapReduce algorithm, where the data is processed in parallel on different CPU nodes. In short, Hadoop framework is capable enough to develop applications capable of running on clusters of computers and they could perform complete statistical analysis for a huge amounts of data. Hadoop MapReduce is a software framework for easily writing applications which process big amounts of data in-parallel on large clusters (thousands of nodes) of commodity hardware in a reliable, fault-tolerant manner.

Hadoop framework includes different modules like MapReduce, Flume, Hive, Pig, Sqoop, Oozie, Zookeeper, Hbase for different functionality as shown in below diagram. I will be using FLUME and HIVE for twitter analysis.



Hadoop use HDFS (Hadoop Distributed File System) file system. The Hadoop Distributed File System (HDFS) is based on the Google File System (GFS) and provides a distributed file system that is designed to run on large clusters (thousands of computers) of small computer machines in a reliable, fault-tolerant manner. HDFS uses a master/slave architecture where master consists of a single Name Node that manages the file system metadata and one or more slave Data Nodes that store the actual data. Benefit of using Hadoop is distributed storage, Distributed Processing, Security, Reliability, Speed, Efficiency, Availability, Scalability and lots more. This is the reason of using Hadoop for tweet processing.

III. FLUME

Apache Flume is a distributed, reliable, and available service for efficiently collecting, aggregating, and moving large amounts of streaming data into the Hadoop Distributed File System (HDFS). It can be used for dumping twitter data in Hadoop HDFS. After the installation of VMWRE and Hadoop for single node next step come the installation of FLUME. For this you need to log in to twitter. After that go to apps on twitter and create an new application. After you agree with all terms and conditions you will got new application. Then set Consumer Key, Consumer Secret, Owner Key and Owner Secret ID. Now access token need to be created. After the creation of access token and refresh you will get all the 4 information. Now you Go to flume home and download Apache Flume. Download the flume-sources-1.0-SNAPSHOT.jar and add it to the flume class path as shown below in the conf/flume-env.sh file

```
FLUME_CLASSPATH="/home/training/Installations/apache-flume-1.3.1-bin/flume-sources-1.0-SNAPSHOT.jar"
```

This will automatically be dumped in downloads. Store in desired library. You need to go to apace flume, then go to downloads and extract here and place it In bin and lib. After this you need to configure the file flume.conf . The flume.conf should have all the agents defined as below:

flume.conf

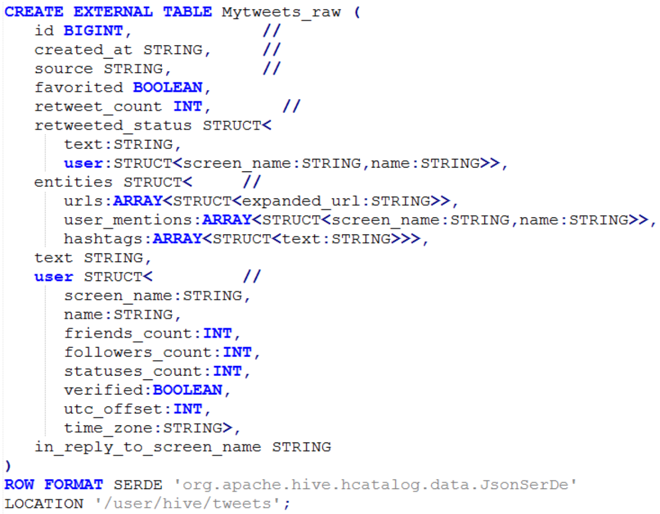
While configuration of this file configure sink as HDFS and Set path for storing tweets in HDFS. Run the configuration file and tweets start downloading in HDFS in specified path. To do this execute flume comments. Start flume using the below command:

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

After a couple of minutes the Tweets should appear in HDFS. If no tweet downloaded in the specified path then refresh. Temporarily data remain in container / Channel and In few seconds tweets start dumping in HDFS. The data downloaded in HDFS is in JSON format. That need to be converted into readable format. Add jsonserde.jar File to convert Json data in readable format.

