|  |
| --- |
| Bhushan O. Kothari |
| Big Data: Document Similarity |
| Homework - 3 |

|  |
| --- |
| Bhushan Kothari  5/12/2015 |

**>>> r1 = sc.textFile('Documents/a3\_IP')  
>>> r1.collect()**

**//Read the input file**

[u'doc6:of,crazy,fox,jumped,fox,ran', u'doc6:fox,ran,fast,over,fence,too,high', u'doc1:fox,jumped', u'doc1:fox,jumped,over,the,fence', u'doc1:of,crazy,fox,jumped,fox,ran', u'doc1:fox,ran,fast,over,fence,too,high', u'doc2:a,crazy,fox,jumped', u'doc2:a,crazy,fox,jumped,over,the,fence,again', u'doc3:fox,ran,fast,over,fence', u'doc1:fox,is,high,on,sugar,but,jumped,but,failed', u'doc2:a,crazy,fox,ran,ran,ran,fast', u'doc3:a,crazy,fox,jumped,jumped,jumped,jumped', u'doc4:a,crazy,fox,ran,jumped,ran,jumped', u'doc4:a,crazy,fox,jumped,jumped', u'doc4:a,crazy,fox,jumped,jumped', u'doc5:a,crazy,fox,jumped,over,fence,very,high', u'doc5:a,crazy,fox,jumped,over,the,fence,again', u'doc6:book,reading,about,fox,and,fence,jumped,but,failed', u'doc1:crazy,fox,jumped,but,failed']  
  
**>>> r2 = r1.map(lambda x:(x.split(':')[0],x.split(':')[1].split(',')))  
>>> r2.collect()  
//Split the input on ‘:’ and then words on ‘,’**

[(u'doc6', [u'of', u'crazy', u'fox', u'jumped', u'fox', u'ran']), (u'doc6', [u'fox', u'ran', u'fast', u'over', u'fence', u'too', u'high']), (u'doc1', [u'fox', u'jumped']), (u'doc1', [u'fox', u'jumped', u'over', u'the', u'fence']), (u'doc1', [u'of', u'crazy', u'fox', u'jumped', u'fox', u'ran']), (u'doc1', [u'fox', u'ran', u'fast', u'over', u'fence', u'too', u'high']), (u'doc2', [u'a', u'crazy', u'fox', u'jumped']), (u'doc2', [u'a', u'crazy', u'fox', u'jumped', u'over', u'the', u'fence', u'again']), (u'doc3', [u'fox', u'ran', u'fast', u'over', u'fence']), (u'doc1', [u'fox', u'is', u'high', u'on', u'sugar', u'but', u'jumped', u'but', u'failed']), (u'doc2', [u'a', u'crazy', u'fox', u'ran', u'ran', u'ran', u'fast']), (u'doc3', [u'a', u'crazy', u'fox', u'jumped', u'jumped', u'jumped', u'jumped']), (u'doc4', [u'a', u'crazy', u'fox', u'ran', u'jumped', u'ran', u'jumped']), (u'doc4', [u'a', u'crazy', u'fox', u'jumped', u'jumped']), (u'doc4', [u'a', u'crazy', u'fox', u'jumped', u'jumped']), (u'doc5', [u'a', u'crazy', u'fox', u'jumped', u'over', u'fence', u'very', u'high']), (u'doc5', [u'a', u'crazy', u'fox', u'jumped', u'over', u'the', u'fence', u'again']), (u'doc6', [u'book', u'reading', u'about', u'fox', u'and', u'fence', u'jumped', u'but', u'failed']), (u'doc1', [u'crazy', u'fox', u'jumped', u'but', u'failed'])]  
  
**>>> n = input('Enter value of n')  
Enter value of n2  
>>>   
>>> def f(r2):  
     list=[]  
     for i in range(len(r2)-(n-1)):  
         list.append(tuple(r2[i:i+n]))  
     return list**  
**>>> r3 = r2.mapValues(f)  
//We accept the value of n from user and create a function which creates the tuples depending upon value of n. Then we use mapValues to operate on the values of the RDD.**

[(u'doc6', [(u'of', u'crazy'), (u'crazy', u'fox'), (u'fox', u'jumped'), (u'jumped', u'fox'), (u'fox', u'ran')]), (u'doc6', [(u'fox', u'ran'), (u'ran', u'fast'), (u'fast', u'over'), (u'over', u'fence'), (u'fence', u'too'), (u'too', u'high')]), (u'doc1', [(u'fox', u'jumped')]), (u'doc1', [(u'fox', u'jumped'), (u'jumped', u'over'), (u'over', u'the'), (u'the', u'fence')]), (u'doc1', [(u'of', u'crazy'), (u'crazy', u'fox'), (u'fox', u'jumped'), (u'jumped', u'fox'), (u'fox', u'ran')]), (u'doc1', [(u'fox', u'ran'), (u'ran', u'fast'), (u'fast', u'over'), (u'over', u'fence'), (u'fence', u'too'), (u'too', u'high')]), (u'doc2', [(u'a', u'crazy'), (u'crazy', u'fox'), (u'fox', u'jumped')]), (u'doc2', [(u'a', u'crazy'), (u'crazy', u'fox'), (u'fox', u'jumped'), (u'jumped', u'over'), (u'over', u'the'), (u'the', u'fence'), (u'fence', u'again')]), (u'doc3', [(u'fox', u'ran'), (u'ran', u'fast'), (u'fast', u'over'), (u'over', u'fence')]), (u'doc1', [(u'fox', u'is'), (u'is', u'high'), (u'high', u'on'), (u'on', u'sugar'), (u'sugar', u'but'), (u'but', u'jumped'), (u'jumped', u'but'), (u'but', u'failed')]), (u'doc2', [(u'a', u'crazy'), (u'crazy', u'fox'), (u'fox', u'ran'), (u'ran', u'ran'), (u'ran', u'ran'), (u'ran', u'fast')]), (u'doc3', [(u'a', u'crazy'), (u'crazy', u'fox'), (u'fox', u'jumped'), (u'jumped', u'jumped'), (u'jumped', u'jumped'), (u'jumped', u'jumped')]), (u'doc4', [(u'a', u'crazy'), (u'crazy', u'fox'), (u'fox', u'ran'), (u'ran', u'jumped'), (u'jumped', u'ran'), (u'ran', u'jumped')]), (u'doc4', [(u'a', u'crazy'), (u'crazy', u'fox'), (u'fox', u'jumped'), (u'jumped', u'jumped')]), (u'doc4', [(u'a', u'crazy'), (u'crazy', u'fox'), (u'fox', u'jumped'), (u'jumped', u'jumped')]), (u'doc5', [(u'a', u'crazy'), (u'crazy', u'fox'), (u'fox', u'jumped'), (u'jumped', u'over'), (u'over', u'fence'), (u'fence', u'very'), (u'very', u'high')]), (u'doc5', [(u'a', u'crazy'), (u'crazy', u'fox'), (u'fox', u'jumped'), (u'jumped', u'over'), (u'over', u'the'), (u'the', u'fence'), (u'fence', u'again')]), (u'doc6', [(u'book', u'reading'), (u'reading', u'about'), (u'about', u'fox'), (u'fox', u'and'), (u'and', u'fence'), (u'fence', u'jumped'), (u'jumped', u'but'), (u'but', u'failed')]), (u'doc1', [(u'crazy', u'fox'), (u'fox', u'jumped'), (u'jumped', u'but'), (u'but', u'failed')])]  
  
