VocalesV1

March 30, 2020

1 CONTEO DE VOCALES CON PYSPARK

1.1 Nicolás Patalagua

1.1.1 Universidad Sergio Arboleda

1.1.2 Infraestructura para BigData

2 Spark

Spark se ha incorporado herramientas de la mayoría de los científicos de datos. Es un framework open source para la computación en paralelo utilizando clusters. Se utiliza especialmente para acelerar la computación iterativa de grandes cantidades de datos o de modelos muy complejos.

[0]:
```
#Realizamos la instalación de todas las librerias y dependencias a ser usadas.
!apt-get install openjdk-8-jdk-headless -qq > /dev/null
!wget -q http://apache.osuosl.org/spark/spark-2.4.5/spark-2.4.5-bin-hadoop2.7.
 ↪tgz
!tar xf spark-2.4.5-bin-hadoop2.7.tgz
!pip install -q findspark
!pip install pyspark
```

```
Collecting pyspark
  Downloading https://files.pythonhosted.org/packages/9a/5a/271c416c1c2185
b6cb0151b29a91fff6fcaed80173c8584ff6d20e46b465/pyspark-2.4.5.tar.gz (217.8MB)
     |                | 217.8MB 62kB/s
Collecting py4j==0.10.7
  Downloading https://files.pythonhosted.org/packages/e3/53/c737818eb9a7dc
32a7cd4f1396e787bd94200c3997c72c1dbe028587bd76/py4j-0.10.7-py2.py3-none-any.whl
(197kB)
     |                | 204kB 36.4MB/s
Building wheels for collected packages: pyspark
  Building wheel for pyspark (setup.py) … done
  Created wheel for pyspark: filename=pyspark-2.4.5-py2.py3-none-any.whl
size=218257927
sha256=1724f14836d813c498b7afeb4353bb3a12a8059787882f33331a04581fed7d34
  Stored in directory: /root/.cache/pip/wheels/bf/db/04/61d66a5939364e756eb1c1be
```

```
4ec5bdce6e04047fc7929a3c3c
Successfully built pyspark
Installing collected packages: py4j, pyspark
Successfully installed py4j-0.10.7 pyspark-2.4.5
```

[0]:
```
#Nos solicitaran un codigo de acceso
import os
os.environ["JAVA_HOME"] = "/usr/lib/jvm/java-8-openjdk-amd64"
os.environ["SPARK_HOME"] = "/content/spark-2.4.5-bin-hadoop2.7"
from google.colab import drive
drive.mount('/content/gdrive')
from pyspark import SparkConf, SparkContext
conf = SparkConf().setAppName("app")
sc = SparkContext.getOrCreate();
```

```
Drive already mounted at /content/gdrive; to attempt to forcibly remount, call
drive.mount("/content/gdrive", force_remount=True).
```

[0]:
```
#La siguiente URL contiene el archivo que vamos a procesar para el ejercicio.
!wget http://textfiles.com/stories/100west.txt
```

```
--2020-03-30 20:47:39--  http://textfiles.com/stories/100west.txt
Resolving textfiles.com (textfiles.com)… 208.86.224.90
Connecting to textfiles.com (textfiles.com)|208.86.224.90|:80… connected.
HTTP request sent, awaiting response… 200 OK
Length: 20839 (20K) [text/plain]
Saving to: ‘100west.txt.1’

100west.txt.1       100%[===================>]  20.35K   112KB/s    in 0.2s

2020-03-30 20:47:40 (112 KB/s) - ‘100west.txt.1’ saved [20839/20839]
```

[0]:
```
#Seleccionamos el archivo a ser procesado, el cual obtuvimos de la URL anterior.
data = sc.textFile("100west.txt")
#Mostramos 10 lineas del archivo.
data.take(10)
```

[0]:
```
['',
 '',
 '                      THIS IS A SHAREWARE TRIAL PROJECT',
 '                                    ',
 '          IT IS NOT "FREEWARE" WE NEED YOUR SUPPORT TO CONTINUE',
 '',
 '',
 '',
 '',
 '                                100 WEST BY 53 NORTH']
```

[0]:
```
#Asignamos a palabras la division por palabras del texto.
palabras = data.flatMap(lambda data : data.split())
```

