12/9/2016

TF-IDF

Term Frequency-Inverse Document Frequency Calculation in Apache Spark using HDFS

Hira Fatima | Graduate Student | DS8003
Master of Data Science and Analytics | RYERSON UNIVERSITY

Table of Contents

1.INTRODUCTION

Per Wikipedia, ***"in information retrieval, tf-idf, short for term frequency-inverse document frequency, is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus."*** This document cover calculation of TF-IDF (Term Frequency-Inverse Document Frequency) from a ground up approach without using existing libraries/APIs that automate this calculation. The calculation is then used to query words in a corpus and find the top N documents with the highest score.

This paper starts with a brief description of the goal for this project, provides an outline of the method(s) that were used, layout of the architecture for the model, comparison of SPARK with other technologies, and, a detailed explanation of the code with an example and screenshots (provided at the end of this document). The model is then tested with different values of N for query phrase of more than one word. Outcome of the tests are discussed in the Test & Results section.

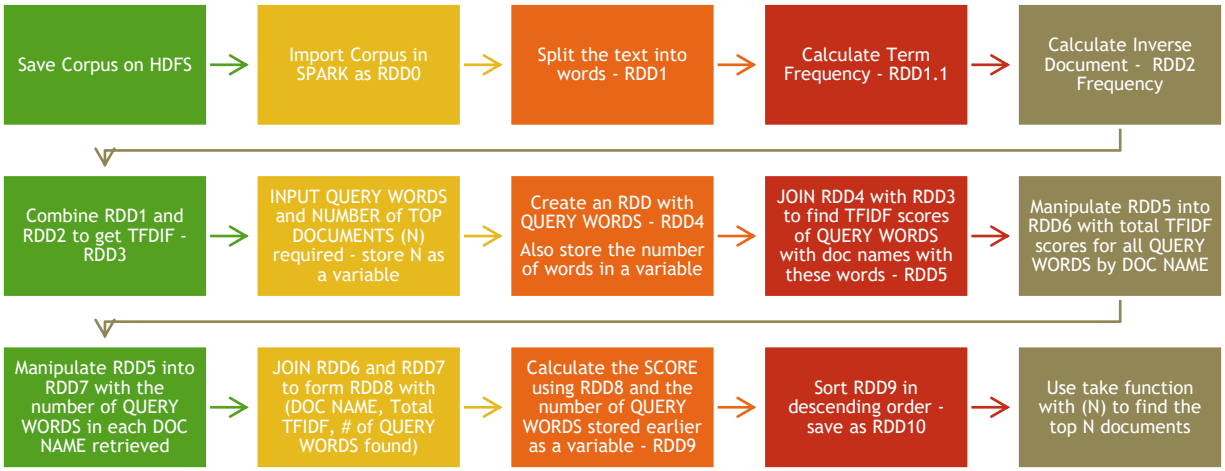
2.GOAL

There were two goals of this project.

1. The first one was to find the optimal technology solution to calculate TFIDF from the given choices listed below:
   MapReduce, Spark, Hive, Pig, HBase, Mongo, HDFS.
2. The second was to write a query to find top N documents with more than 1 words in real-time

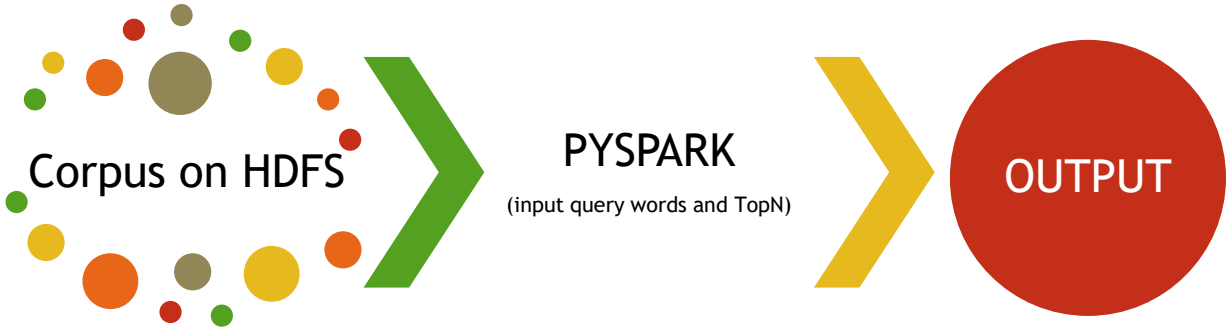
3.METHOD

To achieve these goals, SPARK was used for the entire project. The text files were transferred to HDFS and RDDs in SPARK were used to do the calculations. The entire code is based on working with RDDs in (key, value) pairs. The (key, value) pairs were transformed in various ways to get to the final table containing (Document Name, Score) for a given query. More details are provided in the architecture section.



4.ARCHITECTURE

The corpus was transferred onto HDFS and imported into SPARK as RDD. All computations were done in SPARK with the manipulation and transformation of RDDs from one form to another.



5.WHY SPARK and HDFS?

Christo Wilson, on this website explains in very simple words why HDFS and SPARK together can be a power combination to compute large amounts of data in parallel with very simple code rather quickly.

From working on the assignments, it was obvious that SPARK was easy to use, required minimal code lines and the process of transforming RDDs was quite simple, smooth and fast. There was no need to define schema of tables like in PIG and HIVE, and, also no need to write streaming commands like in MAPREDUCE. I could paste the entire code in PYSPARK and it provided instant results.

6.WHY MY SYSTEM IS THE BEST!

My system that I have built to find the top N documents with a given list of query words using TFIDF calculation is the best because it works in real-time within seconds.

All that needs to be done is to provide the 2 inputs (listed below), copy paste the entire code in PYSPARK and it will provide instant results in real-time.

- Query Words
- N value for TOP N Documents

7.DETAILED CODE with EXPLANATION and EXAMPLE

In this section, the code is provided with a detailed explanation of each step. At a high-level, as explained in Methods section, the entire process revolves around manipulating and transforming the paired RDDs from one from to another. Count() and Collect() functions were used at the end of every step to verify the results and show what the code is actually doing.

From the datasets provided for this project, the TENNIS folder was used here (reference for datasets is provided at the end). "Sharapova game" – these words were used as QUERY WORDS.
The output after each step is also provided to demonstrate what the code is doing. Screenshots are included at the end of this document.

LOAD CORPUS

Step 1

```
#Create RDDS with file path as key and content as values
files = sc.wholeTextFiles("/user/root/finalexam/tennis")
files.collect()[0] # to verify
```

Step 1 – Output

```
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/001.txt', u'Henman overcomes rival Rusedski\n\nTim Henman saved a match point before fighting back to defeat British rival Greg Rusedski 4-6 7-6 (8-6) 6-4 at the Dubai Tennis Championships on Tuesday.\n\nWorld number 46 Rusedski broke in the ninth game to take a tight opening set. Rusedski had match point at 6-5 in the second set tie-break after Henman double-faulted, but missed his chance and Henman rallied to clinch the set. The British number one then showed his superior strength to take the decider and earn his sixth win over Rusedski. Serve was held by both players with few alarms until the seventh game of the final set, when Rusedski\'s wild volley gave Henman a vital break. A furious Rusedski slammed his racket onto the ground in disgust and was warned by the umpire.\n\nHenman, seeded three, then held his serve comfortably thanks to four serve-and-volley winners to take a clear 5-3 lead. Rusedski won his service game but Henman took the first of his three match points with a service winner to secure his place in the second round at Dubai for the first time in three years. It was the first match between the pair for three years - Henman last lost to Rusedski six years ago - and lasted two hours and 40 minutes. The pair are now likely to only face each other on court as rivals - rather than as team-mates - after Henman decided to retire from Davis Cup tennis leaving Rusedski to lead the team out against Israel on 4-6 March. Henman, who now faces Russian Igor Andreev in the last 16, admitted afterwards it was difficult coming up against his compatriot on a fast surface. "You just take it point by point when you\'re fighting to stay in the match," he said. "I had to keep playing aggressively and competing to get a chance. "I now have to recover in time for the next match because the body doesn\'t recover as quick as it used to, especially after two hours and 40 minutes."\n')
```

#TERM FREQUENCY

Step 2

```
#Count the number of documents in corpus and store in variable ndoc for later use
ndoc=files.count()
ndoc
```

Step 2 – Output

```
100
```

Step 3

```
#Split the words in file content
```

```
words = files.flatMapValues(lambda line: line.split( ))
words.collect()[0] # to verify
words.count()
```

Step 4 - Output

```
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/001.txt', u'Henman')
28449
```

Step 4

```
words.keys().collect()[0] # to verify
words.values().collect()[0] # to verify
```

Step 4 – Output

```
u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/001.txt'
u'Henman'
```

Step 5

```
#Reverse the key,value pair such the the word becomes the key and doc name becomes the value
reverse = words.map(lambda fields: (fields[1],fields[0]))
reverse.collect()[0] # to verify
```

Step 5 – Output

```
(u'Henman', u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/001.txt')
```

Step 6

```
#Sort by key - not necessary, but helps in verification of code and analysis
sorted = reverse.sortByKey()
sorted.collect()[0] # to verify
```

Step 6 – Output

```
(u"", u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/080.txt')
```

Step 7

```
#Add 1 to each line/word for word count in each document
count = sorted.map(lambda word: (word,1))
count.collect()[0]
```

Step 7 – Output

```
((u"", u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/080.txt'), 1)
```

Step 8

```
#Add the count to get term frequency
#At this point, the word and doc name together form the key
sum = count.reduceByKey(lambda x,y: x+y)
a=sum.sortByKey()
a.collect()[0] # to verify
a.count()
```

Step 8 - Output

17857

Step 9

```
#Change the layout to make the word key and the rest a tuple of values
aa=a.map(lambda ((x_1, x_2), y): (x_1, (x_2, y)))
aa.keys().collect()[0] # to verify
aa.values().collect()[0] # to verify
```

Step 9 – Output

```
u""
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/080.txt', 1)
```

#INVERSE DOCUMENT FREQUENCY

Step 10

```
#Here we take the previous table that contained word, doc name, and term frequency
#From that table, we keep just the key that consists of word and doc name and get rid of term
frequency
idf = a.keys()
idf.collect()[0]
idf.count()
```

Step 10 – Output

17857

Step 11

```
#We then just keep the words and get rid of the document name as well. This would tell us the number
of document a particular word appeared in without the document names so that we can do a sum
idfkeys=idf.map(lambda line: line[0])
idfkeys.collect()
idfkeys.count()
```

Step 11 – Output

17857

Step 12

```
#Now add 1 to each word for a word count
idfcount = idfkeys.map(lambda word: (word,1))
idfcount.collect()
idfcount.count()
```

Step 12 – Output

17857

Step 13

```
#Reduce by key(word) to get a word count which the document frequency
#The value is the number of documents in which the word appeared in
idfsum = idfcount.reduceByKey(lambda x,y:x+y)
idfsum.collect()[0] to verify
b=idfsum.sortByKey()
b.count()
```

Step 13 – Output

```
4717 to verify
```

Step 14

```
#Now calculate the inverse of document frequency
#Here we use the ndoc that we stored earlier
idfinversed = idfsum.mapValues(lambda y: ndoc/y).sortByKey()
idfinversed.collect() # to verify
#And apply LOG 10 function
import math
idflog = idfinversed.mapValues(lambda y: math.log10(y))
idflog.collect()[0]
```