IV. HIVE

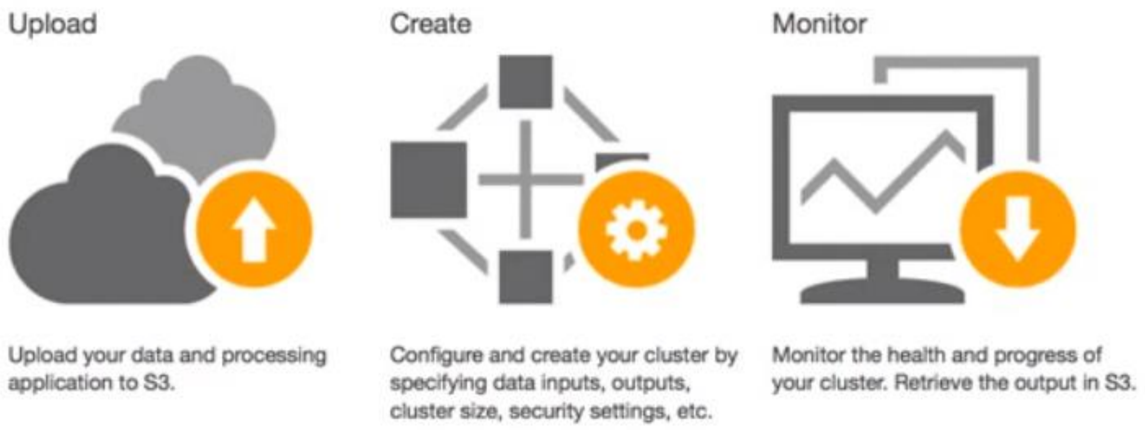
Hive is a data warehouse infrastructure tool to process structured data in Hadoop. It resides on top of Hadoop to summarize Big Data, and makes querying and analyzing easy. Apache Hive (HiveQL) with Hadoop Distributed file System is used for Analysis of data. Hive provides a SQL-like interface to process data stored in HDP. Due its SQL-like interface, Hive is increasingly becoming the technology of choice for using Hadoop. To set up HIVE in Hadoop : Build or Download the JSON SerDe Before we can query the data, we need to ensure that the Hive table can properly interpret the JSON data. By default, Hive expects that input files use a delimited row format, but our Twitter data is in a JSON format, which will not work with the defaults. And we can use the Hive SerDe interface to specify how to interpret what we've loaded. SerDe stands for Serializer and Deserializer, which are interfaces that tell Hive how it should translate the data into something that Hive can process.



```
CREATE EXTERNAL TABLE Mytweets_raw (
   id BIGINT,              //
   created_at STRING,      //
   source STRING,          //
   favorited BOOLEAN,
   retweet_count INT,        //
   retweeted_status STRUCT<
      text:STRING,
      user:STRUCT<screen_name:STRING,name:STRING>>,
   entities STRUCT<     //
      urls:ARRAY<STRUCT<expanded_url:STRING>>,
      user_mentions:ARRAY<STRUCT<screen_name:STRING,name:STRING>>,
      hashtags:ARRAY<STRUCT<text:STRING>>>,
   text STRING,
   user STRUCT<          //
      screen_name:STRING,
      name:STRING,
      friends_count:INT,
      followers_count:INT,
      statuses_count:INT,
      verified:BOOLEAN,
      utc_offset:INT,
      time_zone:STRING>,
   in_reply_to_screen_name STRING
)
ROW FORMAT SERDE 'org.apache.hive.hcatalog.data.JsonSerDe'
LOCATION '/user/hive/tweets';
```

V. HARDWARE

Since the size of data depends upon the number of people tweeting about the topic. Scalability is a big factor here. So we decided to implement this in Amazon Web Services. We used Fully Distributed to run FLUME and HIVE. The following are the hardware specifications of the AWS cluster:



Upload

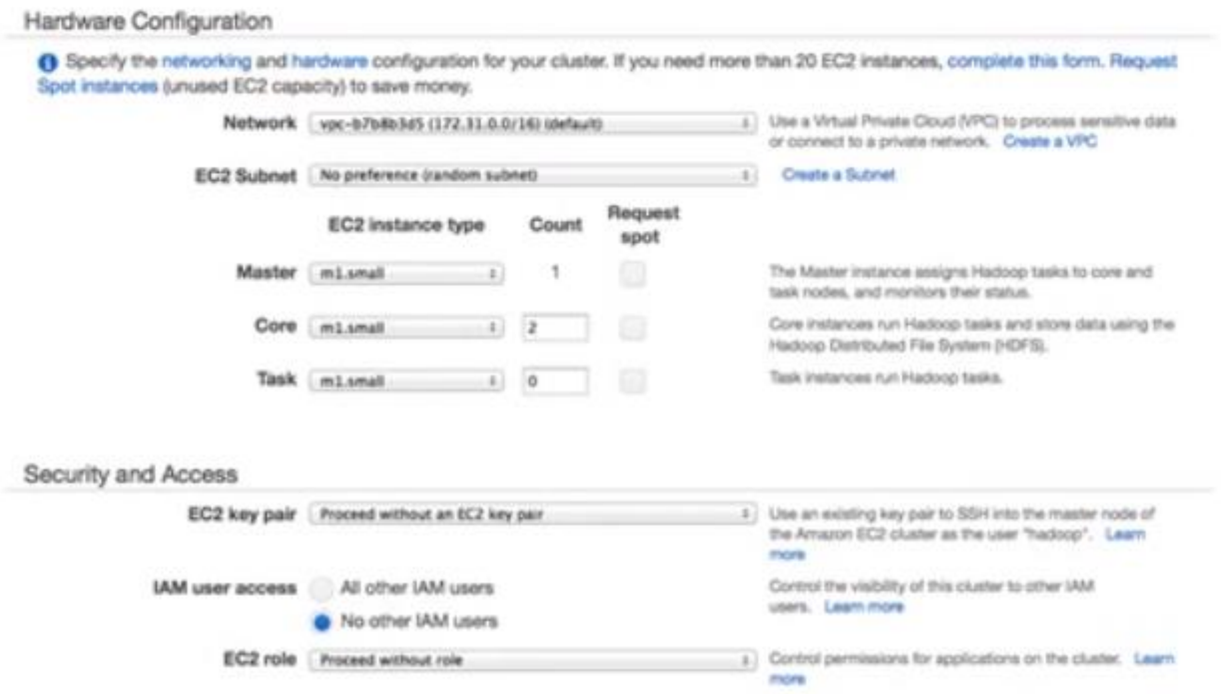
Upload your data and processing application to S3.

Create

Configure and create your cluster by specifying data inputs, outputs, cluster size, security settings, etc.

Monitor

Monitor the health and progress of your cluster. Retrieve the output in S3.



Hardware Configuration

Specify the networking and hardware configuration for your cluster. If you need more than 20 EC2 instances, complete this form. Request Spot instances (unused EC2 capacity) to save money.

Network: vpc-b7b8b3d5 (172.31.0.0/16) (default) — Use a Virtual Private Cloud (VPC) to process sensitive data or connect to a private network. Create a VPC

EC2 Subnet: No preference (random subnet) — Create a Subnet

	EC2 instance type	Count	Request spot	
Master	m1.small	1	□	The Master instance assigns Hadoop tasks to core and task nodes, and monitors their status.
Core	m1.small	2	□	Core instances run Hadoop tasks and store data using the Hadoop Distributed File System (HDFS).
Task	m1.small	0	□	Task instances run Hadoop tasks.