  
**>>> r4 = r3.flatMapValues(lambda x:x)  
>>> r4.collect()**

**//flatMapValues helps in associating Key with every value.**

[(u'doc6', (u'of', u'crazy')), (u'doc6', (u'crazy', u'fox')), (u'doc6', (u'fox', u'jumped')), (u'doc6', (u'jumped', u'fox')), (u'doc6', (u'fox', u'ran')), (u'doc6', (u'fox', u'ran')), (u'doc6', (u'ran', u'fast')), (u'doc6', (u'fast', u'over')), (u'doc6', (u'over', u'fence')), (u'doc6', (u'fence', u'too')), (u'doc6', (u'too', u'high')), (u'doc1', (u'fox', u'jumped')), (u'doc1', (u'fox', u'jumped')), (u'doc1', (u'jumped', u'over')), (u'doc1', (u'over', u'the')), (u'doc1', (u'the', u'fence')), (u'doc1', (u'of', u'crazy')), (u'doc1', (u'crazy', u'fox')), (u'doc1', (u'fox', u'jumped')), (u'doc1', (u'jumped', u'fox')), (u'doc1', (u'fox', u'ran')), (u'doc1', (u'fox', u'ran')), (u'doc1', (u'ran', u'fast')), (u'doc1', (u'fast', u'over')), (u'doc1', (u'over', u'fence')), (u'doc1', (u'fence', u'too')), (u'doc1', (u'too', u'high')), (u'doc2', (u'a', u'crazy')), (u'doc2', (u'crazy', u'fox')), (u'doc2', (u'fox', u'jumped')), (u'doc2', (u'a', u'crazy')), (u'doc2', (u'crazy', u'fox')), (u'doc2', (u'fox', u'jumped')), (u'doc2', (u'jumped', u'over')), (u'doc2', (u'over', u'the')), (u'doc2', (u'the', u'fence')), (u'doc2', (u'fence', u'again')), (u'doc3', (u'fox', u'ran')), (u'doc3', (u'ran', u'fast')), (u'doc3', (u'fast', u'over')), (u'doc3', (u'over', u'fence')), (u'doc1', (u'fox', u'is')), (u'doc1', (u'is', u'high')), (u'doc1', (u'high', u'on')), (u'doc1', (u'on', u'sugar')), (u'doc1', (u'sugar', u'but')), (u'doc1', (u'but', u'jumped')), (u'doc1', (u'jumped', u'but')), (u'doc1', (u'but', u'failed')), (u'doc2', (u'a', u'crazy')), (u'doc2', (u'crazy', u'fox')), (u'doc2', (u'fox', u'ran')), (u'doc2', (u'ran', u'ran')), (u'doc2', (u'ran', u'ran')), (u'doc2', (u'ran', u'fast')), (u'doc3', (u'a', u'crazy')), (u'doc3', (u'crazy', u'fox')), (u'doc3', (u'fox', u'jumped')), (u'doc3', (u'jumped', u'jumped')), (u'doc3', (u'jumped', u'jumped')), (u'doc3', (u'jumped', u'jumped')), (u'doc4', (u'a', u'crazy')), (u'doc4', (u'crazy', u'fox')), (u'doc4', (u'fox', u'ran')), (u'doc4', (u'ran', u'jumped')), (u'doc4', (u'jumped', u'ran')), (u'doc4', (u'ran', u'jumped')), (u'doc4', (u'a', u'crazy')), (u'doc4', (u'crazy', u'fox')), (u'doc4', (u'fox', u'jumped')), (u'doc4', (u'jumped', u'jumped')), (u'doc4', (u'a', u'crazy')), (u'doc4', (u'crazy', u'fox')), (u'doc4', (u'fox', u'jumped')), (u'doc4', (u'jumped', u'jumped')), (u'doc5', (u'a', u'crazy')), (u'doc5', (u'crazy', u'fox')), (u'doc5', (u'fox', u'jumped')), (u'doc5', (u'jumped', u'over')), (u'doc5', (u'over', u'fence')), (u'doc5', (u'fence', u'very')), (u'doc5', (u'very', u'high')), (u'doc5', (u'a', u'crazy')), (u'doc5', (u'crazy', u'fox')), (u'doc5', (u'fox', u'jumped')), (u'doc5', (u'jumped', u'over')), (u'doc5', (u'over', u'the')), (u'doc5', (u'the', u'fence')), (u'doc5', (u'fence', u'again')), (u'doc6', (u'book', u'reading')), (u'doc6', (u'reading', u'about')), (u'doc6', (u'about', u'fox')), (u'doc6', (u'fox', u'and')), (u'doc6', (u'and', u'fence')), (u'doc6', (u'fence', u'jumped')), (u'doc6', (u'jumped', u'but')), (u'doc6', (u'but', u'failed')), (u'doc1', (u'crazy', u'fox')), (u'doc1', (u'fox', u'jumped')), (u'doc1', (u'jumped', u'but')), (u'doc1', (u'but', u'failed'))]  
  
