Assignment – 3 : Rohitha Gutta

#!/anaconda3/bin/python

#-----------------------------------------------------

# This is Assignment - 3: Rohitha\_Gutta

# The goal is to show top-5 words, top-5 anagrams, top-5 bigrams for a given input file

#------------------------------------------------------

from \_\_future\_\_ import print\_function

import sys

from pyspark.sql import SparkSession

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

if len(sys.argv) != 2:

print("Usage: assignment3.py <input-file>", file=sys.stderr)

exit(-1)

spark = SparkSession\

.builder\

.appName("Assignment3")\

.getOrCreate()

# sys.argv[0] is the name of the script.

# sys.argv[1] is the first parameter

input\_path = sys.argv[1]

# read input and create an RDD<String>

records = spark.sparkContext.textFile(input\_path)

#method to generate cleaned words from each record by convert all words to lowercase and remove special character at beginning and ending of each word. Also remove words with length less than 4

def cleanData(record):

word\_list=list()

#converts each record to lowercase

trimed\_record=record.strip().lower()

specialChar=("-", "\_", ",", ".", "?", ";", "/", "%", "!", "#", "^", "@", "&", "'", "(", ")", "[","]", ":", "{", "}")

#tokenize record

record\_tokens=trimed\_record.split()

for word in record\_tokens:

#remove special character at beginning

if word.startswith(specialChar):

word=word[1:len(word)]

#remove special character ending of each word

if word.endswith(specialChar):

word=word[:len(word)-1]

#remove words with length less than 4

if len(word)>=4:

word\_list.append(word)

return word\_list

#cleaned words RDD

cleaned\_records=records.flatMap(cleanData)

#TO FIND TOP-5 WORDS

# create a pair of (word, 1) for all words

pairs\_rdd=cleaned\_records.map(lambda word: (word, 1))

# aggregate the frequencies of each unique word

word\_frequency=pairs\_rdd.reduceByKey(lambda a, b: a + b)

# frequencies of words in decending order

top\_words=word\_frequency.map(lambda a: (a[1],a[0])).sortByKey(False).map(lambda a: (a[1],a[0]))

#print RDD with top-5 words

print("top-5 words : ", top\_words.take(5))

#TO FIND TOP-5 ANAGRAMS

# used cleaned words RDD and sorted each word

sorted\_words=cleaned\_records.map(lambda a: ''.join(sorted(a)))

# create a pair of (sorted\_word, 1) for all words

anagrams\_pairs=sorted\_words.map(lambda word: (word, 1))

# aggregate the frequencies of each unique sorted word

anagrams\_frequency=anagrams\_pairs.reduceByKey(lambda a, b: a + b)

# frequencies of sorted word in decending order

top\_anagrams=anagrams\_frequency.map(lambda a: (a[1],a[0])).sortByKey(False).map(lambda a: (a[1],a[0]))

#print RDD with top-5 anagrams

print("top-5 anagrams : ", top\_anagrams.take(5))

#TO FIND TOP-5 BIGRAMS

#method to generate bigrams words from each record by convert all words to lowercase and remove special character at beginning and ending of each word

def cleanData\_bigrams(record):

bigrams\_list=list()

#converts each record to lowercase

trimed\_record=record.strip().lower()

specialChar=("-", "\_", ",", ".", "?", ";", "/", "%", "!", "#", "^", "@", "&", "'", "(", ")", "[","]", ":", "{", "}")

#tokenize record

record\_tokens=trimed\_record.split()

for word in record\_tokens:

#remove special character at beginning

if word.startswith(specialChar):

word=word[1:len(word)]

#remove special character ending of each word

if word.endswith(specialChar):

word=word[:len(word)-1]

bigrams\_list.append(word)

return [((bigrams\_list[i],bigrams\_list[i+1]),1) for i in range (0, len(bigrams\_list)-1)]

#cleaned bigrams RDD

bigrams\_rdd=records.flatMap(cleanData\_bigrams)

# aggregate the frequencies of bigrams

bigrams\_frequency=bigrams\_rdd.reduceByKey(lambda a, b: a + b)

# frequencies of bigrams in decending order

top\_bigrams=bigrams\_frequency.map(lambda a: (a[1],a[0])).sortByKey(False).map(lambda a: (a[1],a[0]))

#print RDD with top-5 bigrams

print("top-5 bigrams : ", top\_bigrams.take(5))

# done!

spark.stop()

OUTPUT:

top-5 words :

[('which', 2673),

('that', 2069),

('their', 1529),

('they', 1511),

('with', 1081)]

top-5 anagrams :

[('chhiw', 2673),

('ahtt', 2069),

('ehirt', 1529),

('ehty', 1511),

('hitw', 1081)]

top-5 bigrams :

[(('of', 'the'), 3914),

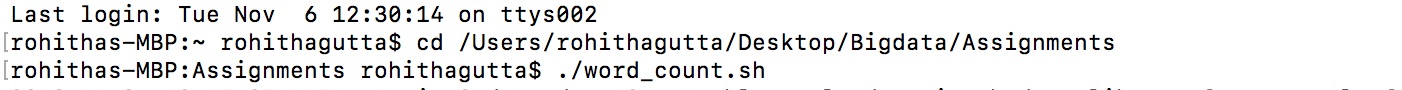
(('in', 'the'), 1539),

(('to', 'the'), 1035),

(('and', 'the'), 783),

(('it', 'is'), 607)]

SCREENSHOTS:



A picture containing text, newspaper

Description automatically generated

A close up of a newspaper

Description automatically generated

A picture containing newspaper, text

Description automatically generated