Security and Access

EC2 key pair: Proceed without an EC2 key pair — Use an existing key pair to SSH into the master node of the Amazon EC2 cluster as the user "hadoop". Learn more

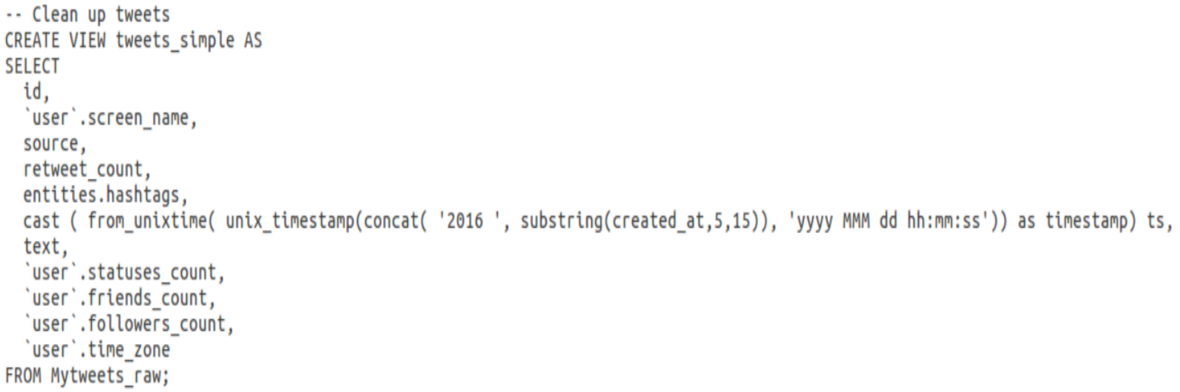
IAM user access: ○ All other IAM users ● No other IAM users — Control the visibility of this cluster to other IAM users. Learn more

EC2 role: Proceed without role — Control permissions for applications on the cluster. Learn more

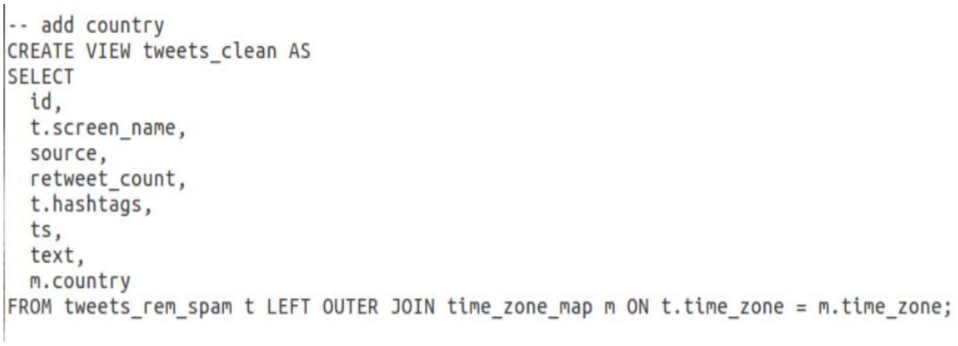
VI. CLEANING OF TWITTER DATA

Twitter data is the end product and hence does not need much cleaning. Cleaning here involves four things:-

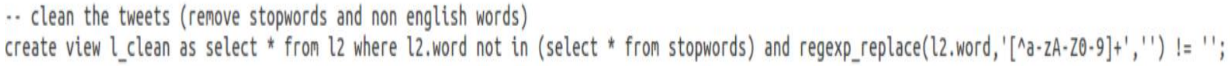
1) Taking forward only those data that are useful in our analysis.
2) Changing the date format from twitter format to UNIX format for processing in HIVE.
3) Removal of foreign language characters
4) Removal of stop words.



```
-- Clean up tweets
CREATE VIEW tweets_simple AS
SELECT
  id,
  `user`.screen_name,
  source,
  retweet_count,
  entities.hashtags,
  cast ( from_unixtime( unix_timestamp(concat( '2016 ', substring(created_at,5,15)), 'yyyy MMM dd hh:mm:ss')) as timestamp) ts,
  text,
  `user`.statuses_count,
  `user`.friends_count,
  `user`.followers_count,
  `user`.time_zone
FROM Mytweets_raw;
```



```
-- add country
CREATE VIEW tweets_clean AS
SELECT
  id,
  t.screen_name,
  source,
  retweet_count,
  t.hashtags,
  ts,
  text,
  m.country
FROM tweets_rem_spam t LEFT OUTER JOIN time_zone_map m ON t.time_zone = m.time_zone;
```

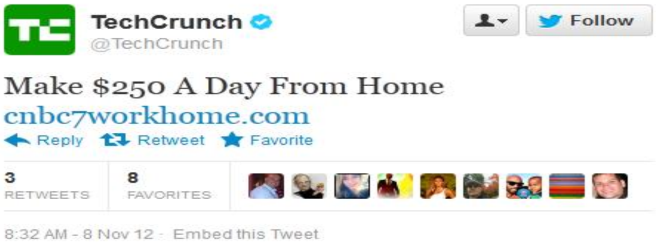


```
-- clean the tweets (remove stopwords and non english words)
create view l_clean as select * from l2 where l2.word not in (select * from stopwords) and regexp_replace(l2.word,'[^a-zA-Z0-9]+','') != '';
```

VII. SPAM DETECTION

To get the best results it’s important to filter the content relevant to the use case, and to remove what is consider as spam. Here are some of the ideas that can be applied to the spam function for removal of spam tweets:

- **Recently Created User**
  A common tactic by spam bots is to create new accounts on-the-fly and tweet from these accounts until they are shut down. We can get the account created date from the data and filter out the accounts.
- **Users with few followers**
  If a Twitter profile is created just to post spam messages, the profile is likely to follow lots of users, but be followed by very few users itself. If this is the case then the users’ followers’ ratio (number of users who follow the user, divided by the number of users they follow) will be low.
- **Users With Short Descriptions**
  If a Twitter profile has no description, again it could be from a bot or at least from a user who doesn't care about their public profile and has little concern for the quality of content they post.
- **Large Numbers Of Hashtags**
  Poor quality content tends to include many hashtags. Many hashtags might be used by spam creators to hope that they can reach as many users listening to those tags as possible
- **Short Content Length**
  Often users will write very short posts, such as '@friend ok' as a response to a question. This content has little value in analysis.