**>>> r5 = r4.map(lambda x:(x[0],(x[1],1)))  
>>> r5.collect()  
//appending 1 to every doc,ngrams instance**

[(u'doc6', ((u'of', u'crazy'), 1)), (u'doc6', ((u'crazy', u'fox'), 1)), (u'doc6', ((u'fox', u'jumped'), 1)), (u'doc6', ((u'jumped', u'fox'), 1)), (u'doc6', ((u'fox', u'ran'), 1)), (u'doc6', ((u'fox', u'ran'), 1)), (u'doc6', ((u'ran', u'fast'), 1)), (u'doc6', ((u'fast', u'over'), 1)), (u'doc6', ((u'over', u'fence'), 1)), (u'doc6', ((u'fence', u'too'), 1)), (u'doc6', ((u'too', u'high'), 1)), (u'doc1', ((u'fox', u'jumped'), 1)), (u'doc1', ((u'fox', u'jumped'), 1)), (u'doc1', ((u'jumped', u'over'), 1)), (u'doc1', ((u'over', u'the'), 1)), (u'doc1', ((u'the', u'fence'), 1)), (u'doc1', ((u'of', u'crazy'), 1)), (u'doc1', ((u'crazy', u'fox'), 1)), (u'doc1', ((u'fox', u'jumped'), 1)), (u'doc1', ((u'jumped', u'fox'), 1)), (u'doc1', ((u'fox', u'ran'), 1)), (u'doc1', ((u'fox', u'ran'), 1)), (u'doc1', ((u'ran', u'fast'), 1)), (u'doc1', ((u'fast', u'over'), 1)), (u'doc1', ((u'over', u'fence'), 1)), (u'doc1', ((u'fence', u'too'), 1)), (u'doc1', ((u'too', u'high'), 1)), (u'doc2', ((u'a', u'crazy'), 1)), (u'doc2', ((u'crazy', u'fox'), 1)), (u'doc2', ((u'fox', u'jumped'), 1)), (u'doc2', ((u'a', u'crazy'), 1)), (u'doc2', ((u'crazy', u'fox'), 1)), (u'doc2', ((u'fox', u'jumped'), 1)), (u'doc2', ((u'jumped', u'over'), 1)), (u'doc2', ((u'over', u'the'), 1)), (u'doc2', ((u'the', u'fence'), 1)), (u'doc2', ((u'fence', u'again'), 1)), (u'doc3', ((u'fox', u'ran'), 1)), (u'doc3', ((u'ran', u'fast'), 1)), (u'doc3', ((u'fast', u'over'), 1)), (u'doc3', ((u'over', u'fence'), 1)), (u'doc1', ((u'fox', u'is'), 1)), (u'doc1', ((u'is', u'high'), 1)), (u'doc1', ((u'high', u'on'), 1)), (u'doc1', ((u'on', u'sugar'), 1)), (u'doc1', ((u'sugar', u'but'), 1)), (u'doc1', ((u'but', u'jumped'), 1)), (u'doc1', ((u'jumped', u'but'), 1)), (u'doc1', ((u'but', u'failed'), 1)), (u'doc2', ((u'a', u'crazy'), 1)), (u'doc2', ((u'crazy', u'fox'), 1)), (u'doc2', ((u'fox', u'ran'), 1)), (u'doc2', ((u'ran', u'ran'), 1)), (u'doc2', ((u'ran', u'ran'), 1)), (u'doc2', ((u'ran', u'fast'), 1)), (u'doc3', ((u'a', u'crazy'), 1)), (u'doc3', ((u'crazy', u'fox'), 1)), (u'doc3', ((u'fox', u'jumped'), 1)), (u'doc3', ((u'jumped', u'jumped'), 1)), (u'doc3', ((u'jumped', u'jumped'), 1)), (u'doc3', ((u'jumped', u'jumped'), 1)), (u'doc4', ((u'a', u'crazy'), 1)), (u'doc4', ((u'crazy', u'fox'), 1)), (u'doc4', ((u'fox', u'ran'), 1)), (u'doc4', ((u'ran', u'jumped'), 1)), (u'doc4', ((u'jumped', u'ran'), 1)), (u'doc4', ((u'ran', u'jumped'), 1)), (u'doc4', ((u'a', u'crazy'), 1)), (u'doc4', ((u'crazy', u'fox'), 1)), (u'doc4', ((u'fox', u'jumped'), 1)), (u'doc4', ((u'jumped', u'jumped'), 1)), (u'doc4', ((u'a', u'crazy'), 1)), (u'doc4', ((u'crazy', u'fox'), 1)), (u'doc4', ((u'fox', u'jumped'), 1)), (u'doc4', ((u'jumped', u'jumped'), 1)), (u'doc5', ((u'a', u'crazy'), 1)), (u'doc5', ((u'crazy', u'fox'), 1)), (u'doc5', ((u'fox', u'jumped'), 1)), (u'doc5', ((u'jumped', u'over'), 1)), (u'doc5', ((u'over', u'fence'), 1)), (u'doc5', ((u'fence', u'very'), 1)), (u'doc5', ((u'very', u'high'), 1)), (u'doc5', ((u'a', u'crazy'), 1)), (u'doc5', ((u'crazy', u'fox'), 1)), (u'doc5', ((u'fox', u'jumped'), 1)), (u'doc5', ((u'jumped', u'over'), 1)), (u'doc5', ((u'over', u'the'), 1)), (u'doc5', ((u'the', u'fence'), 1)), (u'doc5', ((u'fence', u'again'), 1)), (u'doc6', ((u'book', u'reading'), 1)), (u'doc6', ((u'reading', u'about'), 1)), (u'doc6', ((u'about', u'fox'), 1)), (u'doc6', ((u'fox', u'and'), 1)), (u'doc6', ((u'and', u'fence'), 1)), (u'doc6', ((u'fence', u'jumped'), 1)), (u'doc6', ((u'jumped', u'but'), 1)), (u'doc6', ((u'but', u'failed'), 1)), (u'doc1', ((u'crazy', u'fox'), 1)), (u'doc1', ((u'fox', u'jumped'), 1)), (u'doc1', ((u'jumped', u'but'), 1)), (u'doc1', ((u'but', u'failed'), 1))]

