SENTIMENT ANALYSIS OF PRESIDENT TRUMP'S TWEETS

Akshay Ketkar
ahketka2@illinois.edu

Table of Contents

1. INTRODUCTION

In today’s world of social media and networking, there is a tremendous creation and exchange of information. It is much easier today to express one’s opinions on the public platform than what it was in the era gone by. One such social media platform, **twitter.com** has become an important media through which people can express or shape opinions on several matters of social, political, national or international standing.

The project analyzes a dataset released on kaggle.com. The dataset consists of over 7000 of the current US President Mr. Donald Trump’s tweets from 2015 to 2016. The project endeavors to run a sentiment analysis on the dataset to determine the proportion of positive and negative words apart from some other very frequent references using text analysis techniques.

2.LITERATURE REVIEW

1. Finding Similar Items in Structured Datasets

Reference: *IE531-Algorithms for Data Analysis- Lesson 3: Finding Similar Items in Structured Data sets by Prof. R.S. Sreenivas*

This methodology is used for finding similar items among collections. While most cases dealt with are in terms of measuring the similarity between texts, the procedures here can be extended to others as well. Multimedia such as images and audio would require a feature-extraction step, which then proceeds to analyze features on similar grounds

Similarity measures:

The Jaccard similarity of sets S and T is SIM(S, T) = card(S∪T) / card(S∩T) , is the ratio of the number of common elements between two sets to the total number of elements in both sets. For this, the notion of shingling is introduced. That is, we slide a window of size k across the sentence and record what we see. The k-shingles are computed using a window of size k. The common practice, when it comes to text files is to use a value of k = 9. To compute the similarity between two text files, we compute Jaccard Similarity of their 9-shingle set. We use this technique to improve our computation of the proportion of positive and negative terms in President Trump's tweets.

2. Map-Reduce

Reference: *IE531-Algorithms for Data Analysis- Lesson 5: Basics of MapReduce and some related topics by Prof. R.S. Sreenivas*

Although an attempt has been made to get in the code in the Mapper and Reducer form, the execution was not successful on the Cloudera Emulator's Hadoop file system. Instead the codes were successfully run on Cloudera local machine using pipe, to stream twitter words from Mapper code to the Reducer code, whereby the Reducer code does all the analysis.

MapReduce operations:

All **MapReduce** operations have two parts to it a mapper, which produces intermediate values in the form of (key; value) the output of the mapper is sorted based on the key values and presented to the reducer, which does the needful to complete the task. The mapper code and reducer code run on different nodes on different blocks of data.

3. PROJECT FLOW

<u>Step1</u>:

Data Cleaning: The twitter data set is cleaned for the removal of quotes and other special characters such as commas, hashes, question marks, exclamations and other characters such as smileys. All the tweets are converted to lower case and outputted to a file.

<u>Step2</u>:

Mapper Code: The Mapper code parses each line of the twitter text column from the file mentioned above. Then each of the tweets is split to generate words, which are streamed for the Reducer code.

<u>Step3:</u>

Reducer Code: The Reducer code takes the words streamed by the Mapper and does the following:

a) Creates a dictionary of the distinct words used and creating a list of streamed words

b) Using the stream list and dataset containing Positive and Negative words from
https://github.com/jeffreybreen/twitter-sentiment-analysis-tutorial-201107/tree/master/data/opinion-lexicon-English

, it creates a list of positive and negative words, creates 9-character shingles of positive, negative and twitter words.