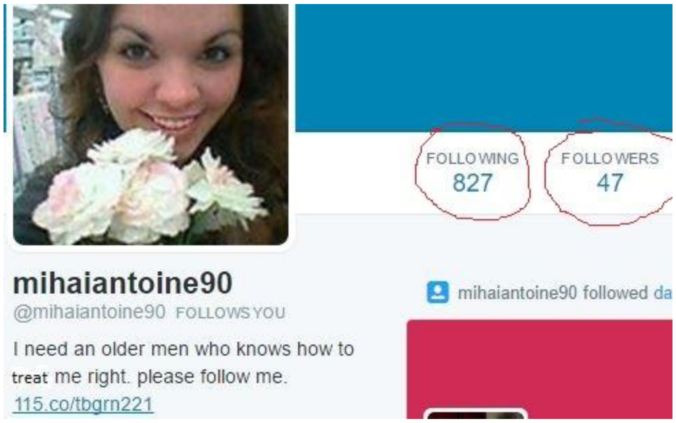


TechCrunch
@TechCrunch

Make $250 A Day From Home
cnbc7workhome.com

3 RETWEETS 8 FAVORITES

8:32 AM - 8 Nov 12 · Embed this Tweet

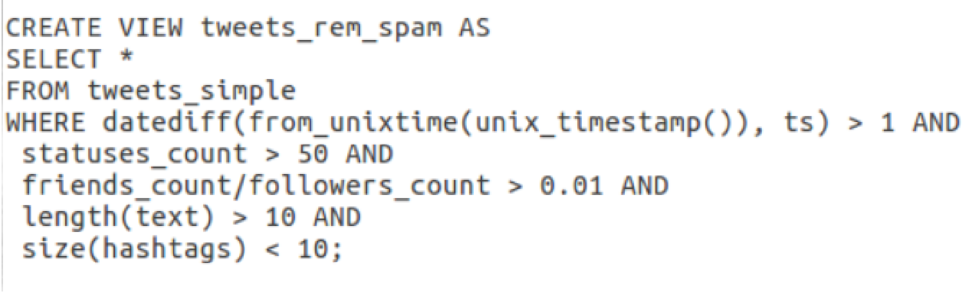


FOLLOWING 827 FOLLOWERS 47

mihaiantoine90
@mihaiantoine90 FOLLOWS YOU

I need an older men who knows how to treat me right. please follow me.
115.co/tbgrn221

mihaiantoine90 followed da



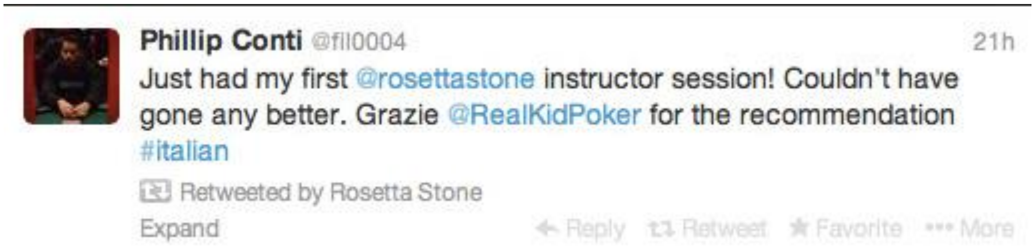
```
CREATE VIEW tweets_rem_spam AS
SELECT *
FROM tweets_simple
WHERE datediff(from_unixtime(unix_timestamp()), ts) > 1 AND
 statuses_count > 50 AND
 friends_count/followers_count > 0.01 AND
 length(text) > 10 AND
 size(hashtags) < 10;
```

VIII. SENTIMENT ANALYSIS AND RATING CALCULATION

What is Sentiment Analysis?

• It's a classification of polarity of a given text in the document, sentence or phrase • The goal is to determine whether the expressed opinion in the text is positive, negative or neutral

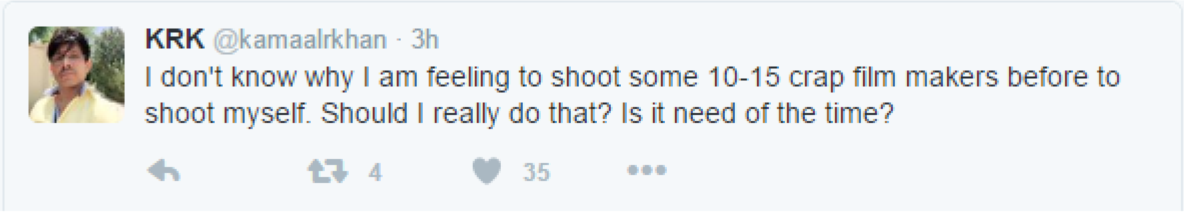
Positive tweet:-



Phillip Conti @fil0004 21h
Just had my first @rosettastone instructor session! Couldn't have gone any better. Grazie @RealKidPoker for the recommendation #italian