**>>> r6 = r5.map(lambda x:((x[0],x[1][0]),x[1][1]))  
>>> r6.collect()**

[((u'doc6', (u'of', u'crazy')), 1), ((u'doc6', (u'crazy', u'fox')), 1), ((u'doc6', (u'fox', u'jumped')), 1), ((u'doc6', (u'jumped', u'fox')), 1), ((u'doc6', (u'fox', u'ran')), 1), ((u'doc6', (u'fox', u'ran')), 1), ((u'doc6', (u'ran', u'fast')), 1), ((u'doc6', (u'fast', u'over')), 1), ((u'doc6', (u'over', u'fence')), 1), ((u'doc6', (u'fence', u'too')), 1), ((u'doc6', (u'too', u'high')), 1), ((u'doc1', (u'fox', u'jumped')), 1), ((u'doc1', (u'fox', u'jumped')), 1), ((u'doc1', (u'jumped', u'over')), 1), ((u'doc1', (u'over', u'the')), 1), ((u'doc1', (u'the', u'fence')), 1), ((u'doc1', (u'of', u'crazy')), 1), ((u'doc1', (u'crazy', u'fox')), 1), ((u'doc1', (u'fox', u'jumped')), 1), ((u'doc1', (u'jumped', u'fox')), 1), ((u'doc1', (u'fox', u'ran')), 1), ((u'doc1', (u'fox', u'ran')), 1), ((u'doc1', (u'ran', u'fast')), 1), ((u'doc1', (u'fast', u'over')), 1), ((u'doc1', (u'over', u'fence')), 1), ((u'doc1', (u'fence', u'too')), 1), ((u'doc1', (u'too', u'high')), 1), ((u'doc2', (u'a', u'crazy')), 1), ((u'doc2', (u'crazy', u'fox')), 1), ((u'doc2', (u'fox', u'jumped')), 1), ((u'doc2', (u'a', u'crazy')), 1), ((u'doc2', (u'crazy', u'fox')), 1), ((u'doc2', (u'fox', u'jumped')), 1), ((u'doc2', (u'jumped', u'over')), 1), ((u'doc2', (u'over', u'the')), 1), ((u'doc2', (u'the', u'fence')), 1), ((u'doc2', (u'fence', u'again')), 1), ((u'doc3', (u'fox', u'ran')), 1), ((u'doc3', (u'ran', u'fast')), 1), ((u'doc3', (u'fast', u'over')), 1), ((u'doc3', (u'over', u'fence')), 1), ((u'doc1', (u'fox', u'is')), 1), ((u'doc1', (u'is', u'high')), 1), ((u'doc1', (u'high', u'on')), 1), ((u'doc1', (u'on', u'sugar')), 1), ((u'doc1', (u'sugar', u'but')), 1), ((u'doc1', (u'but', u'jumped')), 1), ((u'doc1', (u'jumped', u'but')), 1), ((u'doc1', (u'but', u'failed')), 1), ((u'doc2', (u'a', u'crazy')), 1), ((u'doc2', (u'crazy', u'fox')), 1), ((u'doc2', (u'fox', u'ran')), 1), ((u'doc2', (u'ran', u'ran')), 1), ((u'doc2', (u'ran', u'ran')), 1), ((u'doc2', (u'ran', u'fast')), 1), ((u'doc3', (u'a', u'crazy')), 1), ((u'doc3', (u'crazy', u'fox')), 1), ((u'doc3', (u'fox', u'jumped')), 1), ((u'doc3', (u'jumped', u'jumped')), 1), ((u'doc3', (u'jumped', u'jumped')), 1), ((u'doc3', (u'jumped', u'jumped')), 1), ((u'doc4', (u'a', u'crazy')), 1), ((u'doc4', (u'crazy', u'fox')), 1), ((u'doc4', (u'fox', u'ran')), 1), ((u'doc4', (u'ran', u'jumped')), 1), ((u'doc4', (u'jumped', u'ran')), 1), ((u'doc4', (u'ran', u'jumped')), 1), ((u'doc4', (u'a', u'crazy')), 1), ((u'doc4', (u'crazy', u'fox')), 1), ((u'doc4', (u'fox', u'jumped')), 1), ((u'doc4', (u'jumped', u'jumped')), 1), ((u'doc4', (u'a', u'crazy')), 1), ((u'doc4', (u'crazy', u'fox')), 1), ((u'doc4', (u'fox', u'jumped')), 1), ((u'doc4', (u'jumped', u'jumped')), 1), ((u'doc5', (u'a', u'crazy')), 1), ((u'doc5', (u'crazy', u'fox')), 1), ((u'doc5', (u'fox', u'jumped')), 1), ((u'doc5', (u'jumped', u'over')), 1), ((u'doc5', (u'over', u'fence')), 1), ((u'doc5', (u'fence', u'very')), 1), ((u'doc5', (u'very', u'high')), 1), ((u'doc5', (u'a', u'crazy')), 1), ((u'doc5', (u'crazy', u'fox')), 1), ((u'doc5', (u'fox', u'jumped')), 1), ((u'doc5', (u'jumped', u'over')), 1), ((u'doc5', (u'over', u'the')), 1), ((u'doc5', (u'the', u'fence')), 1), ((u'doc5', (u'fence', u'again')), 1), ((u'doc6', (u'book', u'reading')), 1), ((u'doc6', (u'reading', u'about')), 1), ((u'doc6', (u'about', u'fox')), 1), ((u'doc6', (u'fox', u'and')), 1), ((u'doc6', (u'and', u'fence')), 1), ((u'doc6', (u'fence', u'jumped')), 1), ((u'doc6', (u'jumped', u'but')), 1), ((u'doc6', (u'but', u'failed')), 1), ((u'doc1', (u'crazy', u'fox')), 1), ((u'doc1', (u'fox', u'jumped')), 1), ((u'doc1', (u'jumped', u'but')), 1), ((u'doc1', (u'but', u'failed')), 1)]  
>>>   
  
