DS/CMPSC 410 Sparing 2021

Instructor: Professor John Yen

TA: Rupesh Prajapati and Dongkuan Xu

Lab 3: Filtering and Top Hashtags/Twitters in Tweets

The goals of this lab are for you to be able to

- Implement filtering in a data stream in Spark
- Reverse a key-value pair
- Sort a Key Value Pairs RDD by keys
- Filter a Key Value Pairs RDD (by key)
- Apply the above to find top hashtags in a set of tweets

Total Number of Exercises:

- Exercise 1: 5 points
- Exercise 2: 10 points
- Exercise 3: 15 points ## Total Points: 30 points

Due: midnight, February 7, 2021

The first thing we need to do in each Jupyter Notebook running pyspark is to import pyspark first.

In [39]:
```
import pyspark
```

Once we import pyspark, we need to import an important object called "SparkContext". Every spark program needs a SparkContext object

In [40]:
```
from pyspark import SparkContext
```

We then create a Spark Context variable. Once we have a spark context variable, we can execute spark codes.

In [41]:
```
sc=SparkContext("local", "Lab3")
```

In [38]:
```
sc.stop()
```

Exercise 1 (5 points) (a) Add your name below AND (b) replace the path below with the path of your home directory.

Answer for Exercise 1

- a: Your Name:Kangdong Yuan

In [42]:
```
text_RDD = sc.textFile("/storage/home/kky5082/ds410/lab3/TweetsClimateChangeSentiment .csv")
text_RDD
```

Out[42]:
```
/storage/home/kky5082/ds410/lab3/TweetsClimateChangeSentiment .csv MapPartitionsRDD[1] at textFile at NativeMethodAcc
essorImpl.java:0
```

In [43]:
```
token_RDD = text_RDD.flatMap(lambda line: line.strip().split(" "))
```

Filtering an RDD

The syntax for filter (one type of data trasnformation in spark) is

RDD.filter(lambda parameter : condition)

Notice the syntax is not what is described in p. 38 of the textbook.

The result of filtering the input RDD is the collection of all elements that pass the filter condition (i.e., returns True when the filtering condition is applied to the parameter.

For example, the filtering condition in the pyspark conde below checks whether each element of the input RDD (i.e., token_RDD) starts with the character "#", using Python startswith() method for string.

In [44]:
```
hashtag_RDD = token_RDD.filter(lambda token : token.startswith("#"))
hashtag_RDD
```

Out[44]:
```
PythonRDD[2] at RDD at PythonRDD.scala:53
```

In [45]:
```
hashtag_count_RDD = hashtag_RDD.map(lambda hashtag: (hashtag, 1))
hashtag_count_RDD
```

Out[45]:
```
PythonRDD[3] at RDD at PythonRDD.scala:53
```

In [46]:
```
hashtag_total_RDD = hashtag_count_RDD.reduceByKey(lambda a, b: a + b, 1)
hashtag_total_RDD
```

Out[46]:
```
PythonRDD[8] at RDD at PythonRDD.scala:53
```

In [47]:
```
total_hashtag_RDD = hashtag_total_RDD.map(lambda x: tuple(reversed(x)) )
total_hashtag_RDD
```

Out[47]:
```
PythonRDD[9] at RDD at PythonRDD.scala:53
```

In [48]:
```
sorted_total_hashtag_RDD = total_hashtag_RDD.sortByKey(ascending=False)
```

Exercise 2 (10 points) Complete the code below to obtain hashtags that occured at least n time in this set of tweets. You can choose n to be any integer.

In [49]:
```
n = 5
top_count_hashtags_RDD = sorted_total_hashtag_RDD.filter(lambda x: x[0]>n )
```

In [50]:
```
top_count_hashtags_RDD.collect()
```

Out[50]:
```
[(81, '#climatechange'),
 (45, '#ClimateChange'),
 (16, '#IPCC'),
 (14, '#HurricaneMichael'),
 (13, '#GlobalWarming'),
 (12, '#auspol'),
 (10, '#science'),
 (9, '#climate'),
 (9, '#globalwarming'),
 (8, '#ClimateChangeIsReal'),
 (6, '#ClimateAction'),
 (6, '#Michael'),
 (6, '#climateaction')]
```

Exercise 3 (15 points)

Complete pyspark code below to

- (a) Compute total counts of all hashtags in the vaccination_tweets (5 points)
- (b) Sort the count of hashtags in descending order. (5 points)
- (c) Save all hashtags that have occured at least 10 times. (5 points)

Code for Exercise 3:

In [51]:
```
text2_RDD = sc.textFile("/storage/home/kky5082/ds410/lab3/vaccination_tweets_2 .csv")
text2_RDD
```

Out[51]:
```
/storage/home/kky5082/ds410/lab3/vaccination_tweets_2 .csv MapPartitionsRDD[12] at textFile at NativeMethodAccessorIm
pl.java:0
```

In [52]:
```
token2_RDD = text2_RDD.flatMap(lambda line: line.strip().split(" "))
token2_RDD
```

Out[52]:
```
PythonRDD[13] at RDD at PythonRDD.scala:53
```

In [53]:
```
hashtag2_RDD = token2_RDD.filter(lambda token : token.startswith("#"))
hashtag2_RDD
```

Out[53]:
```
PythonRDD[14] at RDD at PythonRDD.scala:53
```

In [54]:
```
hashtag_count2_RDD = hashtag2_RDD.map(lambda hashtag: (hashtag, 1))
hashtag_count2_RDD
```