Step 14 – Output

```
(u"", 2.0)
```

#TERM FREQUENCY - INVERSE DOCUMENT FREQUENCY

Step 15

```
#Join the TF and IDF tables by word to get tfidf table
tfidfjoin = aa.leftOuterJoin(idflog)
tfidfjoin.sortByKey().collect()
tfidfjoin.count()
```

Step 15 – Output

```
(u"", ((u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/080.txt', 1), 2.0))
17857
```

Step 16

```
#The table would be in this format (word,(doc name, TF), IDF) which is converted to (word,
(doc,TF,IDF,TFIDF)) format
tfidffinal = tfidfjoin.map(lambda(w, ((x, y), z)):(w,(x,y,z,z*y)))
tfidffinal.sortByKey().collect()[0]
```

Step 16 – Output

```
(u"", (u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/080.txt', 1, 2.0, 2.0))
```

#TAKE INPUTS FOR QUERY

Step 17

```
#Here is the INPUT for query words
query = sc.parallelize(["Sharapova game"])
query.collect() # to verify
```

Step 17 – Output

```
['Sharapova game']
```

Step 18

```
#topN is the INPUT for how many documents need to be retrieved
topN = 3
```

Step 18 – Output

```
3
```

Step 19

```
#Convert the query words into an RDD of words that can be processed further
#Split the words
querywords = query.flatMap(lambda line: line.split())
querywords.collect()
```

Step 19 – Output

```
['Sharapova', 'game']
```

Step 20

```
#Convert into key,value pair
querywordspair = querywords.map(lambda word: (word,topN))
querywordspair.collect() # to verify
```

Step 20 – Output

```
[('Sharapova', 3), ('game', 3)]
```

Step 21

```
#Store the total number of words queried for calculation later
nofquerywords = querywordspair.count()
nofquerywords
```

Step 21 – Output

```
2
```

Step 22

```
#Now filter out the words queried from the TFIDF table to calculate their scores
filterresults = tfidffinal.join(querywordspair)
filterresults.collect() # to verify
```

Step 22 – Output

[(u'game', ((u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/001.txt', 3,
0.47712125471966244, 1.4313637641589874), 3)), (u'game',
((u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/007.txt', 1,
0.47712125471966244, 0.47712125471966244), 3)), (u'game',
((u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/008.txt', 1,
0.47712125471966244, 0.47712125471966244), 3)), (u'game',
((u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/012.txt', 1,
0.47712125471966244, 0.47712125471966244), 3)), (u'game',
((u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/015.txt', 2,
0.47712125471966244, 0.95424250943932487), 3)), (u'game',
((u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/018.txt', 1,
0.47712125471966244, 0.47712125471966244), 3)), (u'game',
((u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/019.txt', 1,
0.47712125471966244, 0.47712125471966244), 3)), (u'game',
((u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/020.txt', 1,
0.47712125471966244, 0.47712125471966244), 3)), (u'game',
((u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/022.txt', 1,
0.47712125471966244, 0.47712125471966244), 3)), (u'game',
((u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/026.txt', 1,
0.47712125471966244, 0.47712125471966244), 3)), (u'game',
((u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/033.txt', 1,
0.47712125471966244, 0.47712125471966244), 3)), (u'game',
((u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/039.txt', 2,
0.47712125471966244, 0.95424250943932487), 3)), (u'game',
((u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/052.txt', 2,
0.47712125471966244, 0.95424250943932487), 3)), (u'game',
((u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/053.txt', 1,
0.47712125471966244, 0.47712125471966244), 3)), (u'game',
((u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/054.txt', 1,
0.47712125471966244, 0.47712125471966244), 3)), (u'game',
((u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/060.txt', 3,
0.47712125471966244, 1.4313637641589874), 3)), (u'game',
((u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/061.txt', 2,
0.47712125471966244, 0.95424250943932487), 3)), (u'game',
((u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/063.txt', 2,
0.47712125471966244, 0.95424250943932487), 3)), (u'game',
((u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/066.txt', 1,
0.47712125471966244, 0.47712125471966244), 3)), (u'game',
((u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/068.txt', 3,
0.47712125471966244, 1.4313637641589874), 3)), (u'game',
((u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/071.txt', 14,
0.47712125471966244, 6.6796975660752738), 3)), (u'game',
((u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/072.txt', 1,
0.47712125471966244, 0.47712125471966244), 3)), (u'game',
((u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/073.txt', 2,
0.47712125471966244, 0.95424250943932487), 3)), (u'game',
((u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/079.txt', 1,
0.47712125471966244, 0.47712125471966244), 3)), (u'game',
((u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/087.txt', 2,
0.47712125471966244, 0.95424250943932487), 3)), (u'game',
((u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/089.txt', 1,
0.47712125471966244, 0.47712125471966244), 3)), (u'game',
((u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/090.txt', 1,

0.47712125471966244, 0.47712125471966244), 3)), (u'game',
((u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/096.txt', 1,
0.47712125471966244, 0.47712125471966244), 3)), (u'game',
((u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/097.txt', 1,
0.47712125471966244, 0.47712125471966244), 3)), (u'game',
((u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/098.txt', 2,
0.47712125471966244, 0.95424250943932487), 3)), (u'game',
((u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/100.txt', 3,
0.47712125471966244, 1.4313637641589874), 3)), (u'Sharapova',
((u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/047.txt', 1,
1.2041199826559248, 1.2041199826559248), 3)), (u'Sharapova',
((u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/051.txt', 1,
1.2041199826559248, 1.2041199826559248), 3)), (u'Sharapova',
((u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/054.txt', 2,
1.2041199826559248, 2.4082399653118496), 3)), (u'Sharapova',
((u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/056.txt', 1,
1.2041199826559248, 1.2041199826559248), 3)), (u'Sharapova',
((u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/075.txt', 2,
1.2041199826559248, 2.4082399653118496), 3)), (u'Sharapova',
((u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/100.txt', 8,
1.2041199826559248, 9.6329598612473983), 3))]

#CALCULATE SCORE

Step 23

```
#Create an RDD from the TFIDF table with only the document name and TFIDF score per word
#The join gave (word, (doc,TF,IDF,TFIDF),topN) which we now reduce to (doc,TFIDF)
filterresultstfidf1 = filterresults.map(lambda(a, ((b, c, d, e), f)):(b,e))
filterresultstfidf1.collect()
```

Step 23 – Output

[(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/001.txt', 1.4313637641589874),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/007.txt', 0.47712125471966244),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/008.txt', 0.47712125471966244),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/012.txt', 0.47712125471966244),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/015.txt', 0.95424250943932487),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/018.txt', 0.47712125471966244),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/019.txt', 0.47712125471966244),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/020.txt', 0.47712125471966244),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/022.txt', 0.47712125471966244),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/026.txt', 0.47712125471966244),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/033.txt', 0.47712125471966244),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/039.txt', 0.95424250943932487),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/052.txt', 0.95424250943932487),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/053.txt', 0.47712125471966244),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/054.txt', 0.47712125471966244),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/060.txt', 1.4313637641589874),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/061.txt', 0.95424250943932487),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/063.txt', 0.95424250943932487),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/066.txt', 0.47712125471966244),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/068.txt', 1.4313637641589874),

(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/071.txt', 6.6796975660752738),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/072.txt', 0.47712125471966244),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/073.txt', 0.95424250943932487),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/079.txt', 0.47712125471966244),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/087.txt', 0.95424250943932487),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/089.txt', 0.47712125471966244),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/090.txt', 0.47712125471966244),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/096.txt', 0.47712125471966244),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/097.txt', 0.47712125471966244),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/098.txt', 0.95424250943932487),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/100.txt', 1.4313637641589874),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/047.txt', 1.2041199826559248),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/051.txt', 1.2041199826559248),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/054.txt', 2.4082399653118496),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/056.txt', 1.2041199826559248),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/075.txt', 2.4082399653118496),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/100.txt', 9.6329598612473983)]

Step 24

```
#Aggregate the TFIDF scores for all words in each document
sumtfidf=filterresultstfidf1.reduceByKey(lambda x, y: x+y)
sumtfidf.collect()
sumtfidf.count()
```

Step 24 – Output

[(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/072.txt',
0.47712125471966244), (u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/087.txt',
0.95424250943932487), (u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/096.txt',
0.47712125471966244), (u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/071.txt',
6.6796975660752738), (u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/039.txt',
0.95424250943932487), (u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/008.txt',
0.47712125471966244), (u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/079.txt',
0.47712125471966244), (u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/033.txt',
0.47712125471966244), (u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/019.txt',
0.47712125471966244), (u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/066.txt',
0.47712125471966244), (u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/090.txt',
0.47712125471966244), (u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/100.txt',
11.064323625406386), (u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/061.txt',
0.95424250943932487), (u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/089.txt',
0.47712125471966244), (u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/018.txt',
0.47712125471966244), (u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/012.txt',
0.47712125471966244), (u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/052.txt',
0.95424250943932487), (u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/001.txt',
1.4313637641589874), (u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/063.txt',
0.95424250943932487), (u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/051.txt',
1.2041199826559248), (u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/015.txt',
0.95424250943932487), (u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/053.txt',
0.47712125471966244), (u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/026.txt',
0.47712125471966244), (u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/022.txt',
0.47712125471966244), (u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/098.txt',
0.95424250943932487), (u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/054.txt',
2.8853612200315122), (u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/056.txt',
1.2041199826559248), (u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/007.txt',

0.47712125471966244), (u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/047.txt',
1.2041199826559248), (u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/068.txt',
1.4313637641589874), (u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/020.txt',
0.47712125471966244), (u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/060.txt',
1.4313637641589874), (u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/075.txt',
2.4082399653118496), (u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/073.txt',
0.95424250943932487), (u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/097.txt',
0.47712125471966244)]
35

Step 25

```
 #The join gave (word, (doc,TF,IDF,TFIDF),topN) which we now reduce to (doc, (word,1))
#We do this to find out how many words from the query appeared in a given doc
filterresultstfidf2 = filterresults.map(lambda(a, ((b, c, d, e), f)):(b,(a,1)))
filterresultstfidf2.collect()
```

Step 25 – Output

[(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/001.txt', (u'game', 1)),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/007.txt', (u'game', 1)),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/008.txt', (u'game', 1)),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/012.txt', (u'game', 1)),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/015.txt', (u'game', 1)),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/018.txt', (u'game', 1)),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/019.txt', (u'game', 1)),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/020.txt', (u'game', 1)),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/022.txt', (u'game', 1)),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/026.txt', (u'game', 1)),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/033.txt', (u'game', 1)),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/039.txt', (u'game', 1)),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/052.txt', (u'game', 1)),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/053.txt', (u'game', 1)),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/054.txt', (u'game', 1)),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/060.txt', (u'game', 1)),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/061.txt', (u'game', 1)),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/063.txt', (u'game', 1)),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/066.txt', (u'game', 1)),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/068.txt', (u'game', 1)),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/071.txt', (u'game', 1)),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/072.txt', (u'game', 1)),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/073.txt', (u'game', 1)),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/079.txt', (u'game', 1)),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/087.txt', (u'game', 1)),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/089.txt', (u'game', 1)),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/090.txt', (u'game', 1)),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/096.txt', (u'game', 1)),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/097.txt', (u'game', 1)),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/098.txt', (u'game', 1)),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/100.txt', (u'game', 1)),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/047.txt', (u'Sharapova', 1)),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/051.txt', (u'Sharapova', 1)),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/054.txt', (u'Sharapova', 1)),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/056.txt', (u'Sharapova', 1)),

(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/075.txt', (u'Sharapova', 1)),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/100.txt', (u'Sharapova', 1))]