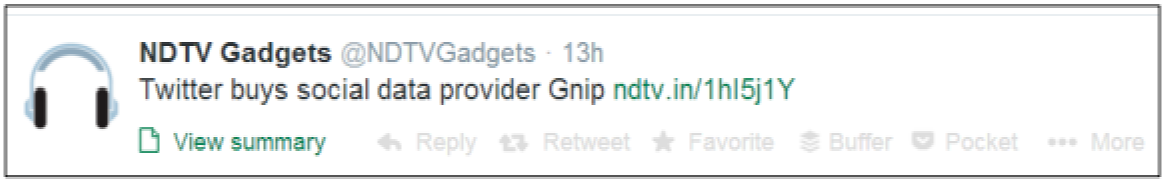
Retweeted by Rosetta Stone

Negative tweet:-



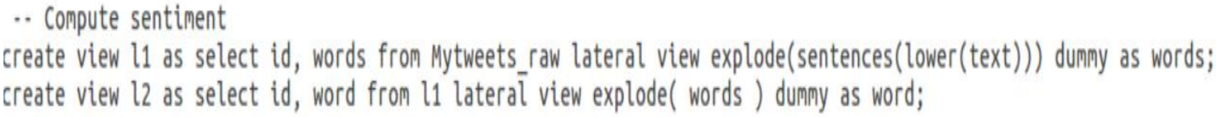
KRK @kamaalrkhan · 3h
I don't know why I am feeling to shoot some 10-15 crap film makers before to shoot myself. Should I really do that? Is it need of the time?

Neutral tweets:



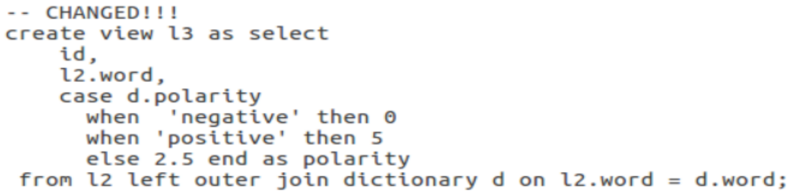
NDTV Gadgets @NDTVGadgets · 13h
Twitter buys social data provider Gnip ndtv.in/1hl5j1Y

1. Convert Sentences into array of words



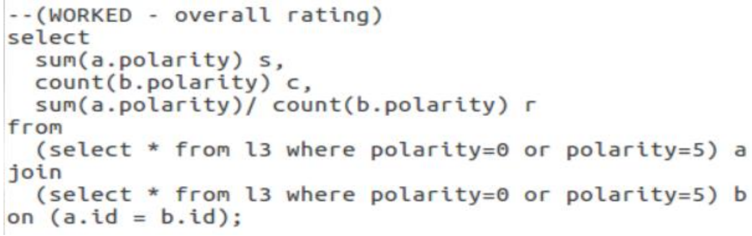
```
 -- Compute sentiment
create view l1 as select id, words from Mytweets_raw lateral view explode(sentences(lower(text))) dummy as words;
create view l2 as select id, word from l1 lateral view explode( words ) dummy as word;
```

2. Look up from the dictionary and find out its polarity



```
-- CHANGED!!!
create view l3 as select
    id,
    l2.word,
    case d.polarity
      when  'negative' then 0
      when 'positive' then 5
      else 2.5 end as polarity
 from l2 left outer join dictionary d on l2.word = d.word;
```

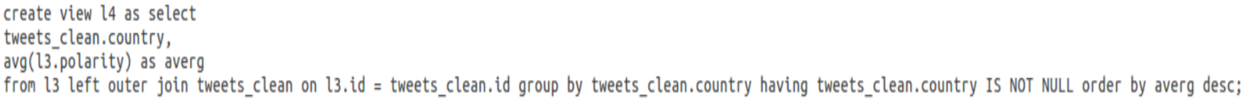
3. Find the Rating



```
--(WORKED - overall rating)
select
  sum(a.polarity) s,
  count(b.polarity) c,
  sum(a.polarity)/ count(b.polarity) r
from
  (select * from l3 where polarity=0 or polarity=5) a
join
  (select * from l3 where polarity=0 or polarity=5) b
on (a.id = b.id);
```

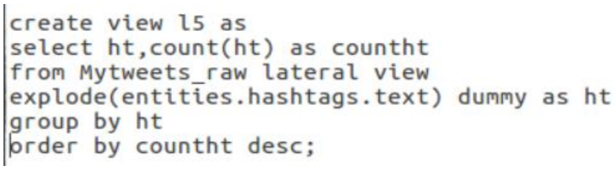
IX. ANALYSIS

1. Country-wise Rating:



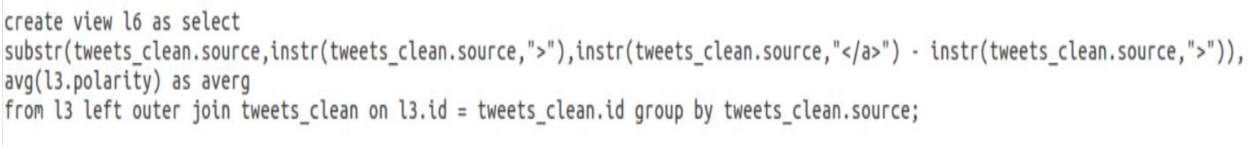
```
create view l4 as select
tweets_clean.country,
avg(l3.polarity) as averg
from l3 left outer join tweets_clean on l3.id = tweets_clean.id group by tweets_clean.country having tweets_clean.country IS NOT NULL order by averg desc;
```

2. Top Hashtags



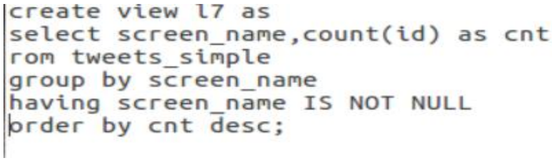
```
create view l5 as
select ht,count(ht) as countht
from Mytweets_raw lateral view
explode(entities.hashtags.text) dummy as ht
group by ht
order by countht desc;
```

