CS246Colab1

September 15, 2020

1 CS246 - Colab 1

1.1 Wordcount in Spark

1.1.1 Setup

Let's setup Spark on your Colab environment. Run the cell below!

[167]:
```
!pip install pyspark
!pip install -U -q PyDrive
!apt install openjdk-8-jdk-headless -qq
import os
os.environ["JAVA_HOME"] = "/usr/lib/jvm/java-8-openjdk-amd64"
```

Now we authenticate a Google Drive client to download the file we will be processing in our Spark job.

Make sure to follow the interactive instructions.

[2]:
```
from pydrive.auth import GoogleAuth
from pydrive.drive import GoogleDrive

from google.colab import auth
from oauth2client.client import GoogleCredentials

# Authenticate and create the PyDrive client
auth.authenticate_user()
gauth = GoogleAuth()
gauth.credentials = GoogleCredentials.get_application_default()
drive = GoogleDrive(gauth)
```

[3]:
```
id='1SE6k_0YukzGd5wK-E4i6mG83nydlfvSa'
downloaded = drive.CreateFile({'id': id})
downloaded.GetContentFile('pg100.txt')
```

If you executed the cells above, you should be able to see the file *pg100.txt* under the "Files" tab on the left panel.

1.1.2 Your task

If you run successfully the setup stage, you are ready to work on the *pg100.txt* file which contains a copy of the complete works of Shakespeare.

Write a Spark application which outputs the number of words that start with each letter. This means that for every letter we want to count the total number of (non-unique) words that start with a specific letter. In your implementation **ignore the letter case**, i.e., consider all words as lower case. Also, you can ignore all the words **starting** with a non-alphabetic character.

[4]:
```
from pyspark.sql import *
from pyspark.sql.functions import *
from pyspark import SparkContext
import pandas as pd

# create the Spark Session
spark = SparkSession.builder.getOrCreate()

# create the Spark Context
sc = spark.sparkContext
```

[5]:
```
spark
```

[5]:
```
<pyspark.sql.session.SparkSession at 0x7f5c11481a20>
```

Spark Examples

[66]:
```
text_file = sc.textFile("pg100.txt")
```

[120]:
```
text_file
```

[120]:
```
pg100.txt MapPartitionsRDD[644] at textFile at NativeMethodAccessorImpl.java:0
```

[82]:
```
text_file.count()
```

[82]:
```
124787
```

[97]:
```
counts = text_file.flatMap(lambda line: line.split(" ")) \
             .map(lambda word: (str.lower(word), 1)) \
             .reduceByKey(lambda a, b: a + b)
```

[98]:
```
counts.count()
```

[98]:
```
59723
```

[99]:
```
counts.take(10)
```

[99]:
```
[('project', 320),
 ('gutenberg', 250),
```

```
 ('ebook', 13),
 ('of', 18126),
 ('shakespeare', 270),
 ('', 506610),
 ('this', 5930),
 ('is', 9168),
 ('use', 509),
 ('anyone', 5)]
```

[135]:
```
countsByFirstLetter = counts.map(lambda pair: (pair[0][0], pair[1]) if pair[0] !
↪= '' else (pair[0], pair[1])).reduceByKey(lambda a, b : a+b)
```

[141]:
```
countsByFirstLetter
```

[141]:
```
PythonRDD[737] at RDD at PythonRDD.scala:53
```

[159]:
```
df = spark.createDataFrame(countsByFirstLetter, ['letter', 'counts'])
```

[160]:
```
df_pd = df.toPandas()
```

[163]:
```
df_pd = df_pd[("a" <= df_pd.letter) & (df_pd.letter <= "z")].
↪set_index("letter").sort_index()
```

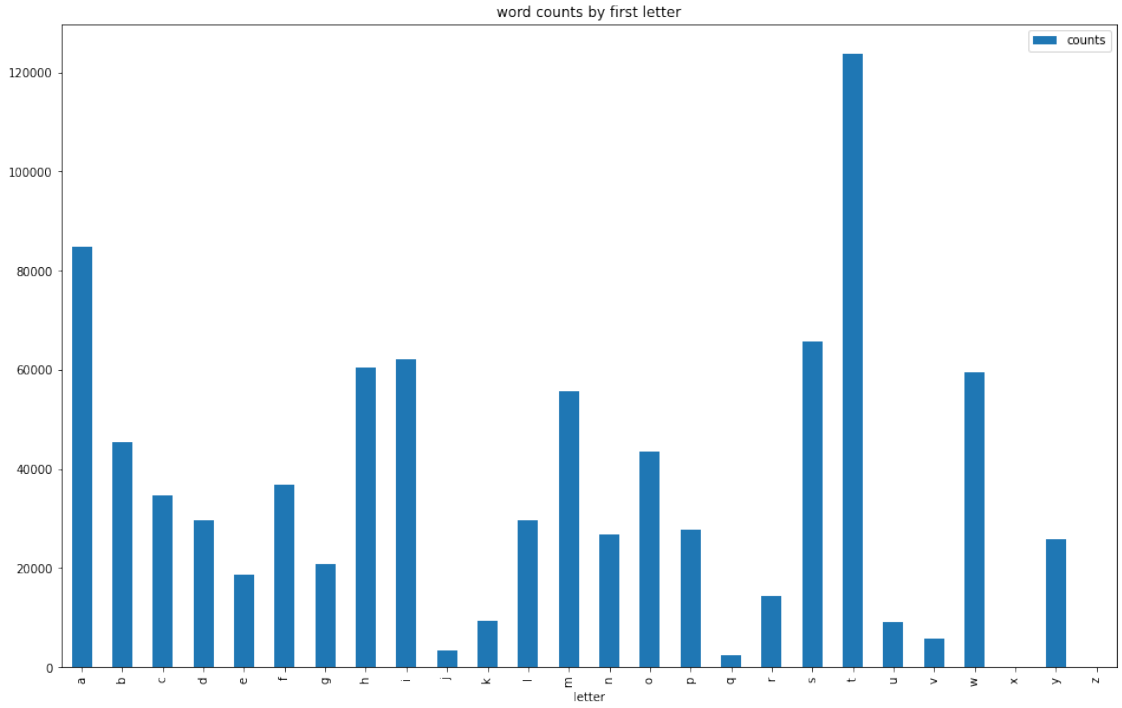
[164]:
```
df_pd.head()
```

[164]:

letter	counts
a	84836
b	45455
c	34567
d	29713
e	18697

[166]:
```
df_pd.plot(kind = "bar", figsize = (16,10), title = "word counts by first␣
↪letter")
```

[166]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f5c0931b908>
```



Once you obtained the desired results, **head over to Gradescope and submit your solution for this Colab**!