Step 26

```
#Take the RDD from Step 3 and remove the word to calculate the total by key (doc name)
wordcountperdocperword=filterresultstfidf2.map(lambda (b,(a,n)):(b,n))
wordcountperdocperword.collect() # to verify
```

Step 26 – Output

[(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/001.txt', 1),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/007.txt', 1),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/008.txt', 1),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/012.txt', 1),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/015.txt', 1),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/018.txt', 1),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/019.txt', 1),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/020.txt', 1),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/022.txt', 1),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/026.txt', 1),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/033.txt', 1),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/039.txt', 1),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/052.txt', 1),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/053.txt', 1),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/054.txt', 1),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/060.txt', 1),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/061.txt', 1),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/063.txt', 1),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/066.txt', 1),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/068.txt', 1),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/071.txt', 1),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/072.txt', 1),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/073.txt', 1),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/079.txt', 1),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/087.txt', 1),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/089.txt', 1),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/090.txt', 1),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/096.txt', 1),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/097.txt', 1),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/098.txt', 1),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/100.txt', 1),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/047.txt', 1),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/051.txt', 1),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/054.txt', 1),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/056.txt', 1),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/075.txt', 1),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/100.txt', 1)]

Step 27

```
#Aggregate the RDD from Step 4 to calculate the number of words found in each doc
wordcountperdoc=wordcountperdocperword.reduceByKey(lambda x,y: x+y)
wordcountperdoc.collect() # to verify
wordcountperdoc.count()
```

Step 27 – Output

```
[(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/072.txt', 1),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/087.txt', 1),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/096.txt', 1),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/071.txt', 1),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/039.txt', 1),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/008.txt', 1),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/079.txt', 1),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/033.txt', 1),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/019.txt', 1),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/066.txt', 1),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/090.txt', 1),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/100.txt', 2),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/061.txt', 1),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/089.txt', 1),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/018.txt', 1),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/012.txt', 1),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/052.txt', 1),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/001.txt', 1),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/063.txt', 1),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/051.txt', 1),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/015.txt', 1),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/053.txt', 1),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/026.txt', 1),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/022.txt', 1),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/098.txt', 1),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/054.txt', 2),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/056.txt', 1),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/007.txt', 1),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/047.txt', 1),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/068.txt', 1),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/020.txt', 1),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/060.txt', 1),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/075.txt', 1),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/073.txt', 1),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/097.txt', 1)]
35
```

Step 28

```
#Now join tables from Step 2 and Step 5
scorejoin=sumtfidf.join(wordcountperdoc)
scorejoin.collect()
```

Step 28 – Output

```
[(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/072.txt',
(0.47712125471966244, 1)),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/087.txt', (0.95424250943932487,
1)), (u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/096.txt',
(0.47712125471966244, 1)),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/008.txt', (0.47712125471966244,
1)), (u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/039.txt',
(0.95424250943932487, 1)),
```

(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/079.txt', (0.47712125471966244, 1)), (u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/033.txt', (0.47712125471966244, 1)),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/019.txt', (0.47712125471966244, 1)), (u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/066.txt', (0.47712125471966244, 1)),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/071.txt', (6.6796975660752738, 1)), (u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/090.txt', (0.47712125471966244, 1)),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/100.txt', (11.064323625406386, 2)),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/061.txt', (0.95424250943932487, 1)), (u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/012.txt', (0.47712125471966244, 1)),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/018.txt', (0.47712125471966244, 1)), (u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/089.txt', (0.47712125471966244, 1)),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/052.txt', (0.95424250943932487, 1)), (u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/001.txt', (1.4313637641589874, 1)),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/063.txt', (0.95424250943932487, 1)), (u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/015.txt', (0.95424250943932487, 1)),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/051.txt', (1.2041199826559248, 1)), (u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/053.txt', (0.47712125471966244, 1)),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/026.txt', (0.47712125471966244, 1)), (u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/022.txt', (0.47712125471966244, 1)),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/098.txt', (0.95424250943932487, 1)), (u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/054.txt', (2.8853612200315122, 2)),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/056.txt', (1.2041199826559248, 1)), (u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/007.txt', (0.47712125471966244, 1)),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/047.txt', (1.2041199826559248, 1)), (u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/068.txt', (1.4313637641589874, 1)),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/075.txt', (2.4082399653118496, 1)), (u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/020.txt', (0.47712125471966244, 1)),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/060.txt', (1.4313637641589874, 1)), (u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/097.txt', (0.47712125471966244, 1)),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/073.txt', (0.95424250943932487, 1))]

Step 29

```
#The joined table contains (doc,(total TFIDF score for query words, number of query words found in doc))
#Apply the score formula that is (sum of tfidf for all words)*(number of words found in doc/total number of words queried)
#The total number of words queried was stored in nofquerywords earlier
score1 = scorejoin.map(lambda (a, (b, c)): (a, b*c/nofquerywords))
```

score1.collect() # to verify

Step 29 – Output

[(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/072.txt',
0.23856062735983122), (u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/087.txt',
0.47712125471966244), (u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/096.txt',
0.23856062735983122), (u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/008.txt',
0.23856062735983122), (u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/039.txt',
0.47712125471966244), (u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/079.txt',
0.23856062735983122), (u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/033.txt',
0.23856062735983122), (u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/019.txt',
0.23856062735983122), (u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/066.txt',
0.23856062735983122), (u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/071.txt',
3.3398487830376369), (u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/090.txt',
0.23856062735983122),
**(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/100.txt',
11.064323625406386)**,
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/061.txt', 0.47712125471966244),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/012.txt', 0.23856062735983122),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/018.txt', 0.23856062735983122),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/089.txt', 0.23856062735983122),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/052.txt', 0.47712125471966244),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/001.txt', 0.71568188207949368),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/063.txt', 0.47712125471966244),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/015.txt', 0.47712125471966244),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/051.txt', 0.6020599913279624),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/053.txt', 0.23856062735983122),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/026.txt', 0.23856062735983122),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/022.txt', 0.23856062735983122),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/098.txt', 0.47712125471966244),
**(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/054.txt',
2.8853612200315122)**,
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/056.txt', 0.6020599913279624),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/007.txt', 0.23856062735983122),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/047.txt', 0.6020599913279624),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/068.txt', 0.71568188207949368),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/075.txt', 1.2041199826559248),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/020.txt', 0.23856062735983122),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/060.txt', 0.71568188207949368),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/097.txt', 0.23856062735983122),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/073.txt', 0.47712125471966244)]

#FINAL TABLE WITH DOCUMENT NAMES AND SCORES

Step 30

#Sort the final table with (doc name, score) in descending order
finalanswer = score1.sortBy((lambda x: x[1]),ascending=False)
finalanswer.collect()

Step 30 – Output

```
[(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/100.txt', 11.064323625406386),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/071.txt', 3.3398487830376369),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/054.txt', 2.8853612200315122),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/075.txt', 1.2041199826559248),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/001.txt', 0.71568188207949368),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/068.txt', 0.71568188207949368),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/060.txt', 0.71568188207949368),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/051.txt', 0.6020599913279624),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/056.txt', 0.6020599913279624),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/047.txt', 0.6020599913279624),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/087.txt', 0.47712125471966244),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/039.txt', 0.47712125471966244),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/061.txt', 0.47712125471966244),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/052.txt', 0.47712125471966244),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/063.txt', 0.47712125471966244),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/015.txt', 0.47712125471966244),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/098.txt', 0.47712125471966244),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/073.txt', 0.47712125471966244),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/072.txt', 0.23856062735983122),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/096.txt', 0.23856062735983122),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/008.txt', 0.23856062735983122),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/079.txt', 0.23856062735983122),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/033.txt', 0.23856062735983122),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/019.txt', 0.23856062735983122),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/066.txt', 0.23856062735983122),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/090.txt', 0.23856062735983122),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/012.txt', 0.23856062735983122),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/018.txt', 0.23856062735983122),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/089.txt', 0.23856062735983122),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/053.txt', 0.23856062735983122),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/026.txt', 0.23856062735983122),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/022.txt', 0.23856062735983122),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/007.txt', 0.23856062735983122),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/020.txt', 0.23856062735983122),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/097.txt', 0.23856062735983122)]
```

#EXTRACT TOP N DOCUMENTS

FINAL RESULT

```
#top N was stored earlier
finalanswer.take(topN)
```

FINAL RESULT – Output

```
[(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/100.txt', 11.064323625406386),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/071.txt', 3.3398487830376369),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/054.txt', 2.8853612200315122)]
```

8.SUMMARIZED CODE

Below is the main code without the outputs or extra verification lines (as used in previous section to explain the code and logic). Comments are still included. Anyone wanting to use this code needs to just replace the lines in red font/yellow highlight with their inputs and it would work like magic!

#TERM FREQUENCY

```
#Create RDDS with file path as key and content as values
files = sc.wholeTextFiles("/user/root/finalexam/tennis")
#Count the number of documents in corpus and store in variable ndoc for later use
ndoc=files.count()
#Split the words in file content
words = files.flatMapValues(lambda line: line.split( ))
#Reverse the key,value pair such the the word becomes the key and doc name becomes the value
reverse = words.map(lambda fields: (fields[1],fields[0]))
#Sort by key - not necessary, but helps in verification of code and analysis
sorted = reverse.sortByKey()
#Add 1 to each line/word for word count in each document
count = sorted.map(lambda word: (word,1))
#Add the count to get term frequency
#At this point, the word and doc name together form the key
sum = count.reduceByKey(lambda x,y: x+y)
a=sum.sortByKey()
#Change the layout to make the word key and the rest a tuple of values
aa=a.map(lambda ((x_1, x_2), y): (x_1, (x_2, y)))
```

#INVERSE DOCUMENT FREQUENCY

```
#Here we take the previous table that contained word, doc name, and term frequency
#From that table, we keep just the key that consists of word and doc name and get rid of term
frequency
idf = a.keys()
#We then just keep the words and get rid of the document name as well. This would tell us the number
of document a particular word appeared in without the document names so that we can do a sum
idfkeys=idf.map(lambda line: line[0])
#Now add 1 to each word for a word count
idfcount = idfkeys.map(lambda word: (word,1))
#Reduce by key(word) to get a word count which the document frequency
#The value is the number of documents in which the word appeared in
idfsum = idfcount.reduceByKey(lambda x,y:x+y)
b=idfsum.sortByKey()
#Now calculate the inverse of document frequency
#Here we use the ndoc that we stored earlier
idfinversed = idfsum.mapValues(lambda y: ndoc/y).sortByKey()
#And apply LOG 10 function
import math
idflog = idfinversed.mapValues(lambda y: math.log10(y))
```

#TERM FREQUENCY - INVERSE DOCUMENT FREQUENCY

```
#Join the TF and IDF tables by word to get tfidf table
tfidfjoin = aa.leftOuterJoin(idflog)
#The table would be in this format (word,(doc name, TF), IDF) which is converted to (word,
(doc,TF,IDF,TFIDF)) format
tfidffinal = tfidfjoin.map(lambda(w, ((x, y), z)):(w,(x,y,z,z*y)))
```

#TAKE INPUTS FOR QUERY

```
#Here is the INPUT for query words
query = sc.parallelize(["Sharapova game"])
#topN is the INPUT for how many documents need to be retrieved
topN = 3
#Convert the query words into an RDD of words that can be processed further
#Split the words
querywords = query.flatMap(lambda line: line.split())
#Convert into key,value pair
querywordspair = querywords.map(lambda word: (word,topN))
#Store the total number of words queried for calculation later
nofquerywords = querywordspair.count()
#Now filter out the words queried from the TFIDF table to calculate their scores
filterresults = tfidffinal.join(querywordspair)
```