3. Ratings by different sources



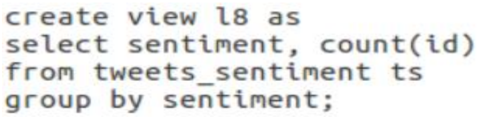
```
create view l6 as select
substr(tweets_clean.source,instr(tweets_clean.source,">"),instr(tweets_clean.source,"</a>") - instr(tweets_clean.source,">")),
avg(l3.polarity) as averg
from l3 left outer join tweets_clean on l3.id = tweets_clean.id group by tweets_clean.source;
```

4. Users with most tweets



```
create view l7 as
select screen_name,count(id) as cnt
rom tweets_simple
group by screen_name
having screen_name IS NOT NULL
order by cnt desc;
```

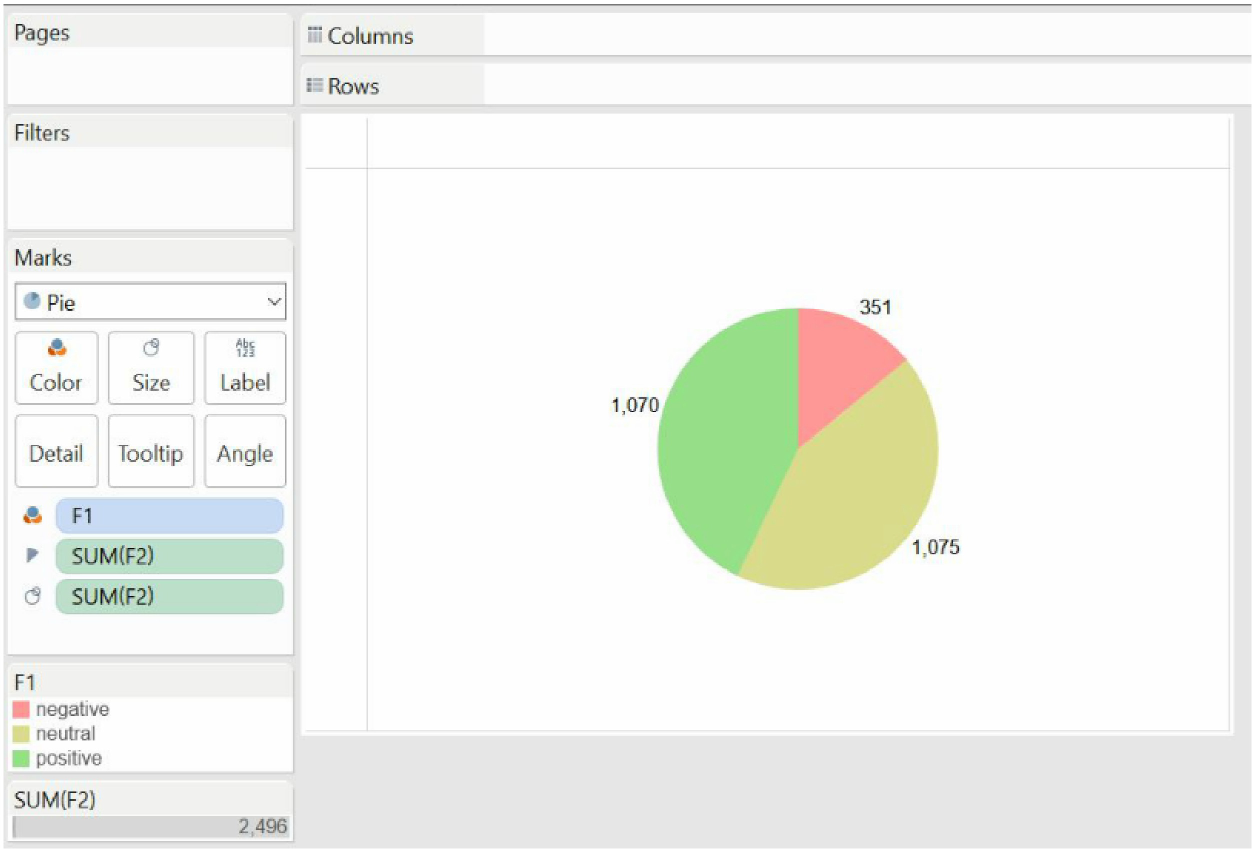
5. Sentiment count



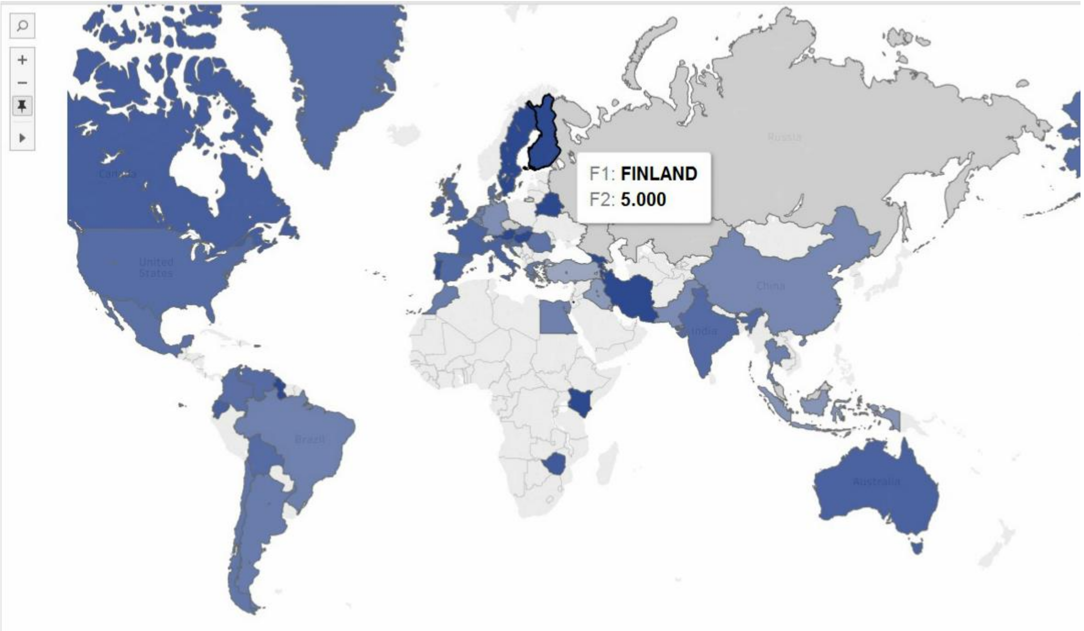
```
create view l8 as
select sentiment, count(id)
from tweets_sentiment ts
group by sentiment;
```