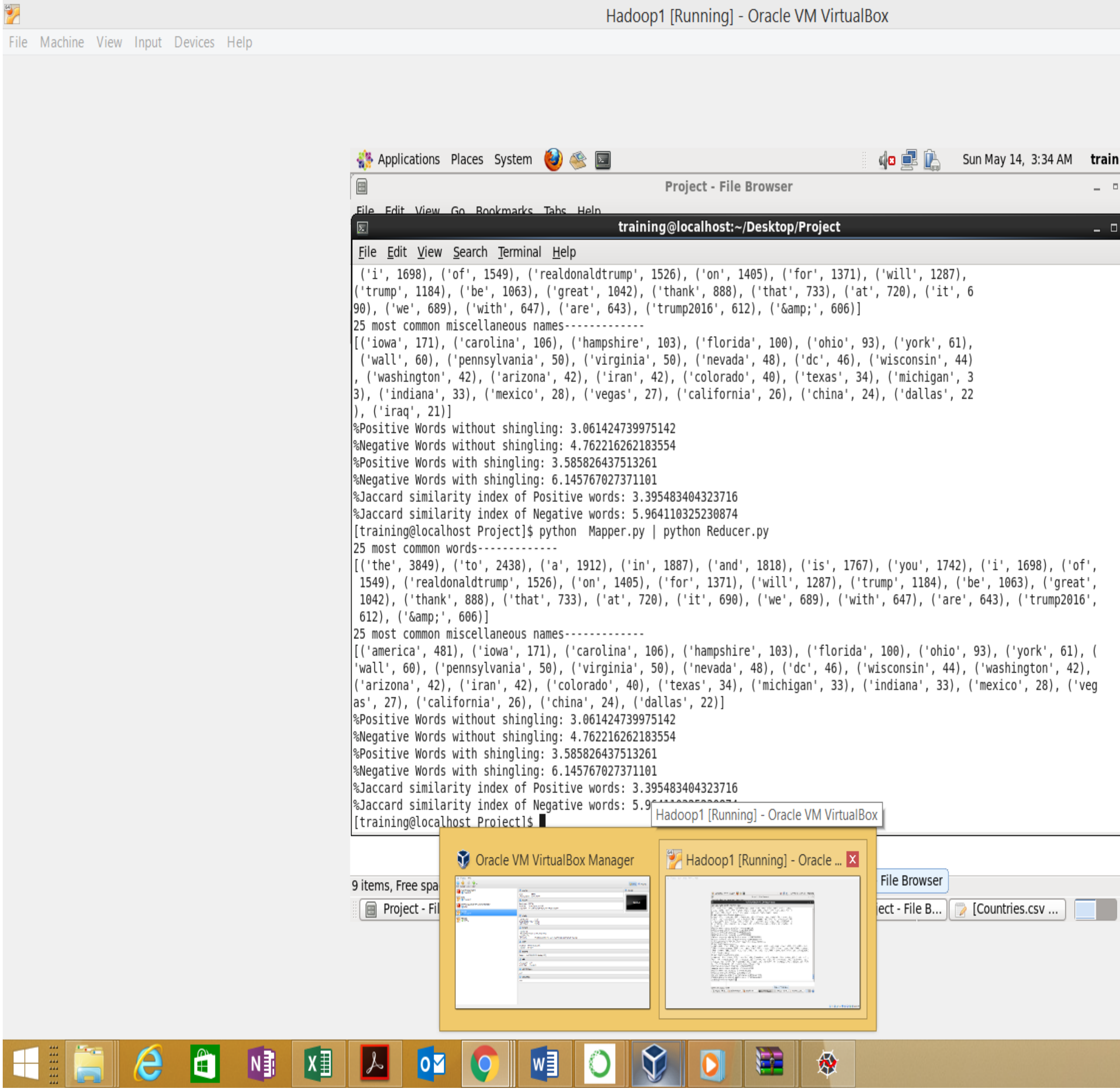
c) A dataset consisting mostly of country and US State names, but also a few terms such as 'wall' has been used to determine how frequently President Trump uses them.

d) The Jaccard similarity and proportion of Positive and Negative words is computed. Also, the dictionary of wordcounts and miscellaneous terms is sorted to reveal the top 25 most frequent entries.

e) The respective details are printed.