#CALCULATE SCORE

```
#Create an RDD from the TFIDF table with only the document name and TFIDF score per word
#The join gave (word, (doc,TF,IDF,TFIDF),topN) which we now reduce to (doc,TFIDF)
filterresultstfidf1 = filterresults.map(lambda(a, ((b, c, d, e), f)):(b,e))
#Aggregate the TFIDF scores for all words in each document
sumtfidf=filterresultstfidf1.reduceByKey(lambda x, y: x+y)
#Step 3 to calculate the score
#The join gave (word, (doc,TF,IDF,TFIDF),topN) which we now reduce to (doc, (word,1))
#We do this to find out how many words from the query appeared in a given doc
filterresultstfidf2 = filterresults.map(lambda(a, ((b, c, d, e), f)):(b,(a,1)))
#Take the RDD from Step 3 and remove the word to calculate the total by key (doc name)
wordcountperdocperword=filterresultstfidf2.map(lambda (b,(a,n)):(b,n))
#Aggregate the RDD from Step 4 to calculate the number of words found in each doc
wordcountperdoc=wordcountperdocperword.reduceByKey(lambda x,y: x+y)
#Now join tables from Step 2 and Step 5
scorejoin=sumtfidf.join(wordcountperdoc)
#The joined table contains (doc,(total TFIDF score for query words, number of query words found in
doc))
#Apply the score formula that is (sum of tfidf for all words)*(number of words found in doc/total
number of words queried)
#The total number of words queried was stored in nofquerywords earlier
score1 = scorejoin.map(lambda (a, (b, c)): (a, b*c/nofquerywords))
```

#FINAL TABLE WITH DOCUMENT NAMES AND SCORES

#Sort the final table with (doc name, score) in descending order
finalanswer = score1.sortBy((lambda x: x[1]),ascending=False)

#EXTRACT TOP N DOCUMENTS

#top N was stored earlier
finalanswer.take(topN)

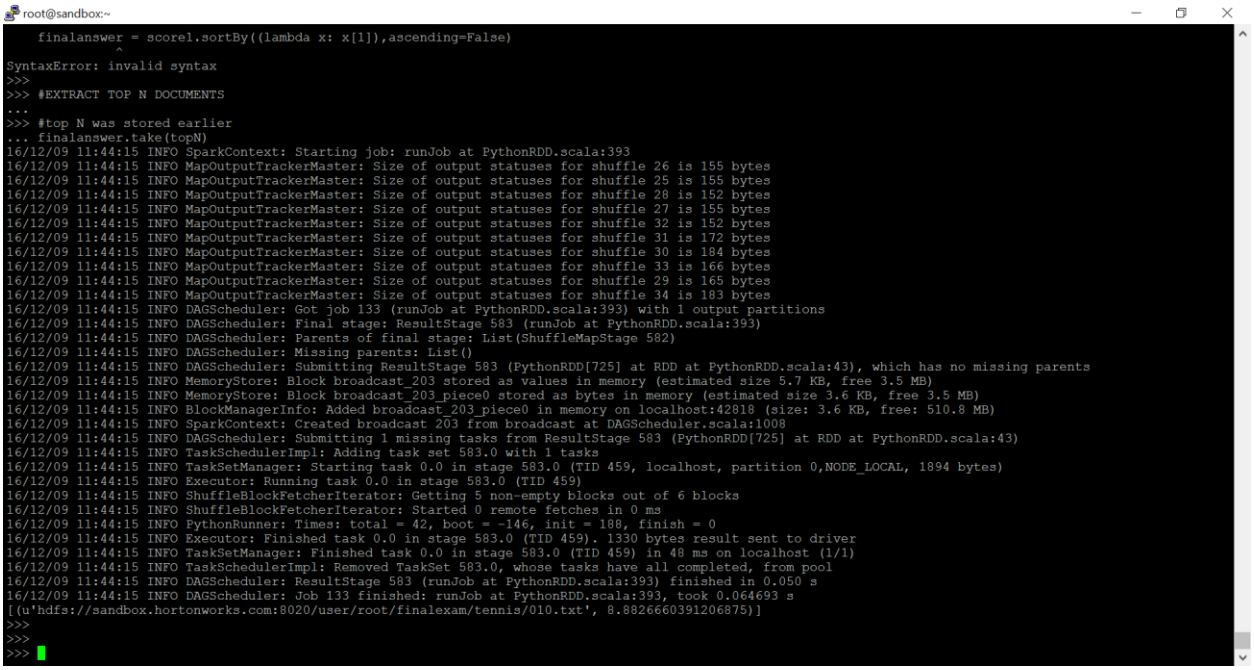
9.TEST & RESULTS

QUERY WORDS

"**Martinez Hantuchova Myskina Lindsay**"

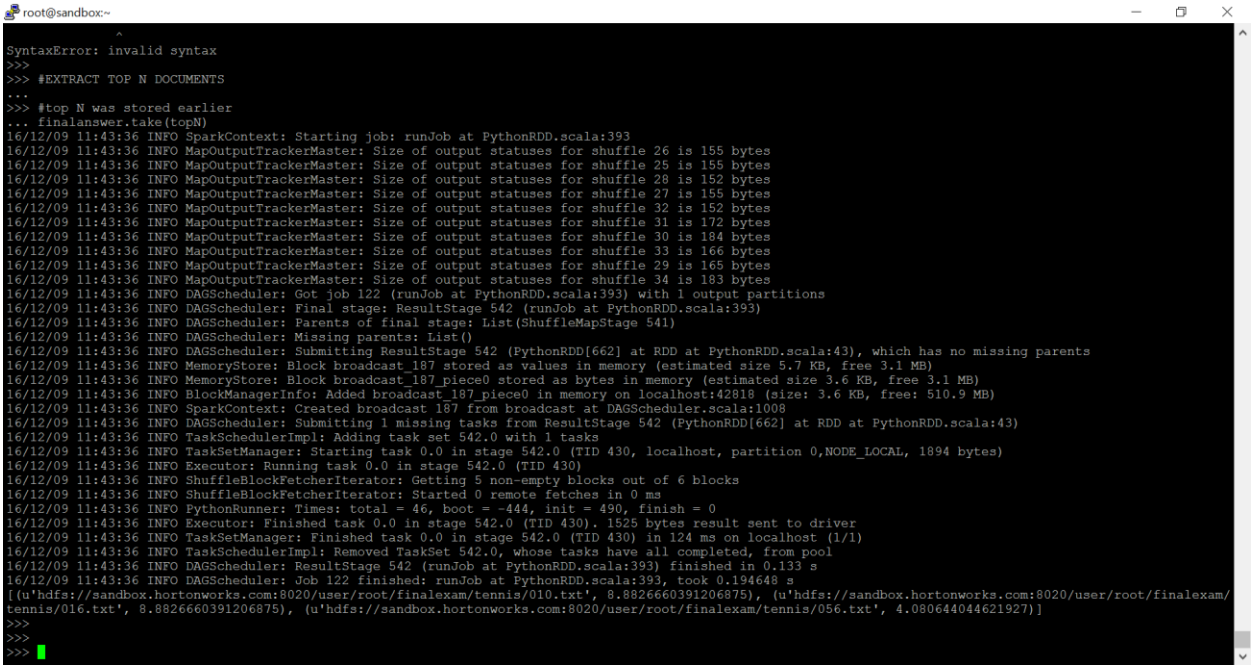
RESULTS FOR N = 1

[(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/010.txt', 8.8826660391206875)]



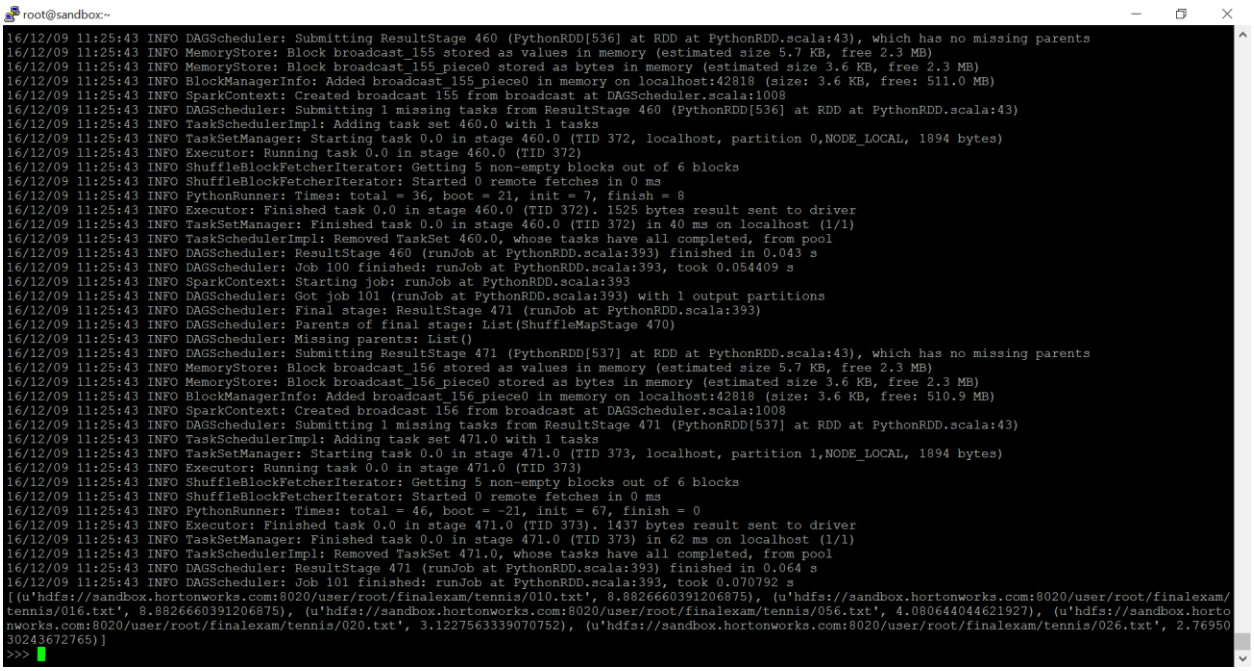
RESULTS FOR N = 3

[(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/010.txt', 8.8826660391206875), (u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/016.txt', 8.8826660391206875), (u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/056.txt', 4.080644044621927)]



RESULTS FOR N = 5

[(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/010.txt', 8.8826660391206875),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/016.txt', 8.8826660391206875),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/056.txt', 4.080644044621927),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/020.txt', 3.1227563339070752),
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/026.txt', 2.7695030243672765)]



10. CONTENT OF TOP 5 DOCUMENTS

The content of top 5 documents retrieved in query result are shown here.

"**Martinez Hantuchova Myskina Lindsay**"

The query words (given above) are highlighted in the text with same colours as shown above.

Document 010

Hantuchova in Dubai last eight

Daniela **Hantuchova** moved into the quarter-finals of the Dubai Open, after beating Elene Likhotseva of Russia 7-5 6-4, and now faces Serena Williams.

Australian Open champion Williams survived an early scare to beat Russia's Elena Bovina 1-6 6-1 6-4. World number one **Lindsay** Davenport and Anastasia **Myskina** also progressed. Davenport defeated China's Jie Zheng 6-2 7-5, while French Open champion **Myskina** sailed through after her opponent Marion Bartoli retired hurt. American Davenport will now face fellow former Wimbledon champion, Conchita **Martinez** of Spain, who ousted seventh-seeded Nathalie Dechy of France 6-1 6-2. **Myskina** will face eighth-seed Patty Schnyder from Switzerland, who defeated China's Li Na 6-3 7-6 (10-8). The other quarter final pits wild card Sania Mirza of India against Jelena Jankovic of Serbia and Montenegro, who both won on Tuesday.

