Store, Analyze and Visualize “tweets” from Twitter

Team members:
Lakshmi Chamala
Gulnoza Khakimova
Khushbu Kolhe

Instructor: Dr. Praveen Rao
CS5540
Principles of Big Data Management

Introduction

Twitter is an online social network where users communicate with each other through messages called “tweets”. Tweets are visible to everyone but communicator can restrict message delivery to just his/her followers. Users can group “tweets” by topic or hashtags. The most posted topics are called “trending topic”. Those topics help users to identify what is happening in the world. On average there are 1.6 billion of messages posted every day.
In this project we will collect “tweets”, analyze them based on specific criteria and visualize results.
Our research consists of three phases: The first step will be to collect tweets on the following topics: “Bitcoin”,”Forbes” and “Winter Olympics”. In the second step we will analyze “tweets” by writing queries. The last step will be demonstration of our project.

Phase 1

In this phase we need to collect “tweets” for the following topics: “Bitcoin”, “Forbes”, “Winter Olympics”. First we had to create and register our application on http://apps.twitter.com in order to access Twitter data and interact with Twitter API. After registration we were given a consumer key, consumer secret and access tokens which are used to get access to Twitter data.

We tried different ways to get tweets using Twitter streaming API . After collecting good amount of tweets, which we stored into google drive (link to collected tweets: https://drive.google.com/open?id=1EM9e3iK-OxoyAe2XetQRdU_LWNjyCBn6) we were able to extract hashtags and URLs for each “tweet” so we could run word count in Apache Hadoop and Apache Spark.

Apache Hadoop

Collect tweets

Below is a code on Python which were used to collect “tweets”:

```
*collecttweets.py (~/Desktop)
File Edit View Search Tools Documents Help

from tweepy.streaming import StreamListener
from tweepy import OAuthHandler
from tweepy import Stream



consumer_key="LBiJTcvUFWxjG2cC5xYaRLaLQ"
consumer_secret="ExuqqOHAWE9Css3GiNQpZLBclwdxWJKgb7CBioMaqFa45dpujP"
access_token="926091370363793413-BXuuOZBlqHS8kUDWs54XVWgH8AEWm8s"
access_token_secret="FskYTykFTJnqplMod1HPZpxAkNr2Dn2ExKl0vdxJ4DTqP"

class StdOutListener(StreamListener):

    def on_data(self, data):
        with open('tweets_output.json','a') as tf:
            tf.write(data)
            print(data)
            return True

if __name__ == '__main__':
    auth = OAuthHandler(consumer_key, consumer_secret)
    auth.set_access_token(access_token, access_token_secret)
    stream = Stream(auth, StdOutListener())
stream.filter(track=['Forbes','Bitcoin','Winter Olympics'])

Python  Tab Width: 4  Ln 1, Col 1  INS
Menu  Store, analyze and vi...  *collecttweets.py (~/...  22:23
```

Word count on Apache Hadoop

Following command was used to perform word count on extracted hashtags and URL's from "tweets" on Apache Hadoop:
bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.8.1.jar wordcount /Project/output.txt /Project/out