X. RESULTS

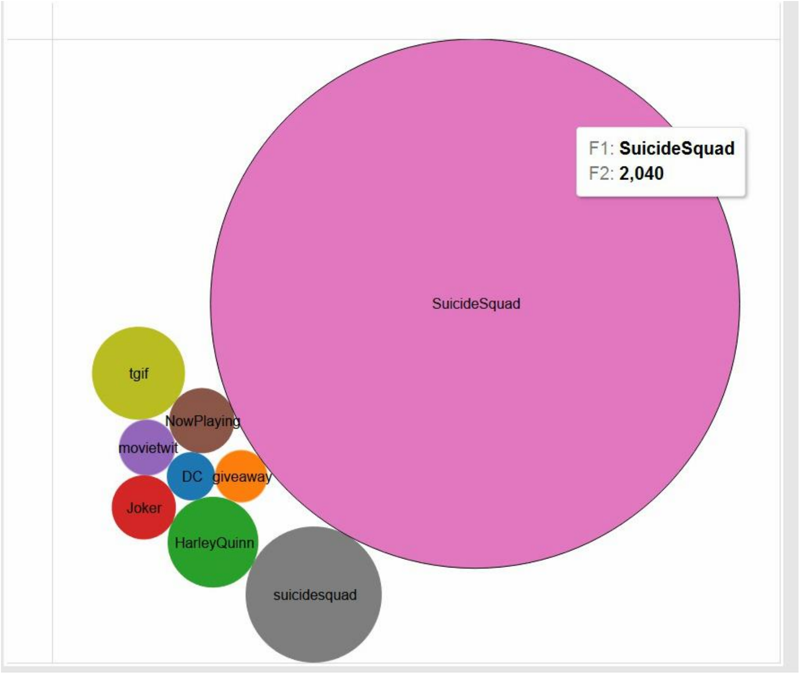
1. Sentiments as positive, negative and neutral