Before her meeting with **Martinez**, Davenport believes there is some room for improvement in her game. "I started well and finished well, but played some so-so games in the middle," she said. Williams was also far from content. "I don't know what I was doing there," she said. "It was really windy and I hadn't played in the wind. All my shots were going out of here." But **Hantuchova** is in upbeat mood ahead of her clash with the younger Williams sister, who was handed a first-round bye. "I feel I have an advantage (over Serena) because I have already played two matches on these courts," she said. "It is a difficult court to play on. Very fast and sometimes you feel you have no control over the ball."

Document 016

Hantuchova in Dubai last eight

Daniela **Hantuchova** moved into the quarter-finals of the Dubai Open, after beating Elene Likhotseva of Russia 7-5 6-4, and now faces Serena Williams.

Australian Open champion Williams survived an early scare to beat Russia's Elena Bovina 1-6 6-1 6-4. World number one **Lindsay** Davenport and Anastasia **Myskina** also progressed. Davenport defeated China's Jie Zheng 6-2 7-5, while French Open champion **Myskina** sailed through after her opponent Marion Bartoli retired hurt. American Davenport will now face fellow former Wimbledon champion, Conchita **Martinez** of Spain, who ousted seventh-seeded Nathalie Dechy of France 6-1 6-2. **Myskina** will face eighth-seed Patty Schnyder from Switzerland, who defeated China's Li Na 6-3 7-6 (10-8). The other quarter final pits wild card Sania Mirza of India against Jelena Jankovic of Serbia and Montenegro, who both won on Tuesday.

Before her meeting with **Martinez**, Davenport believes there is some room for improvement in her game. "I started well and finished well, but played some so-so games in the middle," she said. Williams

was also far from content. "I don't know what I was doing there," she said. "It was really windy and I hadn't played in the wind. All my shots were going out of here." But **Hantuchova** is in upbeat mood ahead of her clash with the younger Williams sister, who was handed a first-round bye. "I feel I have an advantage (over Serena) because I have already played two matches on these courts," she said. "It is a difficult court to play on. Very fast and sometimes you feel you have no control over the ball."

Document 056

Henman to face Saulnier test

British number one Tim Henman will face France's Cyril Saulnier in the first round of next week's Australian Open.

Greg Rusedski, the British number two, is in the same quarter of the draw and could face Andy Roddick in the second round if he beats Swede Jonas Bjorkman. Local favourite Lleyton Hewitt will meet France's Arnaud Clement, while defending champion and world number one Roger Federer faces Fabrice Santoro. Women's top seed **Lindsay** Davenport drew Spanish veteran Conchita **Martinez**.

Henman came from two sets down to defeat Saulnier in the first round of the French Open last year, so he knows he faces a tough test in Melbourne. The seventh seed, who has never gone beyond the quarter-finals in the year's first major and is lined up to meet Roddick in the last eight, is looking forward to the match. "He's tough player on any surface, he's got a lot of ability," he said. "We had a really tight one in Paris that went my way so I'm going to need to play well from the outset because he's a dangerous competitor." Switzerland's Federer, seeded one, is the hot favourite having won three of the four grand slam titles in 2004. He has beaten Santoro in five of their seven previous encounters, but is taking nothing for granted. "It's a tricky match," Federer said. "I played him at the US Open and won quite comfortably then. But you never know, if the rhythm is a bit off, he can keep you guessing and make it difficult. "The most important thing, though, is to get used to playing five-set matches and winning them." The 23-year-old could meet four-time champion Andre Agassi in the quarter-finals before meeting Russian Marat Safin, the player he beat in last year's final.

Eighth-seeded American Agassi is set to play a qualifier in round one if he can shake off a hip injury which ruled him out of the Kooyong Classic. Second seed Andy Roddick will open his campaign against Irakli Labadze of Georgia. The American could meet Rusedski in the second round, seventh seed Henman in the quarter-finals and Hewitt in the last four. Hewitt is hoping to become the first Australian man to win the event since Mark Edmondson in 1976. The 23-year-old has never been beyond round four in eight attempts at Melbourne Park but has at least secured the opposite half of the draw to Federer, who beat him in the Australian Open, Wimbledon and US Open last year. Safin, seeded four, opens his campaign against a qualifier with 16th seed Tommy Haas, the player he beat in the semi-finals in 2002, a possible fourth-round opponent.

In the women's draw, Davenport could encounter eighth-seeded Venus Williams in the quarter-finals and third-ranked Anastasia **Myskina**, the French Open champion, in the semi-finals. Bronchitis ruled Davenport, the 2000 Australian Open champion, out of her Sydney quarter-final on Thursday. Venus Williams, who lost to younger sister Serena in the Melbourne final two years ago, opens against Eleni Daniilidou of Greece. Serena Williams, who won her fourth consecutive grand slam at the 2003 Australian Open, was drawn in the bottom quarter with second seed Amelie Mauresmo, a runner-up in 1999. Serena will open against another Frenchwoman Camille Pin, while Mauresmo plays Australia's Samantha Stosur. Wimbledon champion Maria Sharapova, seeded fourth, drew a qualifier in the first round but could meet fellow Russian Svetlana Kuznetsova, the US Open winner, in the last eight

1 Roger Federer (Switzerland)

2 Andy Roddick (US)

3 Lleyton Hewitt (Australia)

4 Marat Safin (Russia)

5 Carlos Moya (Spain)

6 Guillermo Coria (Argentina)

7 Tim Henman (Britain)

8 Andre Agassi (US)

9 David Nalbandian (Argentina)

10 Gaston Gaudio (Argentina)

11 Joachim Johansson (Sweden)

12 Guillermo Canas (Argentina)

13 Tommy Robredo (Spain)

14 Sebastien Grosjean (France)

15 Mikhail Youzhny (Russia)

16 Tommy Haas (Germany)

17 Andrei Pavel (Romania)

18 Nicolas Massu (Chile)

19 Vincent Spadea (US)

20 Dominik Hrbaty (Slovakia)

21 Nicolas Kiefer (Germany)

22 Ivan Ljubicic (Croatia)

23 Fernando Gonzalez (Chile)

24 Feliciano Lopez (Spain)

25 Juan Ignacio Chela (Argentina)

26 Nikolay Davydenko (Russia)

27 Paradorn Srichaphan (Thailand)

28 Mario Ancic (Croatia)

29 Taylor Dent (US)

30 Thomas Johansson (Sweden)

31 Juan Carlos Ferrero (Spain)

32 Jurgen Melzer (Austria)

1 **Lindsay** Davenport (US)

2 Amelie Mauresmo (France)

3 Anastasia **Myskina** (Russia)

4 Maria Sharapova (Russia)

5 Svetlana Kuznetsova (Russia)

6 Elena Dementieva (Russia)

7 Serena Williams (US)

8 Venus Williams (US)

9 Vera Zvonareva (Russia)

10 Alicia Molik (Australia)

11 Nadia Petrova (Russia)

12 Patty Schnyder (Switzerland)

13 Karolina Sprem (Croatia)

14 Francesca Schiavone (Italy)

15 Silvia Farina Elia (Italy)

16 Ai Sugiyama (Japan)

17 Fabiola Zuluaga (Colombia)

18 Elena Likhovtseva (Russia)

19 Nathalie Dechy (France)

20 Tatiana Golovin (France)

21 Amy Frazier (US)

22 Magdalena Maleeva (Bulgaria)

23 Jelena Jankovic (Serbia and Montenegro)

24 Mary Pierce (France)

25 Lisa Raymond (US)

26 Daniela **Hantuchova** (Slovakia)

27 Anna Smashnova (Israel)

28 Shinobu Asagoe (Japan)

29 Gisela Dulko (Argentina)

30 Flavia Pennetta (Italy)

31 Jelena Kostanic (Croatia)

32 Iveta Benesova (Czech Republic)

Document020

Martinez sees off Vinci challenge

Veteran Spaniard Conchita **Martinez** came from a set down to beat Italian Roberta Vinci at the Qatar Open in Doha.

The 1994 Wimbledon champion won 5-7 6-0 6-2 to earn a second round meeting with French Open champion Anastasia **Myskina**. Fifth seed Patty Schnyder also had a battle as she needed three sets to beat China's Na Li 7-5 3-6 7-5. Slovakian Daniela **Hantuchova** beat Bulgarian Magdaleena Maleeva 4-6 6-4 6-3 to set up a second round clash with Russian Elena Bovina. The veteran **Martinez** found herself in trouble early on against Vinci with the Italian clinching the set thanks to breaks in the third and 11th games. But Vinci's game fell to pieces after that and **Martinez** swept her aside with some crisp cross-court returns and deft volleys. In the day's other matches, Japan's Ai Sugiyama defeated Australian Samantha Stosur 6-2 6-3 while Australian Nicole Pratt beat Tunisian Selima Sfar 7-5 6-2 and will next face compatriot Alicia Molik.

Document026

Davenport dismantles young rival

Top seed **Lindsay** Davenport booked her place in the last 16 of the Australian Open with a convincing 6-2 6-4 win over Nicole Vaidisova of the Czech Republic.

The American had too much power for her 15-year-old opponent, breaking twice in the first set and once in the second. The German-born Vaidisova rallied well at times but was unable to find a way back after falling behind 3-2 in the opening set. Davenport, who closed out with an ace, plays Karolina Sprem in the next round. "I was fully expecting a tough opponent and was able to play well enough to get through it," said Davenport. "I think she hits some great shots. She made some errors but probably some inexperience played a role in that. But she's so young and obviously has a big game and has many, many years to improve on that." Sprem, the Croatian 13th seed, saw off Russia's Elena Likhovtseva 6-3 6-2. Former world number one

powered her way into the fourth round with a straight sets win over Anna Smashnova. The 27th seed from Israel stuck with Williams until 3-3 in the first set before it became one-way traffic.

