INF 553 Homework 5

My two scala codes, were saved in a package called Georgios.HW5. As a result in front of each task, I have to specify the package. Thus the codes to run the three tasks successfully are the following:

Problem1:
spark-submit --class Georgios.HW5.TwitterStreaming Twitter-assembly-1.0.jar

Problem2:
spark-submit --class Georgios.HW5.BloomFilter Twitter-assembly-1.0.jar

First you have to cd where the jar file is and then use the commands provided to run the two tasks. I have the data needed. Additionally, I have some more files. In order to use the twitter streaming, I imported three external jars, which I included in my Solution files. The jars are twitter4j_core, twitter4j_stream and dstream-twitter.
Also, I have included a twitter.txt file, which is one folder 'behind' the jar file. That's how I wrote the code so I had to include it one folder before the jar. This text file has the consumer key, access tokens and everything needed for twitter streaming. I did it this way instead of typing them out in my scala code.

Output:
For the first problem the output that is printed is the same as the one provided in the example:

```
The number of the twitter from beginning: 133
Top 5 hot hashtags:
data:4
RT:2
privacy:2
PwC:2
webcrawling:1
The average length of the twitter is: 104.27
```

For the second problem the output that is printed is the following:

```
Bloom Filter estimated correctly that tweet might be present: 3
Bloom Filter estimated correctly that tweet won't be present: 4
Bloom Filter estimated incorrectly that tweet might be present: 3
Count: 10

Bloom Filter estimated correctly that tweet might be present: 0
Bloom Filter estimated correctly that tweet won't be present: 1
Bloom Filter estimated incorrectly that tweet might be present: 1
Count: 2
```

The first line is the correct estimates of the algorithm and the third line are the false positives. The second line is the 'estimates' that the tweets will not be presented. Bloom filter has no false negatives, so the second line is not really necessary, I just put it there to make the results look better. Since Count is the number of new hashtags received, you can see how all new hashtags are 'described' by the Bloom filter algorithm.

Implementation:
The first problem implementation is pretty straight forward, so I will mostly explain the second one.

For the bloom filter algorithm, I use a bit vector of length 100, and three hash functions. Firstly I saved all the hashtags in a map(hashtagsMapNew), and to each hashtag, I computed their hash values using the 3 hash functions that I implemented. I also have a hashtagsMapStart, where I will add the new hashtags I received at the end of the loop.

I created two bitVectors, where I add 1 to the index of each hashtag. I find this index by the hash function.
I create a bitVectorsStart, where I will make them both equal at the end of the loop.

Also, I created two FalsePositives Lists, where the first one is the comparison of the bitvectors (what the Bloom filter basically does). The second one is the FalsePositivesSure, where I compare the two hashtagsMaps I created.
The first FalsePositive List has values of “M” and “N” for every hashtag. Meaning “maybe” and “no”.
The second FalsePositive List has values of “Y” and “N” for every hashtag. Meaning “yes” and “no”.

Then I compare the two False Positive Lists, to see how well the Bloom Filter algorithm estimated, and what the true results are.
A false positive is when FalsePositiveList has “M” and the Second one has “N”. Meaning that the bloom filter estimated that maybe it will appear, but from the FalsePositiveSure List we can see that the hashtag did not appear before. (third line of my output)
When FalsePositive List has “M” and the FalsePositiveSure has “Y” , that’s a correct estimating from the algorithm. (first line of my output)
When it is “N” and “N”, that’s the correct ‘estimation’ of the hashtag not being present(second line of my output)

OutputFiles:
I know we do not have to provide anything, but I just provided two text files with a sample output from each task.

Note:
I used spark 2.2.1, scala 2.11.0. I used sbt assembly to be able to create a jar file that works, that’s why the name of my jar is Twitter-assembly-1.0.jar. Also when I run the jar through the command, I get a lot of exception errors, but you can see the results as well. If you press one time Ctrl+C, I start seeing the results with less exceptions, and then a question about terminating the program.
I could have included the project, src, main ,target folders which were created in order to create the jar file, but I thought it wasn’t necessary.

Name: Georgios Iliadis

3668057286