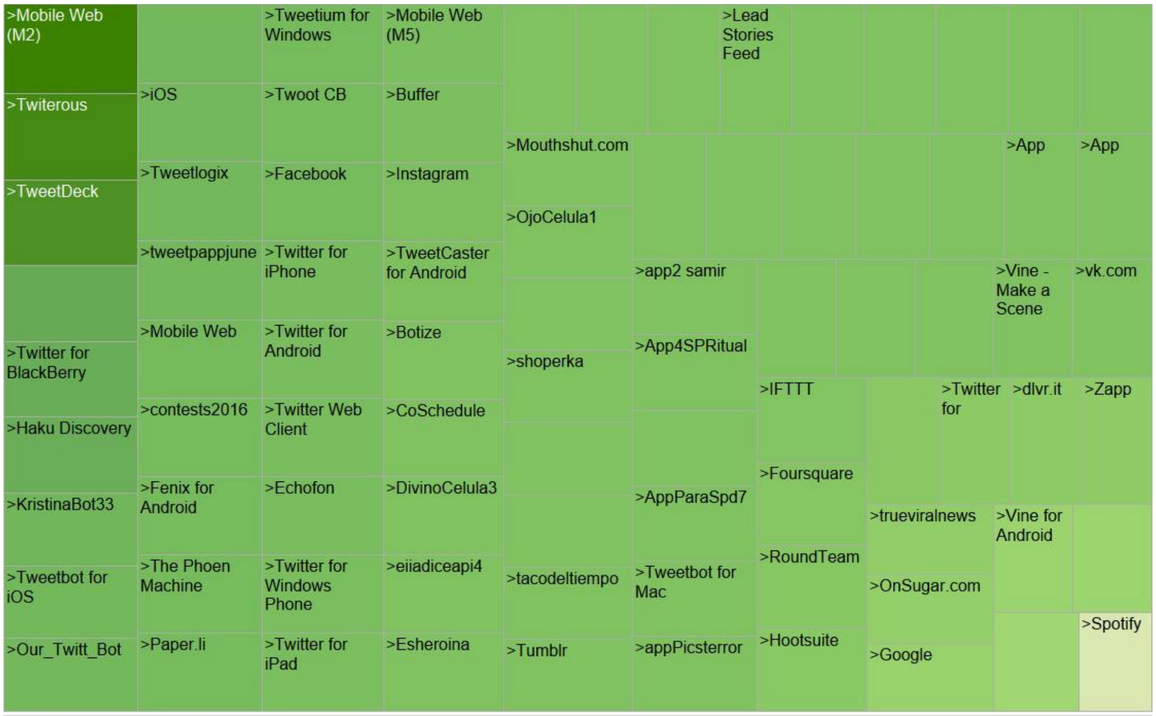
2. Country-wise Ratings



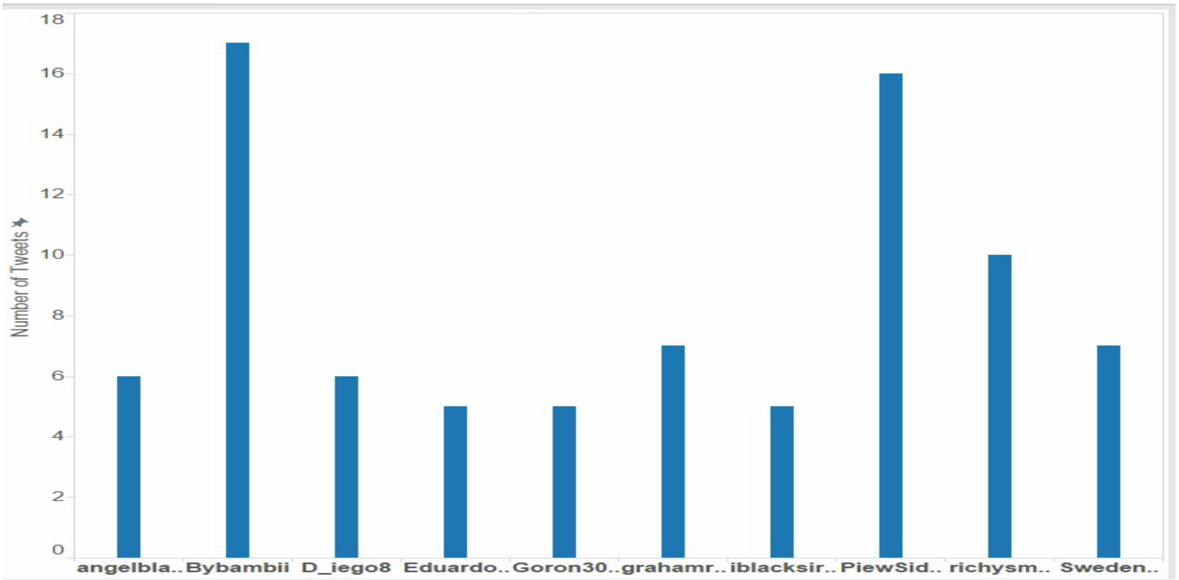
3. Top 10 Hashtags



4. Tweets from different sources



5. Top users who tweeted about the product



XI. PROBLEMS FACED

1. Retrieving the twitter data was difficult as twitter have not made their data public. Data is provided only to twitter apps.
2. Flume setup needed a lot of complex classpath JARs and configurations.
3. Creating a customized JSON SerDe for conversion of unstructured JSON data to structured Hive tables.
4. Setting up Amazon Web Services for running HIVE queries.
5. Connecting HIVE with Tableau.

XII. CONCLUSION

- Apache FLUME is a pretty powerful technology for data streaming.
- We have combined the real stream data with our spam detection function for classification of tweets as Spam.
- We have combined real stream data with NLP dataset to get sentiment analysis
- We successfully calculated the Rating System and carried out analysis on the ratings.

XIII. FUTURE SCOPE

1. **Negation Handling -** Words like ‘no’, ‘not’, and ‘never’ are difficult to handle properly. This can be handled by using n-grams which is an inbuilt function in HIVE which can analyze two or three words together and then look up in the dictionary look up.
2. **Sarcasm Detection –** Sarcasm is tough to detect in written form unless you know the context. When speaking, the tone usually gives away the Sarcasm. That's not the case in written text though. This can be handled to almost the accuracy of 80% by machine learning techniques and by NLP by analyzing the smileys and words like ‘lol’,’wow’,’(not)’,’!!!’.

XIV. REFERENCES

1. Agarwal, A., Xie, B., Vovsha, I., Rambow, O., Passonneau, R.: Sentiment analysis of twitter data. In: Proc. ACL 2011 Workshop on Languages in Social Media. pp. 30–38 (2011)
2. Barbosa, L., Feng, J.: Robust sentiment detection on twitter from biased and noisy data. In: Proceedings of COLING. pp. 36–44 (2010)
3. Gimpel, K., Schneider, N., O'Connor, B., Das, D., Mills, D., Eisenstein, J., Heilman, M., Yogatama, D., Flanigan, J., Smith, N.: Part-of-speech tagging for twitter: Annotation, features, and experiments. Tech. rep., DTIC Document (2010)
4. Go, A., Bhayani, R., Huang, L.: Twitter sentiment classification using distant supervision. CS224N Project Report, Stanford (2009)
5. Guerra, P., Veloso, A., Meira Jr, W., Almeida, V.: From bias to opinion: A transfer-learning approach to real-time sentiment analysis. In: Proceedings of the 17th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD) (2011)
6. Kouloumpis, E., Wilson, T., Moore, J.: Twitter sentiment analysis: The good the bad and the omg! In: Proceedings of the ICWSM (2011)
7. Lin, C., He, Y.: Joint sentiment/topic model for sentiment analysis. In: Proceeding of the 18th ACM conference on Information and knowledge management. pp. 375–384. ACM (2009)
8. Moschitti, A.: Efficient convolution kernels for dependency and constituent syntactic trees. In: Proceedings of the European Conference on Machine Learning. pp. 318–329 (2006)
9. Pak, A., Paroubek, P.: Twitter as a corpus for sentiment analysis and opinion mining. Proceedings of LREC 2010 (2010)
10. Rizzo, G., Troncy, R.: Nerd: Evaluating named entity recognition tools in the web of data. In: Workshop on Web Scale Knowledge Extraction (WEKEX11). vol. 21 (2011)
11. Saif, H., He, Y., Alani, H.: Semantic Smoothing for Twitter Sentiment Analysis. In: Proceeding of the 10th International Semantic Web Conference (ISWC) (2011)
12. Grant Stafford, Louis Lei Yu : An Evaluation of the Effect of Spam on Twitter Trending Topics. In: Gustavus Adolphus College Conference