The American made 26 unforced errors but was still good enough to romp through the contest in exactly an hour. She reeled off nine straight games to finish a 6-3 6-0 winner.

remains on course to become the first Australian to win her home title since Chris O'Neil in 1978. The 10th seed equalled her best performance at a Grand Slam event when she beat unseeded Russian Nadia Petrova 6-3 6-2 to reach the fourth round. After a tough first set, Molik grew in confidence and won in just 56 minutes. She will now meet Venus Williams. "Bring it on," said the 23-year-old. "I played pretty well and it was nice to get through in straight sets." "We were destined to meet, I guess," Williams said referring to her match with Molik. "It will be a huge match for her in Australia. I can tell she's probably very motivated by that so I need to get out there and play well."

beat Slovakia's Daniela **Hantuchova** in a rollercoaster match. Dementieva came through 7-5 5-7 6-4, becoming the seventh Russian woman to reach the last 16 in Melbourne. The match lasted almost three hours and featured 13 service breaks, including three in the final set when Dementieva held her nerve to seal the win. She now faces

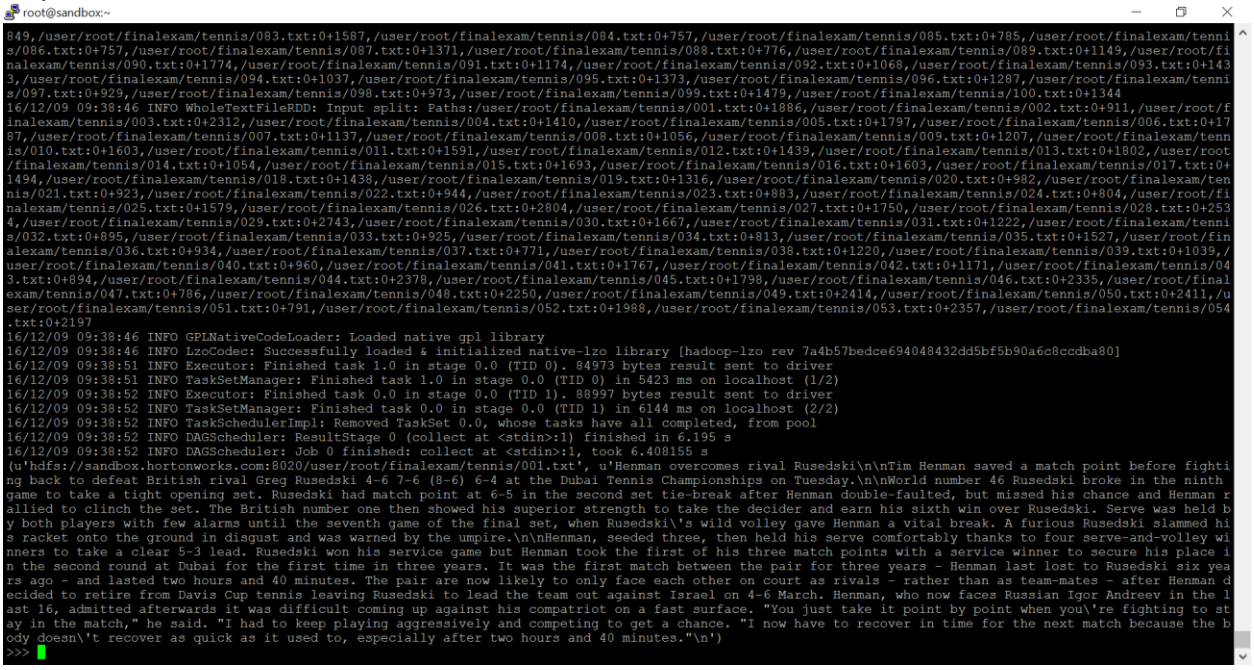
after the Swiss 12th seed beat American Abigail Spears 7-6 6-3. French Open champion

received a free ride into the last 16 after Lisa Raymond was forced to withdraw. Raymond, the 25th seeded American, was ruled out after sustaining a left abdominal muscle tear in the doubles. **Myskina**, the third seed, now plays France's

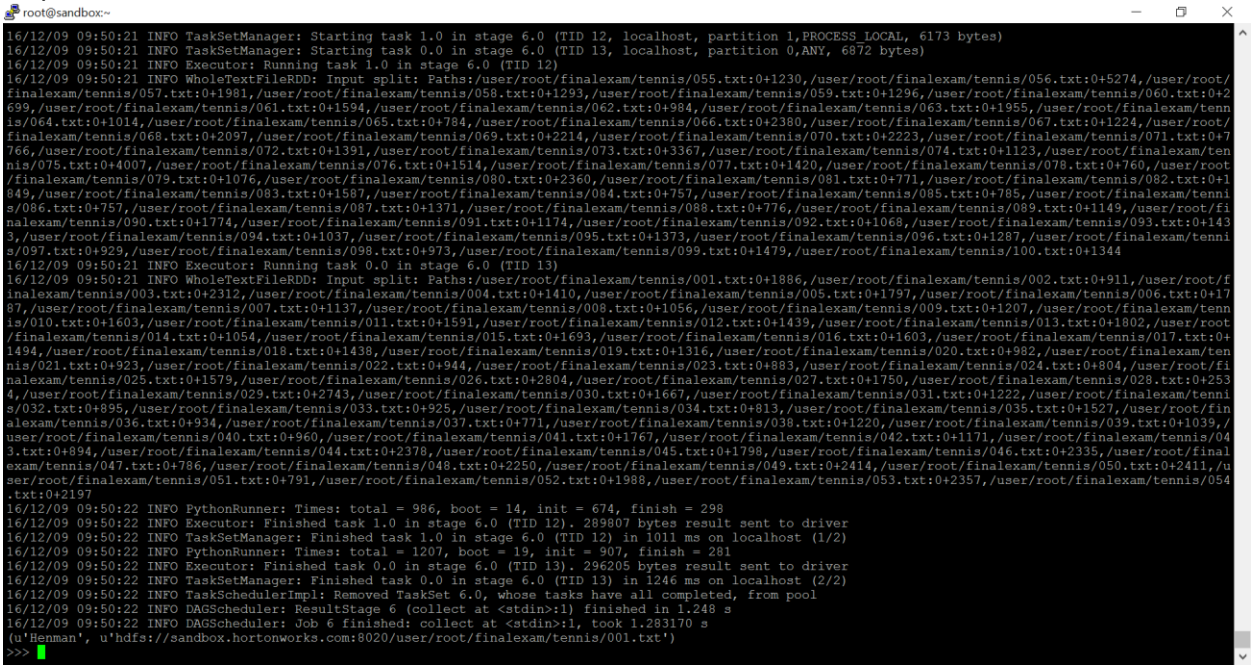
who beat Francesca Schiavone of Italy 6-3 6-3. "I'm extremely disappointed because I couldn't have asked to play better in my first two matches," Raymond said.

11. SCREENSHOTS (for Code Explanation)

Step1



Step5



Step7

root@sandbox:~

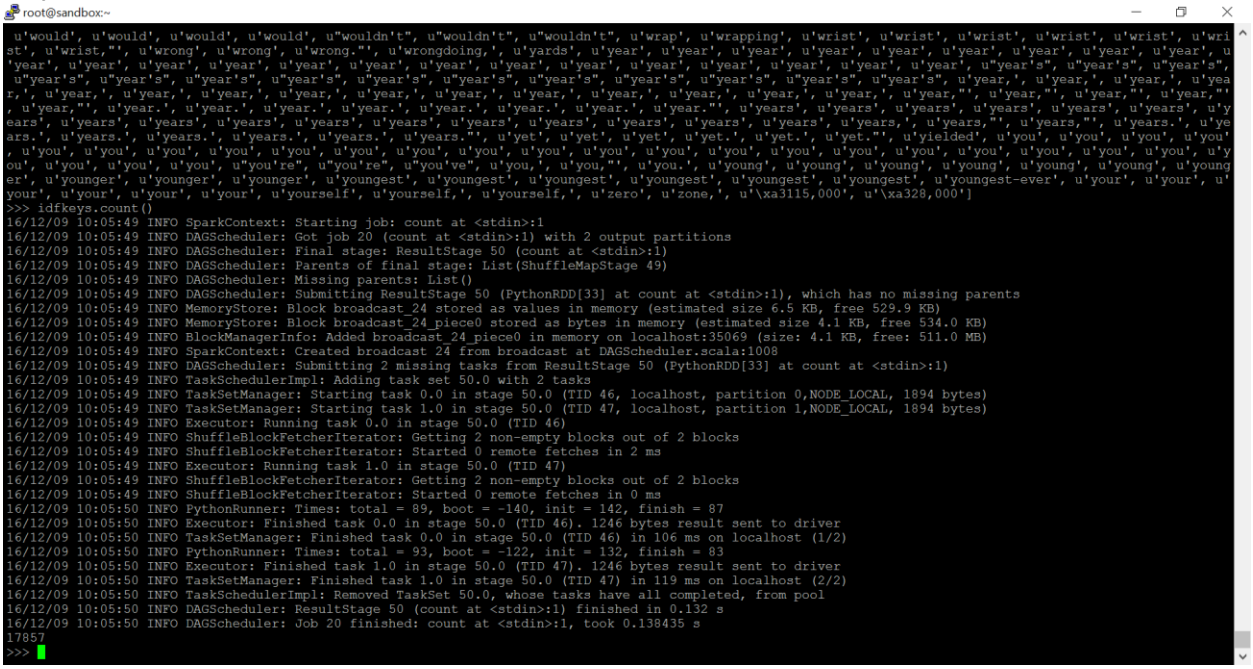
```
16/12/09 10:00:51 INFO PythonRunner: Times: total = 169, boot = -293, init = 399, finish = 63
16/12/09 10:00:51 INFO Executor: Finished task 0.0 in stage 10.0 (TID 20). 367015 bytes result sent to driver
16/12/09 10:00:51 INFO TaskSetManager: Finished task 0.0 in stage 10.0 (TID 20) in 248 ms on localhost (1/2)
16/12/09 10:00:51 INFO PythonRunner: Times: total = 220, boot = -499, init = 601, finish = 118
16/12/09 10:00:51 INFO Executor: Finished task 1.0 in stage 10.0 (TID 21). 309663 bytes result sent to driver
16/12/09 10:00:51 INFO TaskSetManager: Finished task 1.0 in stage 10.0 (TID 21) in 285 ms on localhost (2/2)
16/12/09 10:00:51 INFO TaskSchedulerImpl: Removed TaskSet 10.0, whose tasks have all completed, from pool
16/12/09 10:00:51 INFO DAGScheduler: ResultStage 10 (collect at <stdin>:1) finished in 0.298 s
16/12/09 10:00:51 INFO DAGScheduler: Job 9 finished: collect at <stdin>:1, took 1.760967 s
(u'"', u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/080.txt')
>>> count = sorted.map(lambda word: (word,1))
>>> count.collect()[0]
16/12/09 10:01:59 INFO SparkContext: Starting job: collect at <stdin>:1
16/12/09 10:01:59 INFO MapOutputTrackerMaster: Size of output statuses for shuffle 0 is 155 bytes
16/12/09 10:01:59 INFO DAGScheduler: Got job 10 (collect at <stdin>:1) with 2 output partitions
16/12/09 10:01:59 INFO DAGScheduler: Final stage: ResultStage 12 (collect at <stdin>:1)
16/12/09 10:01:59 INFO DAGScheduler: Parents of final stage: List(ShuffleMapStage 11)
16/12/09 10:01:59 INFO DAGScheduler: Missing parents: List()
16/12/09 10:01:59 INFO DAGScheduler: Submitting ResultStage 12 (PythonRDD[15] at collect at <stdin>:1), which has no missing parents
16/12/09 10:01:59 INFO MemoryStore: Block broadcast_12 stored as values in memory (estimated size 5.5 KB, free 411.2 KB)
16/12/09 10:01:59 INFO MemoryStore: Block broadcast_12_piece0 stored as bytes in memory (estimated size 3.5 KB, free 414.6 KB)
16/12/09 10:01:59 INFO BlockManagerInfo: Added broadcast_12_piece0 in memory on localhost:35069 (size: 3.5 KB, free: 511.1 MB)
16/12/09 10:01:59 INFO SparkContext: Created broadcast 12 from broadcast at DAGScheduler.scala:1008
16/12/09 10:01:59 INFO DAGScheduler: Submitting 2 missing tasks from ResultStage 12 (PythonRDD[15] at collect at <stdin>:1)
16/12/09 10:01:59 INFO TaskSchedulerImpl: Adding task set 12.0 with 2 tasks
16/12/09 10:01:59 INFO TaskSetManager: Starting task 0.0 in stage 12.0 (TID 22, localhost, partition 0,NODE_LOCAL, 1894 bytes)
16/12/09 10:01:59 INFO TaskSetManager: Starting task 1.0 in stage 12.0 (TID 23, localhost, partition 1,NODE_LOCAL, 1894 bytes)
16/12/09 10:01:59 INFO Executor: Running task 0.0 in stage 12.0 (TID 22)
16/12/09 10:01:59 INFO ShuffleBlockFetcherIterator: Getting 2 non-empty blocks out of 2 blocks
16/12/09 10:01:59 INFO ShuffleBlockFetcherIterator: Started 0 remote fetches in 0 ms
16/12/09 10:01:59 INFO Executor: Running task 1.0 in stage 12.0 (TID 23)
16/12/09 10:01:59 INFO ShuffleBlockFetcherIterator: Getting 2 non-empty blocks out of 2 blocks
16/12/09 10:01:59 INFO ShuffleBlockFetcherIterator: Started 0 remote fetches in 1 ms
16/12/09 10:01:59 INFO PythonRunner: Times: total = 100, boot = 6, init = 20, finish = 74
16/12/09 10:01:59 INFO Executor: Finished task 1.0 in stage 12.0 (TID 23). 414136 bytes result sent to driver
16/12/09 10:01:59 INFO PythonRunner: Times: total = 116, boot = 6, init = 22, finish = 88
16/12/09 10:01:59 INFO TaskSetManager: Finished task 1.0 in stage 12.0 (TID 23) in 127 ms on localhost (1/2)
16/12/09 10:01:59 INFO Executor: Finished task 0.0 in stage 12.0 (TID 22). 489351 bytes result sent to driver
16/12/09 10:01:59 INFO TaskSetManager: Finished task 0.0 in stage 12.0 (TID 22) in 131 ms on localhost (2/2)
16/12/09 10:01:59 INFO TaskSchedulerImpl: Removed TaskSet 12.0, whose tasks have all completed, from pool
16/12/09 10:01:59 INFO DAGScheduler: ResultStage 12 (collect at <stdin>:1) finished in 0.134 s
16/12/09 10:01:59 INFO DAGScheduler: Job 10 finished: collect at <stdin>:1, took 0.154957 s
((u'"', u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/080.txt'), 1)
>>>
```