[0]:
```
#Mostramos las primeras palabras obtenidas
palabras.take(10)
```

[0]:
```
['THIS',
 'IS',
 'A',
 'SHAREWARE',
 'TRIAL',
 'PROJECT',
 'IT',
 'IS',
 'NOT',
 '"FREEWARE"']
```

[0]:
```
#Realizamos la normalización del documento, reemplazando caracteres especiales.
linea = data.flatMap(lambda linea : linea.replace(" ","").replace(".","").
 ↪replace('"','').replace("'",'').replace("ñ","n").replace("ú","u").
 ↪replace("ó","o")
.replace("í","i").replace("é","e").replace("á","a").replace("!","").
 ↪replace(",","").lower().split())
```

[0]:
```
#Mostramos 10 datos del archivo una vez normalizado
linea.take(10)
```

[0]:
```
['thisisasharewaretrialproject',
 'itisnotfreewareweneedyoursupporttocontinue',
 '100westby53north',
 'by',
 'jimprentice',
 'copyright1990jimprenticebrandonmanitobacanada',
 'northof53amagicphrasespokenmumbledorthought',
 'inwardlybythousandsofsoulsventuringnorthwardan',
 'imaginarylineshownonlyonmapsandlabelled53degrees',
 'itspresenceindicatedtohighwaytravellersbyroadside']
```

[0]:
```
#Asignamos a la variable letras un recorrido por cada linea del archivo
letras = linea.map(lambda linea: (linea,1))
```

[0]:
```
#Mostramos las 10 primeras lineas con un valor de 1
letras.take(10)
```

[0]:
```
[('thisisasharewaretrialproject', 1),
 ('itisnotfreewareweneedyoursupporttocontinue', 1),
 ('100westby53north', 1),
```

```
 ('by', 1),
 ('jimprentice', 1),
 ('copyright1990jimprenticebrandonmanitobacanada', 1),
 ('northof53amagicphrasespokenmumbledorthought', 1),
 ('inwardlybythousandsofsoulsventuringnorthwardan', 1),
 ('imaginarylineshownonlyonmapsandlabelled53degrees', 1),
 ('itspresenceindicatedtohighwaytravellersbyroadside', 1)]
```

[0]:
```
#Asignamos a la varible lista cada una de las lineas
lista = letras.collect()
```

[0]:
```
#Imprimimos una de las lineas
print(lista[4][0])
```

```
jimprentice
```

[0]:
```
#Definimos una funciión para contar vocales en el archivo
def VowelsU(cad):
  suma = sum(c in {"a", "A", "e", "E", "i", "I", "o", "O", "u", "U"} for c in␣
 ↪cad)
  a= sum(c in{"a"} for c in cad)
  e= sum(c in{"e"} for c in cad)
  i= sum(c in{"i"} for c in cad)
  o= sum(c in{"o"} for c in cad)
  u= sum(c in{"u"} for c in cad)
  return [a,e,i,o,u]
```

[0]:
```
#Asignamos a una lista la cantidad de repeticiones de caa vocal
a1=0
e1=0
i1=0
o1=0
u1=0
lst = [VowelsU(x) for x in letras.keys().collect()]
for i in range(len(lst)-1):
  a1+=lst[i][0]
  e1+=lst[i][1]
  i1+=lst[i][2]
  o1+=lst[i][3]
  u1+=lst[i][4]
```

[57]:
```
#Imprimimos el valor de repeticiones de cada vocal en el documento
print("Cantidad de A: "+str(a1))
print("Cantidad de E: "+str(e1))
print("Cantidad de I: "+str(i1))
print("Cantidad de O: "+str(o1))
print("Cantidad de U: "+str(u1))
```

```
Cantidad de A: 1118
Cantidad de E: 1841
Cantidad de I: 982
Cantidad de O: 1093
Cantidad de U: 368
```

[55]:
```
suma = a1+e1+i1+o1+u1
print("Promedio de A por linea: "+str(suma/a1))
print("Promedio de E por linea: "+str(suma/e1))
print("Promedio de I por linea: "+str(suma/i1))
print("Promedio de O por linea: "+str(suma/o1))
print("Promedio de U por linea: "+str(suma/u1))
```

```
a: 4.831842576028622
e: 2.9342748506246603
i: 5.5010183299389
o: 4.942360475754803
u: 14.679347826086957
```