  
**>>> r7 = r6.reduceByKey(lambda x,y:x+y).map(lambda x:(x[0][0],(x[0][1],x[1])))  
>>> r7.collect()  
// Find the count of certain ngrams in a particular document**

[(u'doc1', ((u'high', u'on'), 1)), (u'doc6', ((u'fence', u'jumped'), 1)), (u'doc3', ((u'fast', u'over'), 1)), (u'doc1', ((u'over', u'fence'), 1)), (u'doc6', ((u'but', u'failed'), 1)), (u'doc6', ((u'book', u'reading'), 1)), (u'doc3', ((u'crazy', u'fox'), 1)), (u'doc5', ((u'crazy', u'fox'), 2)), (u'doc1', ((u'fox', u'jumped'), 4)), (u'doc5', ((u'fence', u'again'), 1)), (u'doc6', ((u'fox', u'ran'), 2)), (u'doc5', ((u'a', u'crazy'), 2)), (u'doc6', ((u'and', u'fence'), 1)), (u'doc1', ((u'the', u'fence'), 1)), (u'doc4', ((u'crazy', u'fox'), 3)), (u'doc2', ((u'fox', u'jumped'), 2)), (u'doc2', ((u'jumped', u'over'), 1)), (u'doc5', ((u'jumped', u'over'), 2)), (u'doc6', ((u'fast', u'over'), 1)), (u'doc3', ((u'ran', u'fast'), 1)), (u'doc6', ((u'crazy', u'fox'), 1)), (u'doc5', ((u'very', u'high'), 1)), (u'doc4', ((u'jumped', u'ran'), 1)), (u'doc4', ((u'fox', u'ran'), 1)), (u'doc6', ((u'fence', u'too'), 1)), (u'doc1', ((u'jumped', u'fox'), 1)), (u'doc1', ((u'of', u'crazy'), 1)), (u'doc1', ((u'fox', u'ran'), 2)), (u'doc1', ((u'but', u'failed'), 2)), (u'doc1', ((u'is', u'high'), 1)), (u'doc2', ((u'crazy', u'fox'), 3)), (u'doc2', ((u'ran', u'ran'), 2)), (u'doc3', ((u'fox', u'jumped'), 1)), (u'doc5', ((u'over', u'fence'), 1)), (u'doc3', ((u'over', u'fence'), 1)), (u'doc4', ((u'a', u'crazy'), 3)), (u'doc1', ((u'crazy', u'fox'), 2)), (u'doc6', ((u'fox', u'jumped'), 1)), (u'doc3', ((u'jumped', u'jumped'), 3)), (u'doc6', ((u'of', u'crazy'), 1)), (u'doc6', ((u'about', u'fox'), 1)), (u'doc4', ((u'fox', u'jumped'), 2)), (u'doc6', ((u'ran', u'fast'), 1)), (u'doc4', ((u'ran', u'jumped'), 2)), (u'doc5', ((u'the', u'fence'), 1)), (u'doc1', ((u'fence', u'too'), 1)), (u'doc6', ((u'fox', u'and'), 1)), (u'doc1', ((u'fast', u'over'), 1)), (u'doc1', ((u'too', u'high'), 1)), (u'doc3', ((u'a', u'crazy'), 1)), (u'doc6', ((u'jumped', u'fox'), 1)), (u'doc6', ((u'over', u'fence'), 1)), (u'doc3', ((u'fox', u'ran'), 1)), (u'doc2', ((u'the', u'fence'), 1)), (u'doc1', ((u'ran', u'fast'), 1)), (u'doc1', ((u'jumped', u'over'), 1)), (u'doc1', ((u'fox', u'is'), 1)), (u'doc1', ((u'jumped', u'but'), 2)), (u'doc2', ((u'over', u'the'), 1)), (u'doc2', ((u'ran', u'fast'), 1)), (u'doc6', ((u'reading', u'about'), 1)), (u'doc1', ((u'sugar', u'but'), 1)), (u'doc2', ((u'fence', u'again'), 1)), (u'doc2', ((u'fox', u'ran'), 1)), (u'doc5', ((u'fox', u'jumped'), 2)), (u'doc1', ((u'over', u'the'), 1)), (u'doc1', ((u'but', u'jumped'), 1)), (u'doc6', ((u'jumped', u'but'), 1)), (u'doc2', ((u'a', u'crazy'), 3)), (u'doc4', ((u'jumped', u'jumped'), 2)), (u'doc1', ((u'on', u'sugar'), 1)), (u'doc5', ((u'over', u'the'), 1)), (u'doc5', ((u'fence', u'very'), 1)), (u'doc6', ((u'too', u'high'), 1))]  
  
  
**>>> r8 = r7.cartesian(r7)  
>>> r8.collect()**

**// Output too huge. Thus excluded for readability purposes. However its of form ((u'doc5', ((u'fence', u'very'), 1)), (u'doc6', ((u'too', u'high'), 1))**

**>>> r9 = r8.filter(lambda x: x[0][0] < x[1][0])  
>>> r9.collect()  
//It filters out records where d1<d2 (To avoid repetition and same document rows)**

**r10 = r9.filter(lambda x: x[0][1][0] == x[1][1][0])**

**// Output too huge. Thus excluded for readability purposes. It filters the instances of same ngramss present in both the documents.  
  