Step9

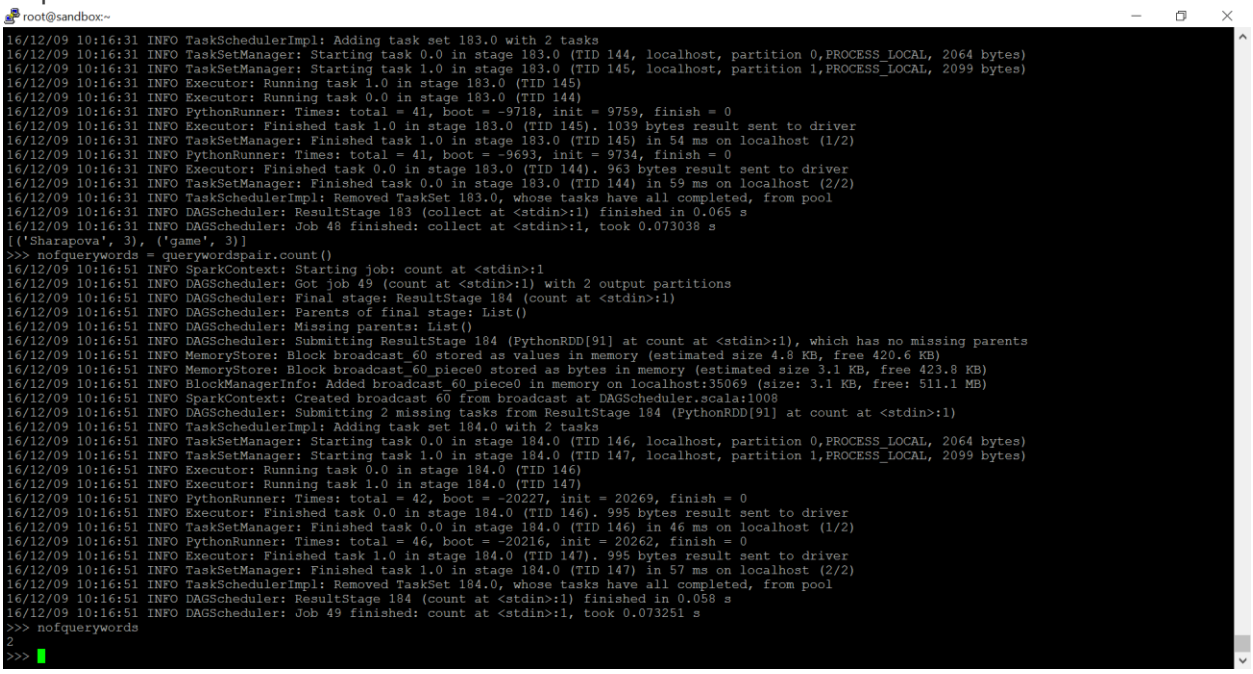
root@sandbox:~

```
16/12/09 10:03:43 INFO ShuffleBlockFetcherIterator: Getting 2 non-empty blocks out of 2 blocks
16/12/09 10:03:43 INFO ShuffleBlockFetcherIterator: Started 0 remote fetches in 0 ms
16/12/09 10:03:43 INFO PythonRunner: Times: total = 86, boot = -32265, init = 32275, finish = 76
16/12/09 10:03:43 INFO Executor: Finished task 0.0 in stage 30.0 (TID 36). 143938 bytes result sent to driver
16/12/09 10:03:43 INFO TaskSetManager: Finished task 0.0 in stage 30.0 (TID 36) in 98 ms on localhost (1/2)
16/12/09 10:03:43 INFO PythonRunner: Times: total = 96, boot = -32281, init = 32294, finish = 83
16/12/09 10:03:43 INFO Executor: Finished task 1.0 in stage 30.0 (TID 37). 122569 bytes result sent to driver
16/12/09 10:03:43 INFO TaskSetManager: Finished task 1.0 in stage 30.0 (TID 37) in 112 ms on localhost (2/2)
16/12/09 10:03:43 INFO TaskSchedulerImpl: Removed TaskSet 30.0, whose tasks have all completed, from pool
16/12/09 10:03:43 INFO DAGScheduler: ResultStage 30 (collect at <stdin>:1) finished in 0.119 s
16/12/09 10:03:43 INFO DAGScheduler: Job 15 finished: collect at <stdin>:1, took 0.132324 s
u'"'
>>> aa.values().collect()[0] # to verify
16/12/09 10:03:43 INFO SparkContext: Starting job: collect at <stdin>:1
16/12/09 10:03:43 INFO DAGScheduler: Got job 16 (collect at <stdin>:1) with 2 output partitions
16/12/09 10:03:43 INFO DAGScheduler: Final stage: ResultStage 34 (collect at <stdin>:1)
16/12/09 10:03:43 INFO DAGScheduler: Parents of final stage: List(ShuffleMapStage 33)
16/12/09 10:03:43 INFO DAGScheduler: Missing parents: List()
16/12/09 10:03:43 INFO DAGScheduler: Submitting ResultStage 34 (PythonRDD[29] at collect at <stdin>:1), which has no missing parents
16/12/09 10:03:43 INFO MemoryStore: Block broadcast_20 stored as values in memory (estimated size 5.8 KB, free 491.2 KB)
16/12/09 10:03:43 INFO MemoryStore: Block broadcast_20_piece0 stored as bytes in memory (estimated size 3.6 KB, free 494.8 KB)
16/12/09 10:03:43 INFO BlockManagerInfo: Added broadcast_20_piece0 in memory on localhost:35069 (size: 3.6 KB, free: 511.1 MB)
16/12/09 10:03:43 INFO SparkContext: Created broadcast 20 from broadcast at DAGScheduler.scala:1008
16/12/09 10:03:43 INFO DAGScheduler: Submitting 2 missing tasks from ResultStage 34 (PythonRDD[29] at collect at <stdin>:1)
16/12/09 10:03:43 INFO TaskSchedulerImpl: Adding task set 34.0 with 2 tasks
16/12/09 10:03:43 INFO TaskSetManager: Starting task 0.0 in stage 34.0 (TID 38, localhost, partition 0,NODE_LOCAL, 1894 bytes)
16/12/09 10:03:43 INFO TaskSetManager: Starting task 1.0 in stage 34.0 (TID 39, localhost, partition 1,NODE_LOCAL, 1894 bytes)
16/12/09 10:03:43 INFO Executor: Running task 0.0 in stage 34.0 (TID 38)
16/12/09 10:03:43 INFO Executor: Running task 1.0 in stage 34.0 (TID 39)
16/12/09 10:03:43 INFO ShuffleBlockFetcherIterator: Getting 2 non-empty blocks out of 2 blocks
16/12/09 10:03:43 INFO ShuffleBlockFetcherIterator: Started 0 remote fetches in 0 ms
16/12/09 10:03:43 INFO ShuffleBlockFetcherIterator: Getting 2 non-empty blocks out of 2 blocks
16/12/09 10:03:43 INFO ShuffleBlockFetcherIterator: Started 0 remote fetches in 1 ms
16/12/09 10:03:43 INFO PythonRunner: Times: total = 92, boot = -44, init = 56, finish = 80
16/12/09 10:03:43 INFO Executor: Finished task 0.0 in stage 34.0 (TID 38). 721673 bytes result sent to driver
16/12/09 10:03:43 INFO TaskSetManager: Finished task 0.0 in stage 34.0 (TID 38) in 108 ms on localhost (1/2)
16/12/09 10:03:43 INFO PythonRunner: Times: total = 112, boot = -49, init = 74, finish = 87
16/12/09 10:03:43 INFO Executor: Finished task 1.0 in stage 34.0 (TID 39). 644895 bytes result sent to driver
16/12/09 10:03:43 INFO TaskSetManager: Finished task 1.0 in stage 34.0 (TID 39) in 122 ms on localhost (2/2)
16/12/09 10:03:43 INFO TaskSchedulerImpl: Removed TaskSet 34.0, whose tasks have all completed, from pool
16/12/09 10:03:43 INFO DAGScheduler: ResultStage 34 (collect at <stdin>:1) finished in 0.125 s
16/12/09 10:03:43 INFO DAGScheduler: Job 16 finished: collect at <stdin>:1, took 0.143836 s
(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/080.txt', 1)
>>>
```