Out[54]:
```
PythonRDD[15] at RDD at PythonRDD.scala:53
```

In [55]:
```
hashtag_total2_RDD = hashtag_count2_RDD.reduceByKey(lambda a, b: a + b, 1)
hashtag_total2_RDD
```

Out[55]:
```
PythonRDD[20] at RDD at PythonRDD.scala:53
```

In [56]:
```
total_hashtag2_RDD = hashtag_total2_RDD.map(lambda x: tuple(reversed(x)) )
total_hashtag2_RDD
```

Out[56]:
```
PythonRDD[21] at RDD at PythonRDD.scala:53
```

In [57]:
```
sorted_total2_hashtag_RDD = total_hashtag2_RDD.sortByKey(ascending=False)
```

In [58]:
```
n = 10
top_count2_hashtags_RDD = sorted_total2_hashtag_RDD.filter(lambda x: x[0]>n )
```

In [59]:
```
top_count2_hashtags_RDD.collect()
```

Out[59]:
```
[(2024, '#PfizerBioNTech'),
 (480, '#COVID19'),
 (333, '#vaccine'),
 (317, '#CovidVaccine'),
 (189, '#Pfizer'),
 (133, '#PfizerBioNTech…'),
 (125, '#Moderna'),
 (111, '#coronavirus'),
 (96, '#PfizerVaccine'),
 (73, '#vaccination'),
 (68, '#Covid19'),
 (67, '#AstraZeneca'),
 (64, '#Pfizervaccine'),
 (61, '#vaccines'),
 (55, '#NHS'),
 (53, '#COVID19Vaccine'),
 (50, '#COVID19…'),
 (48, '#PfizerCovidVaccine'),
 (48, '#BLM'),
 (47, '#Pfizer…'),
 (46, '#Vaccine'),
 (44, '#Covid_19'),
 (41, '#COVIDVaccination'),
 (37, '#Diabetes'),
 (35, '#ItsNotJustCovid'),
 (35, '#ContinuityOfCare'),
 (35, '#3.5%'),
 (34, '#FBPE,Cornwall'),
 (34, '#RejoinEU,'),
 (34, '#ProEU'),
 (34, '#GTTO",7/18/11'),
 (33, '#covid19'),
 (31, '#BioNTech'),
 (30, '#vaccinated'),
 (28, '#vaccine…'),
 (28, '#Dubai'),
 (27, '#pfizerbiontech'),
 (26, '#EU'),
 (26, '#Israel'),
 (26, '#news'),
 (25, '#PatientsAtTheCentre'),
 (24, '#Iran'),
 (24, '#CovidVaccine…'),
 (24, '#UK'),
 (24, '#mRNA'),
 (24, '#COVID'),
 (23, '#2'),
 (23, '#Canada'),
 (23, '#FBPE'),
 (23, '#RejoinEU'),
 (22, '#oxfordastrazeneca'),
 (21, '#CoronavirusVaccine'),
 (21, '#COVID19vaccine'),
 (21, '#CoronaVaccine'),
 (19, '#US'),
 (19, '#covid'),
 (19, '#Qatar'),
 (19, '#modernavaccine'),
 (19, '#covidvaccines'),
 (19, '#Norway'),
 (18, '#VaccinesWork'),
 (18, '#1'),
 (18, '#PfizerVaccine…'),
 (17, "#PfizerBioNTech's"),
 (17, "#PfizerBioNTech,['PfizerBioNTech'],Twitter"),
 (17, '#Asia'),
 (17, '#PfizerCOVIDvaccine'),
 (16, '#Doha,7/25/09'),
 (16, '#WHO'),
 (15, '#technology'),
 (15, '#PfizerBioNTech.'),
 (15, '#Sinovac'),
 (15, '#tech'),
 (15, '#covidvacccine'),
 (15, '#COVIDvaccines'),
 (15, '#GTTO,,"With'),
 (14, '#coronavirus…'),
 (14, '#COVID—19'),
 (14, '#Emirati'),
 (14, '#Politics'),
 (13, '#SARSCoV2'),
 (13, '#FDA'),
 (13, '#HumanRights'),
 (13, '#Coronavirus'),
 (13, '#digital'),
 (13, '#Moderna…'),
 (13, '#counterTerrorism,'),
 (13, '#الله_ثم_الوطن_ثم_رئيس_الدولة'),
 (13, '#Covid'),
 (13, '#vaccinations'),
 (13, '#vaccines.…'),
 (12, '#SputnikV'),
 (12, '#Pfizervaccine…'),
 (12, '#vaccine,'),
 (12, '#healthcare'),
 (12, '#FMcy",9/29/10'),
 (12, '#Chronoptimist'),
 (12, '#TeamGP💙Trainer'),
 (12, '#Covid19UK'),
 (12, '#vaccine.'),
 (12, '#USA'),
 (12, '#COVID20'),
 (12, '#Vaccin'),
 (12, '#PfizerBioNTech,'),
 (12, '#OxfordVaccine'),
 (12, '#FBPE🇪🇺🇬🇧🇭🇺,Earth,"""It'),
 (12, '#History'),
 (12, '#ElectoralReform'),
 (12, '#WHUFC",9/22/17'),
 (11, '#oxfordvaccine'),
 (11, '#Turkey'),
 (11, '#littleBRIC:'),
 (11, '#GetVaccinated'),
 (11, '#PfizerBioNTech",[\'PfizerBioNTech\'],Twitter')]
```

In [60]:
```
output_path = "/storage/home/kky5082/ds410/lab3/Lab3_ouput_top_hashtag.txt"
top_count2_hashtags_RDD.saveAsTextFile(output_path)
```