r11 = r10.map(lambda x:((x[0][0],x[1][0]),(x[0][1][1],x[1][1][1])))  
>>> r11.collect()  
//We can ignore the ngrams, as we need the counts of their occurrences.**

[((u'doc3', u'doc6'), (1, 1)), ((u'doc3', u'doc5'), (1, 2)), ((u'doc3', u'doc4'), (1, 3)), ((u'doc3', u'doc6'), (1, 1)), ((u'doc1', u'doc5'), (1, 1)), ((u'doc1', u'doc3'), (1, 1)), ((u'doc1', u'doc6'), (1, 1)), ((u'doc4', u'doc5'), (3, 2)), ((u'doc5', u'doc6'), (2, 1)), ((u'doc1', u'doc2'), (4, 2)), ((u'doc4', u'doc6'), (3, 1)), ((u'doc1', u'doc3'), (4, 1)), ((u'doc1', u'doc6'), (4, 1)), ((u'doc1', u'doc4'), (4, 2)), ((u'doc1', u'doc5'), (1, 1)), ((u'doc1', u'doc2'), (1, 1)), ((u'doc1', u'doc5'), (4, 2)), ((u'doc1', u'doc6'), (2, 1)), ((u'doc2', u'doc3'), (3, 1)), ((u'doc4', u'doc6'), (1, 2)), ((u'doc1', u'doc6'), (2, 2)), ((u'doc2', u'doc5'), (3, 2)), ((u'doc2', u'doc4'), (3, 3)), ((u'doc2', u'doc5'), (1, 2)), ((u'doc1', u'doc4'), (2, 1)), ((u'doc2', u'doc6'), (3, 1)), ((u'doc2', u'doc3'), (2, 1)), ((u'doc2', u'doc6'), (2, 1)), ((u'doc2', u'doc4'), (2, 2)), ((u'doc3', u'doc6'), (1, 1)), ((u'doc1', u'doc6'), (1, 1)), ((u'doc1', u'doc6'), (1, 1)), ((u'doc1', u'doc3'), (2, 1)), ((u'doc2', u'doc5'), (2, 2)), ((u'doc1', u'doc2'), (2, 1)), ((u'doc1', u'doc3'), (1, 1)), ((u'doc1', u'doc3'), (2, 1)), ((u'doc4', u'doc5'), (3, 2)), ((u'doc1', u'doc5'), (2, 2)), ((u'doc1', u'doc4'), (2, 3)), ((u'doc3', u'doc5'), (1, 2)), ((u'doc3', u'doc6'), (1, 2)), ((u'doc2', u'doc5'), (1, 1)), ((u'doc1', u'doc6'), (2, 1)), ((u'doc1', u'doc2'), (2, 3)), ((u'doc1', u'doc6'), (1, 1)), ((u'doc1', u'doc6'), (1, 1)), ((u'doc3', u'doc4'), (1, 1)), ((u'doc1', u'doc3'), (1, 1)), ((u'doc1', u'doc2'), (1, 1)), ((u'doc1', u'doc5'), (1, 2)), ((u'doc2', u'doc3'), (1, 1)), ((u'doc3', u'doc6'), (1, 1)), ((u'doc3', u'doc4'), (1, 2)), ((u'doc5', u'doc6'), (1, 1)), ((u'doc3', u'doc5'), (1, 1)), ((u'doc3', u'doc6'), (1, 1)), ((u'doc4', u'doc6'), (2, 1)), ((u'doc3', u'doc4'), (1, 3)), ((u'doc2', u'doc5'), (1, 1)), ((u'doc1', u'doc6'), (1, 1)), ((u'doc1', u'doc2'), (1, 1)), ((u'doc2', u'doc6'), (1, 1)), ((u'doc3', u'doc5'), (1, 2)), ((u'doc3', u'doc4'), (3, 2)), ((u'doc4', u'doc5'), (2, 2)), ((u'doc1', u'doc6'), (1, 1)), ((u'doc1', u'doc6'), (2, 1)), ((u'doc2', u'doc5'), (1, 1)), ((u'doc2', u'doc6'), (1, 2)), ((u'doc2', u'doc5'), (3, 2)), ((u'doc2', u'doc4'), (1, 1)), ((u'doc2', u'doc3'), (1, 1)), ((u'doc5', u'doc6'), (2, 1)), ((u'doc1', u'doc2'), (1, 1)), ((u'doc2', u'doc4'), (3, 3)), ((u'doc2', u'doc3'), (3, 1)), ((u'doc1', u'doc5'), (1, 1))]