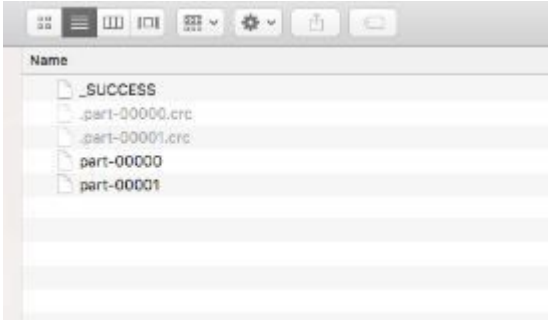
Apache Spark

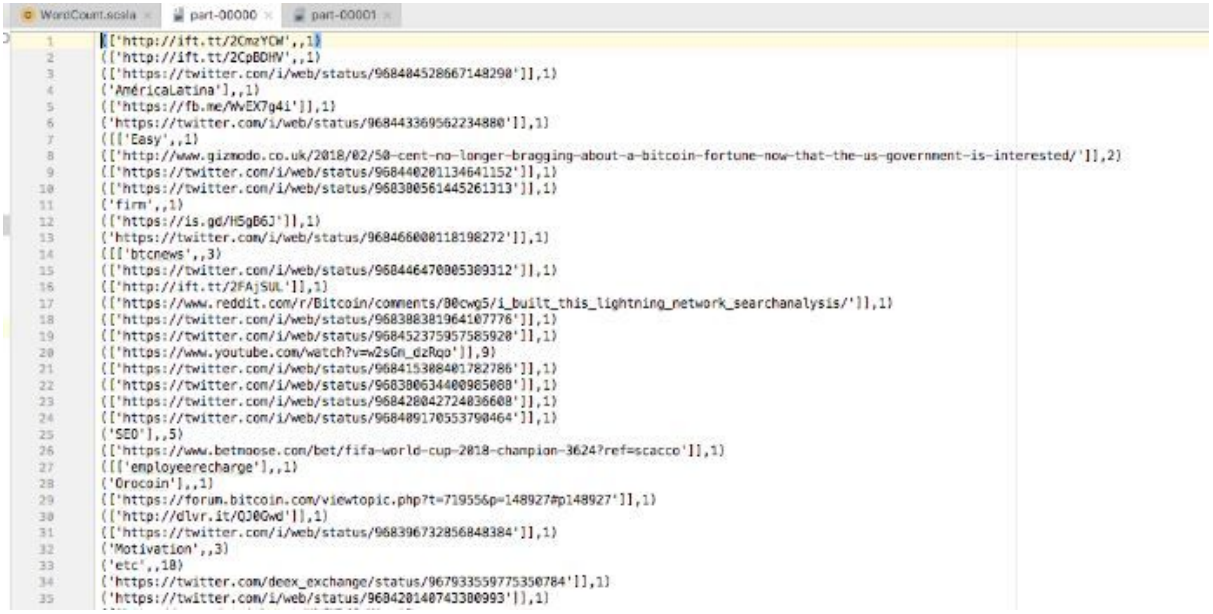
Below is a code for Apache Spark which was used to perform "wordcount" on extracted hashtags and URLs

WordCount.scala

```
import org.apache.spark.{SparkContext, SparkConf}

object WordCount {

  def main(args: Array[String]) {

    val sparkConf = new SparkConf().setAppName("WordCount").setMaster("local[*]")

    val sc=new SparkContext(sparkConf)

    val input=sc.textFile("/Volumes/Data/PrinciplesOfBigData/WordCountHashTags/hashtagsAndUrl.txt")

    val wc=input.flatMap(line=>{line.split(" ")}).map(word=>(word,1)).cache()

    val output=wc.reduceByKey(_+_)

    output.saveAsTextFile("sparkWordCountOutput")

    val o=output.collect()

    var s:String="Words:Count \n"
    o.foreach{case(word,count)=>{

      s+=word+" : "+count+"\n"

    }}

  }

}
```

Output:





Hashtags and URLs extraction

Code wich was used to get Hashtags and URLs from collected tweets:

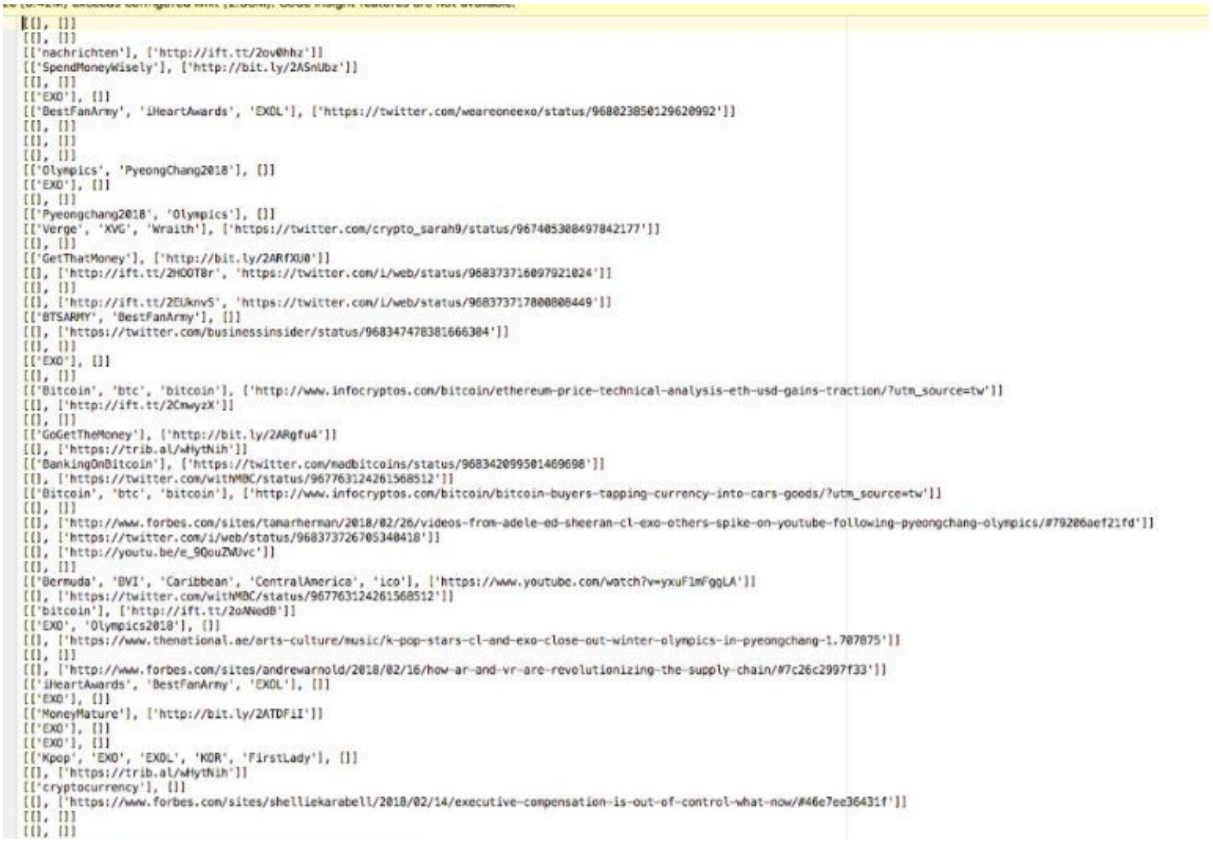
collecttweets.py | HashtagAndUrlExtraction.py | hashtagAndURLoutput.txt

```python
import codecs
from datetime import datetime
import json
# import requests
import os
import string
import sys
import time


def parse_json_tweet(line):
    tweet = json.loads(line)

    hashtags = [hashtag['text'] for hashtag in tweet['entities']['hashtags']]
    urls = [url['expanded_url'] for url in tweet['entities']['urls']]



    return [hashtags, urls]


'''start main'''
if __name__ == "__main__":
    file_timeordered_json_tweets = codecs.open("tweets_output.json", 'r', 'utf-8')
    fout = codecs.open("hashtagAndURLoutput.txt", 'w', 'utf-8')

    for line in file_timeordered_json_tweets:
        try:
            [hashtags, urls] = parse_json_tweet(line)
            fout.write(str([ hashtags,urls]) + "\n")
          #  fout.write("Hashtag"+ str([ hashtags]) + "URLs"+ str([urls]) + "\n")

        except:
            pass
    file_timeordered_json_tweets.close()
    fout.close()
```

collecttweets.py | HashtagAndUrlExtraction.py

```python
import codecs
from datetime import datetime
import json
# import requests
import os
import string
import sys
import time


def parse_json_tweet(line):
    tweet = json.loads(line)

    hashtags = [hashtag['text'] for hashtag in tweet['entities']['hashtags']]
    urls = [url['expanded_url'] for url in tweet['entities']['urls']]



    return [hashtags, urls]


'''start main'''
if __name__ == "__main__":
    file_timeordered_json_tweets = codecs.open("tweets_output.json", 'r', 'utf-8')
    fout = codecs.open("hashtagAndURLoutput.txt", 'w', 'utf-8')

    for line in file_timeordered_json_tweets:
        try:
            [hashtags, urls] = parse_json_tweet(line)
            fout.write(str([ hashtags,urls]) + "\n")
          #  fout.write("Hashtag"+ str([ hashtags]) + "URLs"+ str([urls]) + "\n")

        except:
            pass
    file_timeordered_json_tweets.close()
    fout.close()
```

Output:



Link to Github repository which contains code, input, output and logs:
https://github.com/Gnkhakimova/CS5540-Twitter