Step11



Step21



Step22

root@sandbox:~

```
16/12/09 10:18:11 INFO ShuffleBlockFetcherIterator: Started 0 remote fetches in 0 ms
16/12/09 10:18:11 INFO PythonRunner: Times: total = 66, boot = -52, init = 92, finish = 26
16/12/09 10:18:11 INFO Executor: Finished task 4.0 in stage 208.0 (TID 170). 1213 bytes result sent to driver
16/12/09 10:18:11 INFO TaskSetManager: Finished task 4.0 in stage 208.0 (TID 170) in 87 ms on localhost (5/6)
16/12/09 10:18:11 INFO PythonRunner: Times: total = 38, boot = -74, init = 97, finish = 15
16/12/09 10:18:11 INFO Executor: Finished task 5.0 in stage 208.0 (TID 171). 1213 bytes result sent to driver
16/12/09 10:18:11 INFO TaskSetManager: Finished task 5.0 in stage 208.0 (TID 171) in 48 ms on localhost (6/6)
16/12/09 10:18:11 INFO TaskSchedulerImpl: Removed TaskSet 208.0, whose tasks have all completed, from pool
16/12/09 10:18:11 INFO DAGScheduler: ResultStage 208 (collect at <stdin>:1) finished in 0.224 s
16/12/09 10:18:11 INFO DAGScheduler: Job 52 finished: collect at <stdin>:1, took 0.263222 s
[(u'game', ((u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/001.txt', 3, 0.47712125471966244, 1.4313637641589874), 3)), (u'game', ((u'hdfs:
//sandbox.hortonworks.com:8020/user/root/finalexam/tennis/007.txt', 1, 0.47712125471966244, 0.47712125471966244), 3)), (u'game', ((u'hdfs://sandbox.hortonwor
ks.com:8020/user/root/finalexam/tennis/008.txt', 1, 0.47712125471966244, 0.47712125471966244), 3)), (u'game', ((u'hdfs://sandbox.hortonworks.com:8020/user/ro
ot/finalexam/tennis/012.txt', 1, 0.47712125471966244, 0.47712125471966244), 3)), (u'game', ((u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis
/015.txt', 2, 0.47712125471966244, 0.95424250943932487), 3)), (u'game', ((u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/018.txt', 1, 0.477
12125471966244, 0.47712125471966244), 3)), (u'game', ((u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/019.txt', 1, 0.47712125471966244, 0.4
7712125471966244), 3)), (u'game', ((u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/020.txt', 1, 0.47712125471966244, 0.47712125471966244),
3)), (u'game', ((u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/022.txt', 1, 0.47712125471966244, 0.47712125471966244), 3)), (u'game', ((u'
hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/026.txt', 1, 0.47712125471966244, 0.47712125471966244), 3)), (u'game', ((u'hdfs://sandbox.hort
onworks.com:8020/user/root/finalexam/tennis/033.txt', 1, 0.47712125471966244, 0.47712125471966244), 3)), (u'game', ((u'hdfs://sandbox.hortonworks.com:8020/us
er/root/finalexam/tennis/039.txt', 2, 0.47712125471966244, 0.95424250943932487), 3)), (u'game', ((u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/t
ennis/052.txt', 2, 0.47712125471966244, 0.95424250943932487), 3)), (u'game', ((u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/053.txt', 1,
0.47712125471966244, 0.47712125471966244), 3)), (u'game', ((u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/054.txt', 1, 0.47712125471966244
, 0.47712125471966244), 3)), (u'game', ((u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/060.txt', 3, 0.47712125471966244, 1.431363764158987
4), 3)), (u'game', ((u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/061.txt', 2, 0.47712125471966244, 0.95424250943932487), 3)), (u'game',
((u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/063.txt', 2, 0.47712125471966244, 0.95424250943932487), 3)), (u'game', ((u'hdfs://sandbox.
hortonworks.com:8020/user/root/finalexam/tennis/066.txt', 1, 0.47712125471966244, 0.47712125471966244), 3)), (u'game', ((u'hdfs://sandbox.hortonworks.com:802
0/user/root/finalexam/tennis/068.txt', 3, 0.47712125471966244, 1.4313637641589874), 3)), (u'game', ((u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexa
m/tennis/071.txt', 14, 0.47712125471966244, 6.6796975660752738), 3)), (u'game', ((u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/072.txt',
1, 0.47712125471966244, 0.47712125471966244), 3)), (u'game', ((u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/073.txt', 2, 0.47712125471966
244, 0.95424250943932487), 3)), (u'game', ((u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/079.txt', 1, 0.47712125471966244, 0.477121254719
66244), 3)), (u'game', ((u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/087.txt', 2, 0.47712125471966244, 0.95424250943932487), 3)), (u'gam
e', ((u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/089.txt', 1, 0.47712125471966244, 0.47712125471966244), 3)), (u'game', ((u'hdfs://sand
box.hortonworks.com:8020/user/root/finalexam/tennis/090.txt', 1, 0.47712125471966244, 0.47712125471966244), 3)), (u'game', ((u'hdfs://sandbox.hortonworks.com
:8020/user/root/finalexam/tennis/096.txt', 1, 0.47712125471966244, 0.47712125471966244), 3)), (u'game', ((u'hdfs://sandbox.hortonworks.com:8020/user/root/fin
alexam/tennis/097.txt', 1, 0.47712125471966244, 0.47712125471966244), 3)), (u'game', ((u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/098.t
xt', 2, 0.47712125471966244, 0.95424250943932487), 3)), (u'game', ((u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/100.txt', 3, 0.477121254
71966244, 1.4313637641589874), 3)), (u'Sharapova', ((u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/047.txt', 1, 1.2041199826559248, 1.2041
199826559248), 3)), (u'Sharapova', ((u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/051.txt', 1, 1.2041199826559248, 1.2041199826559248), 3
)), (u'Sharapova', ((u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/054.txt', 2, 1.2041199826559248, 2.4082399653118496), 3)), (u'Sharapova
', ((u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/056.txt', 1, 1.2041199826559248, 1.2041199826559248), 3)), (u'Sharapova', ((u'hdfs://sa
ndbox.hortonworks.com:8020/user/root/finalexam/tennis/075.txt', 2, 1.2041199826559248, 2.4082399653118496), 3)), (u'Sharapova', ((u'hdfs://sandbox.hortonwork
s.com:8020/user/root/finalexam/tennis/100.txt', 8, 1.2041199826559248, 9.6329598612473983), 3))]
>>>
```

Step30/FinalOutput

root@sandbox:~

```
/finalexam/tennis/033.txt', 0.23856062735983122), (u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/019.txt', 0.23856062735983122), (u'hdfs:/
/sandbox.hortonworks.com:8020/user/root/finalexam/tennis/066.txt', 0.23856062735983122), (u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/09
0.txt', 0.23856062735983122), (u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/012.txt', 0.23856062735983122), (u'hdfs://sandbox.hortonworks
.com:8020/user/root/finalexam/tennis/018.txt', 0.23856062735983122), (u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/089.txt', 0.2385606273
5983122), (u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/053.txt', 0.23856062735983122), (u'hdfs://sandbox.hortonworks.com:8020/user/root/
finalexam/tennis/026.txt', 0.23856062735983122), (u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/022.txt', 0.23856062735983122), (u'hdfs://
sandbox.hortonworks.com:8020/user/root/finalexam/tennis/007.txt', 0.23856062735983122), (u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/020
.txt', 0.23856062735983122), (u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/097.txt', 0.23856062735983122)]
>>> finalanswer.take(topN)
16/12/09 10:35:27 INFO SparkContext: Starting job: runJob at PythonRDD.scala:393
16/12/09 10:35:27 INFO MapOutputTrackerMaster: Size of output statuses for shuffle 0 is 155 bytes
16/12/09 10:35:27 INFO MapOutputTrackerMaster: Size of output statuses for shuffle 1 is 155 bytes
16/12/09 10:35:27 INFO MapOutputTrackerMaster: Size of output statuses for shuffle 2 is 152 bytes
16/12/09 10:35:27 INFO MapOutputTrackerMaster: Size of output statuses for shuffle 3 is 155 bytes
16/12/09 10:35:27 INFO MapOutputTrackerMaster: Size of output statuses for shuffle 5 is 152 bytes
16/12/09 10:35:27 INFO MapOutputTrackerMaster: Size of output statuses for shuffle 6 is 172 bytes
16/12/09 10:35:27 INFO MapOutputTrackerMaster: Size of output statuses for shuffle 10 is 185 bytes
16/12/09 10:35:27 INFO MapOutputTrackerMaster: Size of output statuses for shuffle 11 is 172 bytes
16/12/09 10:35:27 INFO MapOutputTrackerMaster: Size of output statuses for shuffle 12 is 172 bytes
16/12/09 10:35:27 INFO MapOutputTrackerMaster: Size of output statuses for shuffle 13 is 179 bytes
16/12/09 10:35:27 INFO DAGScheduler: Got job 65 (runJob at PythonRDD.scala:393) with 1 output partitions
16/12/09 10:35:27 INFO DAGScheduler: Final stage: ResultStage 330 (runJob at PythonRDD.scala:393)
16/12/09 10:35:27 INFO DAGScheduler: Parents of final stage: List(ShuffleMapStage 329)
16/12/09 10:35:27 INFO DAGScheduler: Missing parents: List()
16/12/09 10:35:27 INFO DAGScheduler: Submitting ResultStage 330 (PythonRDD[127] at RDD at PythonRDD.scala:43), which has no missing parents
16/12/09 10:35:27 INFO MemoryStore: Block broadcast_80 stored as values in memory (estimated size 5.7 KB, free 694.9 KB)
16/12/09 10:35:27 INFO MemoryStore: Block broadcast_80_piece0 stored as bytes in memory (estimated size 3.6 KB, free 698.6 KB)
16/12/09 10:35:27 INFO BlockManagerInfo: Added broadcast_80_piece0 in memory on localhost:35069 (size: 3.6 KB, free: 511.0 MB)
16/12/09 10:35:27 INFO SparkContext: Created broadcast 80 from broadcast at DAGScheduler.scala:1008
16/12/09 10:35:27 INFO DAGScheduler: Submitting 1 missing tasks from ResultStage 330 (PythonRDD[127] at RDD at PythonRDD.scala:43)
16/12/09 10:35:27 INFO TaskSchedulerImpl: Adding task set 330.0 with 1 tasks
16/12/09 10:35:27 INFO TaskSetManager: Starting task 0.0 in stage 330.0 (TID 262, localhost, partition 0,NODE_LOCAL, 1894 bytes)
16/12/09 10:35:27 INFO Executor: Running task 0.0 in stage 330.0 (TID 262)
16/12/09 10:35:27 INFO ShuffleBlockFetcherIterator: Getting 6 non-empty blocks out of 6 blocks
16/12/09 10:35:27 INFO ShuffleBlockFetcherIterator: Started 0 remote fetches in 0 ms
16/12/09 10:35:27 INFO PythonRunner: Times: total = 52, boot = 10, init = 28, finish = 14
16/12/09 10:35:27 INFO Executor: Finished task 0.0 in stage 330.0 (TID 262). 1525 bytes result sent to driver
16/12/09 10:35:27 INFO TaskSetManager: Finished task 0.0 in stage 330.0 (TID 262) in 70 ms on localhost (1/1)
16/12/09 10:35:27 INFO TaskSchedulerImpl: Removed TaskSet 330.0, whose tasks have all completed, from pool
16/12/09 10:35:27 INFO DAGScheduler: ResultStage 330 (runJob at PythonRDD.scala:393) finished in 0.074 s
16/12/09 10:35:27 INFO DAGScheduler: Job 65 finished: runJob at PythonRDD.scala:393, took 0.147311 s
[(u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/100.txt', 11.064323625406386), (u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/
tennis/071.txt', 3.3398487830376369), (u'hdfs://sandbox.hortonworks.com:8020/user/root/finalexam/tennis/054.txt', 2.8853612200315122)]
>>>
```

12. REFERENCES

ThefollowinglinkswereusedtolearnthesyntaxforcodinginPYSPARK

https://spark.apache.org/docs/latest/programming-guide.html
https://www.safaribooksonline.com/library/view/learning-spark/9781449359034/ch04.html#chap-pair-RDDS
https://spark.apache.org/docs/1.1.1/api/python/pyspark.rdd.RDD-class.html#values
https://www.tutorialspoint.com/python/python_basic_operators.htm

DefinitionsandotherInformation

https://en.wikipedia.org/wiki/Tf%E2%80%93idf
http://www.ccs.neu.edu/home/cbw/spark.html

Datasets

D.GreeneandP.Cunningham."PracticalSolutionstotheProblemofDiagonalDominanceinKernelDocumentClustering",Proc.ICML2006.