**>>> r12 = r11.map(lambda x: (x[0], x[1][0] if (x[1][0]<x[1][1]) else x[1][1]))  
>>> r12.collect()**

**//Keep the minimum value present between the 2 documents.**

[((u'doc3', u'doc6'), 1), ((u'doc3', u'doc5'), 1), ((u'doc3', u'doc4'), 1), ((u'doc3', u'doc6'), 1), ((u'doc1', u'doc5'), 1), ((u'doc1', u'doc3'), 1), ((u'doc1', u'doc6'), 1), ((u'doc4', u'doc5'), 2), ((u'doc5', u'doc6'), 1), ((u'doc1', u'doc2'), 2), ((u'doc4', u'doc6'), 1), ((u'doc1', u'doc3'), 1), ((u'doc1', u'doc6'), 1), ((u'doc1', u'doc4'), 2), ((u'doc1', u'doc5'), 1), ((u'doc1', u'doc2'), 1), ((u'doc1', u'doc5'), 2), ((u'doc1', u'doc6'), 1), ((u'doc2', u'doc3'), 1), ((u'doc4', u'doc6'), 1), ((u'doc1', u'doc6'), 2), ((u'doc2', u'doc5'), 2), ((u'doc2', u'doc4'), 3), ((u'doc2', u'doc5'), 1), ((u'doc1', u'doc4'), 1), ((u'doc2', u'doc6'), 1), ((u'doc2', u'doc3'), 1), ((u'doc2', u'doc6'), 1), ((u'doc2', u'doc4'), 2), ((u'doc3', u'doc6'), 1), ((u'doc1', u'doc6'), 1), ((u'doc1', u'doc6'), 1), ((u'doc1', u'doc3'), 1), ((u'doc2', u'doc5'), 2), ((u'doc1', u'doc2'), 1), ((u'doc1', u'doc3'), 1), ((u'doc1', u'doc3'), 1), ((u'doc4', u'doc5'), 2), ((u'doc1', u'doc5'), 2), ((u'doc1', u'doc4'), 2), ((u'doc3', u'doc5'), 1), ((u'doc3', u'doc6'), 1), ((u'doc2', u'doc5'), 1), ((u'doc1', u'doc6'), 1), ((u'doc1', u'doc2'), 2), ((u'doc1', u'doc6'), 1), ((u'doc1', u'doc6'), 1), ((u'doc3', u'doc4'), 1), ((u'doc1', u'doc3'), 1), ((u'doc1', u'doc2'), 1), ((u'doc1', u'doc5'), 1), ((u'doc2', u'doc3'), 1), ((u'doc3', u'doc6'), 1), ((u'doc3', u'doc4'), 1), ((u'doc5', u'doc6'), 1), ((u'doc3', u'doc5'), 1), ((u'doc3', u'doc6'), 1), ((u'doc4', u'doc6'), 1), ((u'doc3', u'doc4'), 1), ((u'doc2', u'doc5'), 1), ((u'doc1', u'doc6'), 1), ((u'doc1', u'doc2'), 1), ((u'doc2', u'doc6'), 1), ((u'doc3', u'doc5'), 1), ((u'doc3', u'doc4'), 2), ((u'doc4', u'doc5'), 2), ((u'doc1', u'doc6'), 1), ((u'doc1', u'doc6'), 1), ((u'doc2', u'doc5'), 1), ((u'doc2', u'doc6'), 1), ((u'doc2', u'doc5'), 2), ((u'doc2', u'doc4'), 1), ((u'doc2', u'doc3'), 1), ((u'doc5', u'doc6'), 1), ((u'doc1', u'doc2'), 1), ((u'doc2', u'doc4'), 3), ((u'doc2', u'doc3'), 1), ((u'doc1', u'doc5'), 1)]  
>>> 

**>>> r13 = r12.reduceByKey(lambda x,y:x+y)  
>>> r13.collect()  
//Count the intersection ngrams in a doc.**

[((u'doc5', u'doc6'), 3), ((u'doc2', u'doc4'), 9), ((u'doc3', u'doc4'), 6), ((u'doc3', u'doc5'), 4), ((u'doc1', u'doc2'), 9), ((u'doc1', u'doc3'), 6), ((u'doc2', u'doc3'), 5), ((u'doc4', u'doc6'), 3), ((u'doc4', u'doc5'), 6), ((u'doc3', u'doc6'), 6), ((u'doc1', u'doc4'), 5), ((u'doc1', u'doc5'), 8), ((u'doc2', u'doc6'), 4), ((u'doc1', u'doc6'), 13), ((u'doc2', u'doc5'), 10)]  
>>>   
  
  
**>>> r14 = r7.map(lambda x: (x[0],x[1][1]))  
>>> r14.collect()**

**//Remove the ngramss as we need only count of ngramss in a documents.**

[(u'doc1', 1), (u'doc6', 1), (u'doc3', 1), (u'doc1', 1), (u'doc6', 1), (u'doc6', 1), (u'doc3', 1), (u'doc5', 2), (u'doc1', 4), (u'doc5', 1), (u'doc6', 2), (u'doc5', 2), (u'doc6', 1), (u'doc1', 1), (u'doc4', 3), (u'doc2', 2), (u'doc2', 1), (u'doc5', 2), (u'doc6', 1), (u'doc3', 1), (u'doc6', 1), (u'doc5', 1), (u'doc4', 1), (u'doc4', 1), (u'doc6', 1), (u'doc1', 1), (u'doc1', 1), (u'doc1', 2), (u'doc1', 2), (u'doc1', 1), (u'doc2', 3), (u'doc2', 2), (u'doc3', 1), (u'doc5', 1), (u'doc3', 1), (u'doc4', 3), (u'doc1', 2), (u'doc6', 1), (u'doc3', 3), (u'doc6', 1), (u'doc6', 1), (u'doc4', 2), (u'doc6', 1), (u'doc4', 2), (u'doc5', 1), (u'doc1', 1), (u'doc6', 1), (u'doc1', 1), (u'doc1', 1), (u'doc3', 1), (u'doc6', 1), (u'doc6', 1), (u'doc3', 1), (u'doc2', 1), (u'doc1', 1), (u'doc1', 1), (u'doc1', 1), (u'doc1', 2), (u'doc2', 1), (u'doc2', 1), (u'doc6', 1), (u'doc1', 1), (u'doc2', 1), (u'doc2', 1), (u'doc5', 2), (u'doc1', 1), (u'doc1', 1), (u'doc6', 1), (u'doc2', 3), (u'doc4', 2), (u'doc1', 1), (u'doc5', 1), (u'doc5', 1), (u'doc6', 1)]  
  
  
**>>> r15 = r14.reduceByKey(lambda x,y:x+y)  
>>>r15.collect()**

**//Count all the ngrams present in a doc.**

[(u'doc2', 16), (u'doc3', 10), (u'doc1', 28), (u'doc6', 19), (u'doc4', 14), (u'doc5', 14)]  
  
