Python Script for Pulling Tweets:

import oauth2 as oauth

import urllib2 as urllib

''' THis script helps us to get as much twitter data as we can '''

# See Assignment 1 instructions or README for how to get these credentials

access\_token\_key = "283132017-MlzIKlJHWpCKjPylkMIO3f4xlKeUXo2bnm6EZgEl"

access\_token\_secret = "lLDfQZ445j5LBuVPm81ZLNajUod5ROnu2CKkfOjU4vpVe"

consumer\_key = "1hM0o74f73Hh4lyZVWYA"

consumer\_secret = "4yvJ7j14ZNbzq50agG40NLtKspyrwegZC7C667Cx2Ck"

\_debug = 0

oauth\_token = oauth.Token(key=access\_token\_key, secret=access\_token\_secret)

oauth\_consumer = oauth.Consumer(key=consumer\_key, secret=consumer\_secret)

signature\_method\_hmac\_sha1 = oauth.SignatureMethod\_HMAC\_SHA1()

http\_method = "GET"

http\_handler = urllib.HTTPHandler(debuglevel=\_debug)

https\_handler = urllib.HTTPSHandler(debuglevel=\_debug)

'''

Construct, sign, and open a twitter request

using the hard-coded credentials above.

'''

def twitterreq(url, method, parameters):

req = oauth.Request.from\_consumer\_and\_token(oauth\_consumer,

token=oauth\_token,

http\_method=http\_method,

http\_url=url,

parameters=parameters)

req.sign\_request(signature\_method\_hmac\_sha1, oauth\_consumer, oauth\_token)

headers = req.to\_header()

if http\_method == "POST":

encoded\_post\_data = req.to\_postdata()

else:

encoded\_post\_data = None

url = req.to\_url()

opener = urllib.OpenerDirector()

opener.add\_handler(http\_handler)

opener.add\_handler(https\_handler)

response = opener.open(url, encoded\_post\_data)

return response

def fetchsamples():

url = "https://stream.twitter.com/1.1/statuses/filter.json?track=feeling great,feeling good,feeling amazing,feeling sick,not feeling good,not feeling well,happy,sad,wonderful,lonely&language=en"

parameters = []

response = twitterreq(url, "GET", parameters)

for line in response:

print line.strip()

if \_\_name\_\_ == '\_\_main\_\_':

fetchsamples()

Map Reduce :

import java.io.BufferedReader;

import java.io.BufferedWriter;

import java.io.File;

import java.io.FileWriter;

import java.io.IOException;

import java.io.InputStreamReader;

import java.io.OutputStream;

import java.net.HttpURLConnection;

import java.net.URL;

import java.util.StringTokenizer;

import org.apache.hadoop.conf.Configuration;

import org.apache.hadoop.fs.Path;

import org.apache.hadoop.io.\*;

import org.apache.hadoop.mapreduce.Job;

import org.apache.hadoop.mapreduce.Mapper;

import org.apache.hadoop.mapreduce.Reducer;

import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;

import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

import org.json.simple.JSONObject;

import org.json.simple.parser.JSONParser;

import org.json.simple.parser.ParseException;

public class WordCount {

public static class Map

extends Mapper<LongWritable, Text, Text,IntWritable>{

private final static IntWritable one = new IntWritable(1); // type of output value

private Text word = new Text(); // type of output key

private Text word2 =new Text();

public void map(LongWritable key, Text value, Context context

) throws IOException, InterruptedException {

//System.out.println(value);

//System.out.println("I am in the map method");

JSONParser parser = new JSONParser();

// if file doesnt exists, then create it

// if (!file.exists()) {

//file.createNewFile();

// }

//FileWriter fw = new FileWriter(file.getAbsoluteFile());

//bw= new BufferedWriter(fw);

String tweet=null,timeZone=null, NullString=null, Emotion = "No Emotion";

Long id=null ;

String output = null ;

Object obj=null;

try {

//System.out.println(value.toString());

obj = parser.parse(value.toString());

//System.out.println(obj);

} catch (ParseException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

JSONObject jsonObject = (JSONObject)obj;

JSONObject userObject = (JSONObject)jsonObject.get("user");

if (jsonObject.get("text")!= null){

tweet = (String) jsonObject.get("text");

timeZone = (String) userObject.get("location");

id = (Long) jsonObject.get("id");

Emotion = "No Emotion";

// output= "No Emotion"+" "+" "+tweet;

while(tweet.contains("\n")){

tweet =tweet.replace("\n", " ");

}

if( timeZone == null ){

timeZone = "Default";

}

if ( tweet.contains("great")|| tweet.contains("good")||tweet.contains("amazing")||tweet.contains("happy")|| tweet.contains("Wondeful")||tweet.contains("awesome"))