4. RESULTS

1. Screenshot:


Hadoop1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places System
Sun May 14, 3:34 AM
train
Project - File Browser
File Edit View Go Bookmarks Tabs Help
training@localhost:~/Desktop/Project
File Edit View Search Terminal Help
Hadoop1 [Running] - Oracle VM VirtualBox
Oracle VM VirtualBox Manager
Hadoop1 [Running] - Oracle ...
9 items, Free spa
Project - Fil
File Browser
ject - File B...
[Countries.csv ...

```
 ('i', 1698), ('of', 1549), ('realdonaldtrump', 1526), ('on', 1405), ('for', 1371), ('will', 1287),
('trump', 1184), ('be', 1063), ('great', 1042), ('thank', 888), ('that', 733), ('at', 720), ('it', 6
90), ('we', 689), ('with', 647), ('are', 643), ('trump2016', 612), ('&amp;', 606)]
25 most common miscellaneous names-------------
[('iowa', 171), ('carolina', 106), ('hampshire', 103), ('florida', 100), ('ohio', 93), ('york', 61),
 ('wall', 60), ('pennsylvania', 50), ('virginia', 50), ('nevada', 48), ('dc', 46), ('wisconsin', 44)
, ('washington', 42), ('arizona', 42), ('iran', 42), ('colorado', 40), ('texas', 34), ('michigan', 3
3), ('indiana', 33), ('mexico', 28), ('vegas', 27), ('california', 26), ('china', 24), ('dallas', 22
), ('iraq', 21)]
%Positive Words without shingling: 3.061424739975142
%Negative Words without shingling: 4.762216262183554
%Positive Words with shingling: 3.585826437513261
%Negative Words with shingling: 6.145767027371101
%Jaccard similarity index of Positive words: 3.395483404323716
%Jaccard similarity index of Negative words: 5.964110325230874
[training@localhost Project]$ python  Mapper.py | python Reducer.py
25 most common words-------------
[('the', 3849), ('to', 2438), ('a', 1912), ('in', 1887), ('and', 1818), ('is', 1767), ('you', 1742), ('i', 1698), ('of',
 1549), ('realdonaldtrump', 1526), ('on', 1405), ('for', 1371), ('will', 1287), ('trump', 1184), ('be', 1063), ('great',
 1042), ('thank', 888), ('that', 733), ('at', 720), ('it', 690), ('we', 689), ('with', 647), ('are', 643), ('trump2016',
 612), ('&amp;', 606)]
25 most common miscellaneous names-------------
[('america', 481), ('iowa', 171), ('carolina', 106), ('hampshire', 103), ('florida', 100), ('ohio', 93), ('york', 61), (
'wall', 60), ('pennsylvania', 50), ('virginia', 50), ('nevada', 48), ('dc', 46), ('wisconsin', 44), ('washington', 42),
('arizona', 42), ('iran', 42), ('colorado', 40), ('texas', 34), ('michigan', 33), ('indiana', 33), ('mexico', 28), ('veg
as', 27), ('california', 26), ('china', 24), ('dallas', 22)]
%Positive Words without shingling: 3.061424739975142
%Negative Words without shingling: 4.762216262183554
%Positive Words with shingling: 3.585826437513261
%Negative Words with shingling: 6.145767027371101
%Jaccard similarity index of Positive words: 3.395483404323716
%Jaccard similarity index of Negative words: 5.9
[training@localhost Project]$
```

2. Interesting Results:

a) '25 most common words' and '25 most common miscellaneous terms:

Among the 25 most common words, the words that stand out are **'great'(Count:1042) and 'thank'(Count:888) and 'trump'(Count:1184).** Some of the 'great' may refer to **'Let's make America great again!'** as the count for the term **'a(A)merica' is uttered 481** times as well. The utterance of the word **'thank'** may refer to President Trump's thank you for several compliments and good wishes.

Among American states, the state **Iowa** features **171** times. Among countries, the country **Iran** features **42 times.** Among other words, the word **'wall' features 60 times**. Some of the sentences consisting of 'wall' could point towards building a wall on the border with Mexico.

b) Results of analysis using Similarity Measures: Shingling and Jaccard Similarity index:

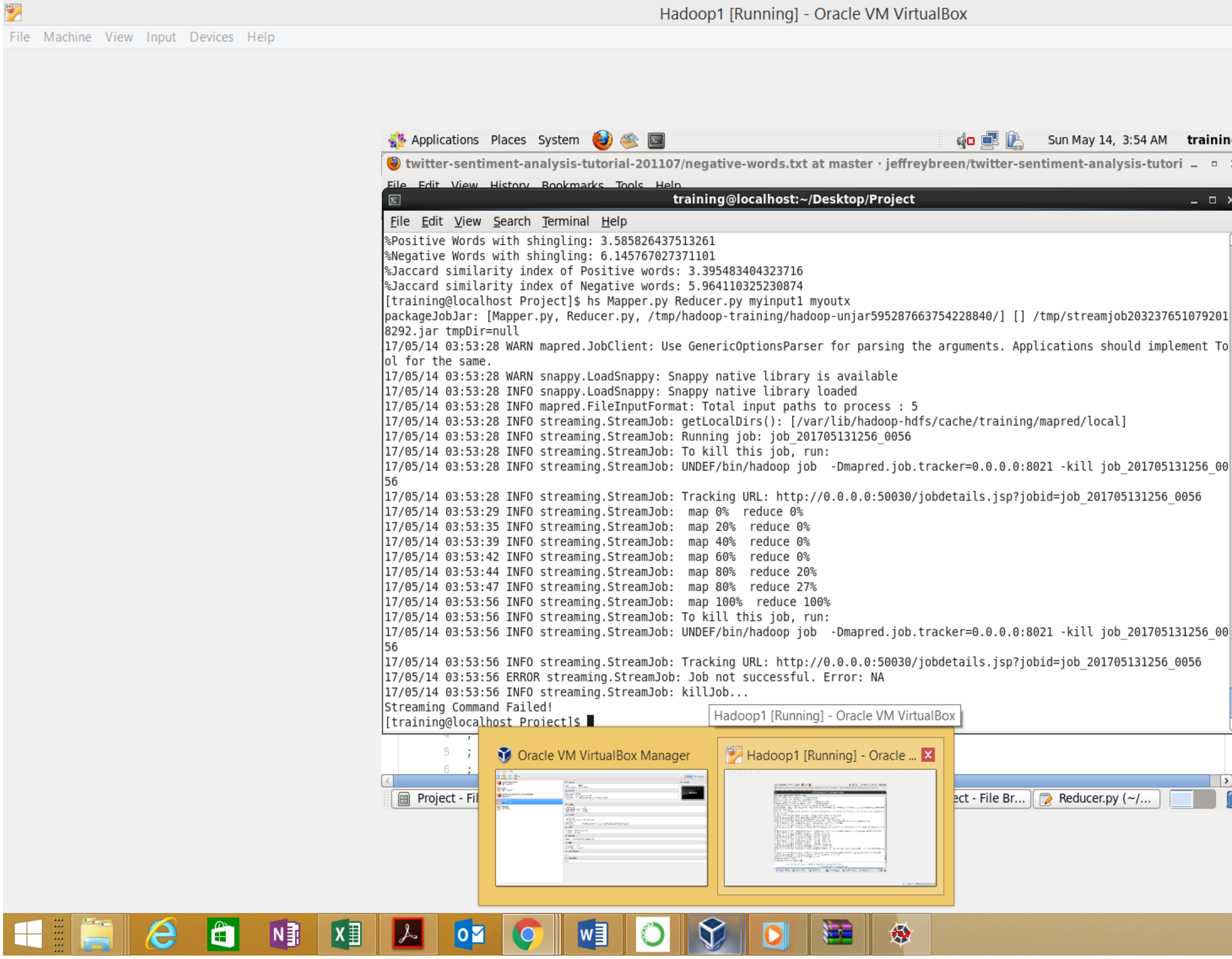
- Taking **direct ratio** of the number of **Positive** and **Negative** words with the **total number of distinct twitter words** yields **3.06%** and **4.76%** words as Positive and Negative respectively.

- However, **using 9-character shingles** (as per convention), we get a slight improvement in the results, with the percentage of **Positive** and **Negative** word estimate increasing to **3.58%** and **6.15%.** The Jaccard similarity index is **3.39%** and **5.96%** for **Positive** and **Negative** words respectively.

5. CONCLUSION, HADOOP AND SCOPE FOR IMPROVEMENT

It can be concluded from the Sentiment Analysis that there is a consistently higher percentage of Negative words (With or without shingling), indicating a marginally higher negative sentiment than a positive sentiment.