{

Emotion = "Happy";

//word.set(timeZone+" Happy");

word.set("[{\"id\":\""+timeZone+"\",\"Emotion\":\""+Emotion+"\"");

// context.write(word, one);

context.write(word, one);

}

else if (((tweet.contains("feeling")||tweet.contains("feel"))&& tweet.contains("not")) || (tweet.contains("sick")|| tweet.contains("sad")||tweet.contains("lonely")))

{

Emotion = "Sad";

word.set("[{\"id\":\""+timeZone+"\",\"Emotion\":\""+Emotion+"\"");

context.write(word, one);

}else {

Emotion= "No Emotion";

}

if(!Emotion.equals("No Emotion") && !timeZone.equals("Default")){

output = "[{\"id\":\""+timeZone+" Sad\",\"Emotion\":\""+Emotion+"\",\"Location\":\""+timeZone+"\"}]" ;

System.out.println(output);

// word.set(output);

// one.set("Hi");

// context.write(one, word);

}

}

}

}

public static class Reduce

extends Reducer<Text,IntWritable,Text,Text> {

private IntWritable result = new IntWritable();

private Text value =new Text();

public void reduce(Text key, Iterable<IntWritable> values,

Context context

) throws IOException, InterruptedException {

int sum = 0; // initialize the sum for each keyword

for (IntWritable val : values) {

sum += val.get();

}

result.set(sum);

value.set(",\"Count\":\""+sum+"\"}]");

context.write(key,value);

}

}

// Driver program

public static void main(String[] args) throws Exception {

Configuration conf = new Configuration();

// String[] otherArgs = new GenericOptionsParser(conf, args).getRemainingArgs(); // get all args

if (args.length != 2) {

System.err.println("Usage: WordCount <in> <out>");

System.exit(2);

}

System.out.println("I am in main thread");

// create a job with name "wordcount"

System.out.println("Before new Job");

Job job = new Job(conf, "wordcount");

System.out.println("after new Job");

System.out.println("Before set jar by class");

job.setJarByClass(WordCount.class);

System.out.println("after set ajr by class");

System.out.println("Before set mapper class");

job.setMapperClass(Map.class);

System.out.println("after set mapper class ");

job.setReducerClass(Reduce.class);

// uncomment the following line to add the Combiner

// job.setCombinerClass(Reduce.class);

// set output key type

job.setOutputKeyClass(Text.class);

// set output value type

job.setOutputValueClass(IntWritable.class);

//set the HDFS path of the input data

FileInputFormat.addInputPath(job, new Path(args[0]));

// set the HDFS path for the output

FileOutputFormat.setOutputPath(job, new Path(args[1]));

//Wait till job completion

System.exit(job.waitForCompletion(true) ? 0 : 1);

}

}

**The index Html file :**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>Trending Topics</title>

<link href="css/bootstrap.min.css" rel="stylesheet">

<style>

footer {

position: fixed;

height: 100px;

bottom: 0;

width: 100%;

}

</style>

<script type='text/javascript' src='https://www.google.com/jsapi'></script>

<script type='text/javascript'>

google.load('visualization', '1', {'packages': ['geochart']});

google.setOnLoadCallback(drawRegionsMap);

function drawRegionsMap() {

var data = google.visualization.arrayToDataTable([

['Country', 'HappyTweets'],

['UAE',43],

['Japan',60],

['America',2243],

['Australia',154],

['Newzealand',58],

['Jakarta',553],

['China',1251],

['Mexico',155],

['India',19],

['United Kingdom',463],

['Spain',49],

['Greenland',168],

]);

var options = {};

var chart = new google.visualization.GeoChart(document.getElementById('chart\_div'));

chart.draw(data, options);

};

</script>

<

</head>

<body>

<div class="container-fluid">

<div class="jumbotron" style="padding-top: 6px;padding-bottom: 6px;color: #FFFFFF;background-color:#61BAF6" >

<h1>World Sentiment Analysis </h1>

<p>the map displays the number of happy tweets from different countries, we can see the how people in America and China are much happier than the rest of the world. China has pure numbers and America has a huge social media presence </p>

</div>

</div>

<div class="container">

<div class="row">

<div id="chart\_div"></div>

</div>

</div>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.0/jquery.min.js"></script>

<script src="js/bootstrap.min.js"></script>

</body>

</html>

Referene:

<https://portal.futuregrid.org/manual/hadoop-wordcount>

<https://developers.google.com/chart/interactive/docs/gallery/geomap>

<https://github.com/uwescience/datasci_course_materials/blob/master/assignment1/twitterstream.py>