Hadoop Screenshot



Hadoop1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places System
Sun May 14, 3:54 AM
twitter-sentiment-analysis-tutorial-201107/negative-words.txt at master · jeffreybreen/twitter-sentiment-analysis-tutori
File Edit View History Bookmarks Tools Help
training@localhost:~/Desktop/Project
File Edit View Search Terminal Help
Hadoop1 [Running] - Oracle VM VirtualBox
Oracle VM VirtualBox Manager
Hadoop1 [Running] - Oracle ...
Project - Fil
ect - File Br...
Reducer.py (~/...

```
%Positive Words with shingling: 3.585826437513261
%Negative Words with shingling: 6.145767027371101
%Jaccard similarity index of Positive words: 3.395483404323716
%Jaccard similarity index of Negative words: 5.964110325230874
[training@localhost Project]$ hs Mapper.py Reducer.py myinput1 myoutx
packageJobJar: [Mapper.py, Reducer.py, /tmp/hadoop-training/hadoop-unjar595287663754228840/] [] /tmp/streamjob203237651079201
8292.jar tmpDir=null
17/05/14 03:53:28 WARN mapred.JobClient: Use GenericOptionsParser for parsing the arguments. Applications should implement To
ol for the same.
17/05/14 03:53:28 WARN snappy.LoadSnappy: Snappy native library is available
17/05/14 03:53:28 INFO snappy.LoadSnappy: Snappy native library loaded
17/05/14 03:53:28 INFO mapred.FileInputFormat: Total input paths to process : 5
17/05/14 03:53:28 INFO streaming.StreamJob: getLocalDirs(): [/var/lib/hadoop-hdfs/cache/training/mapred/local]
17/05/14 03:53:28 INFO streaming.StreamJob: Running job: job_201705131256_0056
17/05/14 03:53:28 INFO streaming.StreamJob: To kill this job, run:
17/05/14 03:53:28 INFO streaming.StreamJob: UNDEF/bin/hadoop job  -Dmapred.job.tracker=0.0.0.0:8021 -kill job_201705131256_00
56
17/05/14 03:53:28 INFO streaming.StreamJob: Tracking URL: http://0.0.0.0:50030/jobdetails.jsp?jobid=job_201705131256_0056
17/05/14 03:53:29 INFO streaming.StreamJob:  map 0%  reduce 0%
17/05/14 03:53:35 INFO streaming.StreamJob:  map 20%  reduce 0%
17/05/14 03:53:39 INFO streaming.StreamJob:  map 40%  reduce 0%
17/05/14 03:53:42 INFO streaming.StreamJob:  map 60%  reduce 0%
17/05/14 03:53:44 INFO streaming.StreamJob:  map 80%  reduce 20%
17/05/14 03:53:47 INFO streaming.StreamJob:  map 80%  reduce 27%
17/05/14 03:53:56 INFO streaming.StreamJob:  map 100%  reduce 100%
17/05/14 03:53:56 INFO streaming.StreamJob: To kill this job, run:
17/05/14 03:53:56 INFO streaming.StreamJob: UNDEF/bin/hadoop job  -Dmapred.job.tracker=0.0.0.0:8021 -kill job_201705131256_00
56
17/05/14 03:53:56 INFO streaming.StreamJob: Tracking URL: http://0.0.0.0:50030/jobdetails.jsp?jobid=job_201705131256_0056
17/05/14 03:53:56 ERROR streaming.StreamJob: Job not successful. Error: NA
17/05/14 03:53:56 INFO streaming.StreamJob: killJob...
Streaming Command Failed!
[training@localhost Project]$
```

Scope for Improvement:

It was not possible to run the MapReduce on Cloudera emulator. If there is a way to directly connect to a Hadoop cluster and determine the reasons for the failure, there could be a successful run possible.

6. REFERENCES

1. https://www.kaggle.com/kingburrito666/better-donald-trump-tweets

2.https://github.com/jeffreybreen/twitter-sentiment-analysis-tutorial-201107/tree/master/data/opinion-lexicon-English

3. *IE531-Algorithms for Data Analysis- Lesson 3: Finding Similar Items in Structured Data sets by Prof. R.S. Sreenivas*

4. *IE531-Algorithms for Data Analysis- Lesson 5: Basics of MapReduce and some related topics by Prof. R.S. Sreenivas*