  
**>>> r16 = r15.cartesian(r15)  
>>> r16.collect()**

[((u'doc2', 16), (u'doc2', 16)), ((u'doc2', 16), (u'doc3', 10)), ((u'doc2', 16), (u'doc1', 28)), ((u'doc2', 16), (u'doc6', 19)), ((u'doc2', 16), (u'doc4', 14)), ((u'doc2', 16), (u'doc5', 14)), ((u'doc3', 10), (u'doc2', 16)), ((u'doc1', 28), (u'doc2', 16)), ((u'doc3', 10), (u'doc3', 10)), ((u'doc3', 10), (u'doc1', 28)), ((u'doc1', 28), (u'doc3', 10)), ((u'doc1', 28), (u'doc1', 28)), ((u'doc3', 10), (u'doc6', 19)), ((u'doc3', 10), (u'doc4', 14)), ((u'doc3', 10), (u'doc5', 14)), ((u'doc1', 28), (u'doc6', 19)), ((u'doc1', 28), (u'doc4', 14)), ((u'doc1', 28), (u'doc5', 14)), ((u'doc6', 19), (u'doc2', 16)), ((u'doc4', 14), (u'doc2', 16)), ((u'doc5', 14), (u'doc2', 16)), ((u'doc6', 19), (u'doc3', 10)), ((u'doc6', 19), (u'doc1', 28)), ((u'doc4', 14), (u'doc3', 10)), ((u'doc4', 14), (u'doc1', 28)), ((u'doc5', 14), (u'doc3', 10)), ((u'doc5', 14), (u'doc1', 28)), ((u'doc6', 19), (u'doc6', 19)), ((u'doc6', 19), (u'doc4', 14)), ((u'doc6', 19), (u'doc5', 14)), ((u'doc4', 14), (u'doc6', 19)), ((u'doc4', 14), (u'doc4', 14)), ((u'doc4', 14), (u'doc5', 14)), ((u'doc5', 14), (u'doc6', 19)), ((u'doc5', 14), (u'doc4', 14)), ((u'doc5', 14), (u'doc5', 14))]  
  
  
**>>> r17 = r16.filter(lambda x: x[0][0] < x[1][0])  
>>> r17.collect()**

[((u'doc2', 16), (u'doc3', 10)), ((u'doc2', 16), (u'doc6', 19)), ((u'doc2', 16), (u'doc4', 14)), ((u'doc2', 16), (u'doc5', 14)), ((u'doc1', 28), (u'doc2', 16)), ((u'doc1', 28), (u'doc3', 10)), ((u'doc3', 10), (u'doc6', 19)), ((u'doc3', 10), (u'doc4', 14)), ((u'doc3', 10), (u'doc5', 14)), ((u'doc1', 28), (u'doc6', 19)), ((u'doc1', 28), (u'doc4', 14)), ((u'doc1', 28), (u'doc5', 14)), ((u'doc4', 14), (u'doc6', 19)), ((u'doc4', 14), (u'doc5', 14)), ((u'doc5', 14), (u'doc6', 19))]  
  
  
**>>> r18 = r17.map(lambda x:((x[0][0],x[1][0]),(x[0][1],x[1][1])))  
>>> r18.collect()**

[((u'doc2', u'doc3'), (16, 10)), ((u'doc2', u'doc6'), (16, 19)), ((u'doc2', u'doc4'), (16, 14)), ((u'doc2', u'doc5'), (16, 14)), ((u'doc1', u'doc2'), (28, 16)), ((u'doc1', u'doc3'), (28, 10)), ((u'doc3', u'doc6'), (10, 19)), ((u'doc3', u'doc4'), (10, 14)), ((u'doc3', u'doc5'), (10, 14)), ((u'doc1', u'doc6'), (28, 19)), ((u'doc1', u'doc4'), (28, 14)), ((u'doc1', u'doc5'), (28, 14)), ((u'doc4', u'do  
  
**>>> r19 = r18.map(lambda x:(x[0], (x[1][0]+x[1][1])))  
>>> r19.collect()**

[((u'doc2', u'doc3'), 26), ((u'doc2', u'doc6'), 35), ((u'doc2', u'doc4'), 30), ((u'doc2', u'doc5'), 30), ((u'doc1', u'doc2'), 44), ((u'doc1', u'doc3'), 38), ((u'doc3', u'doc6'), 29), ((u'doc3', u'doc4'), 24), ((u'doc3', u'doc5'), 24), ((u'doc1', u'doc6'), 47), ((u'doc1', u'doc4'), 42), ((u'doc1', u'doc5'), 42), ((u'doc4', u'doc6'), 33), ((u'doc4', u'doc5'), 28), ((u'doc5', u'doc6'), 33)]  
  
  
  
**>>> r20 = r13.join(r19)  
>>> r20.collect()**

**//Joins the intersection RDD and (d1+d2) RDD.**

[((u'doc2', u'doc4'), (9, 30)), ((u'doc3', u'doc5'), (4, 24)), ((u'doc2', u'doc6'), (4, 35)), ((u'doc1', u'doc3'), (6, 38)), ((u'doc4', u'doc6'), (3, 33)), ((u'doc1', u'doc5'), (8, 42)), ((u'doc5', u'doc6'), (3, 33)), ((u'doc3', u'doc4'), (6, 24)), ((u'doc1', u'doc2'), (9, 44)), ((u'doc2', u'doc3'), (5, 26)), ((u'doc4', u'doc5'), (6, 28)), ((u'doc3', u'doc6'), (6, 29)), ((u'doc1', u'doc4'), (5, 42)), ((u'doc1', u'doc6'), (13, 47)), ((u'doc2', u'doc5'), (10, 30))]

**>>> r21 = r20.map(lambda x:(x[0],(x[1][0],(x[1][1]-x[1][0]))))  
>>> r21.collect()**

**//(AUB) = A+B-(A Intersect B)**

[((u'doc2', u'doc4'), (9, 21)), ((u'doc3', u'doc5'), (4, 20)), ((u'doc2', u'doc6'), (4, 31)), ((u'doc1', u'doc3'), (6, 32)), ((u'doc4', u'doc6'), (3, 30)), ((u'doc1', u'doc5'), (8, 34)), ((u'doc5', u'doc6'), (3, 30)), ((u'doc3', u'doc4'), (6, 18)), ((u'doc1', u'doc2'), (9, 35)), ((u'doc2', u'doc3'), (5, 21)), ((u'doc4', u'doc5'), (6, 22)), ((u'doc3', u'doc6'), (6, 23)), ((u'doc1', u'doc4'), (5, 37)), ((u'doc1', u'doc6'), (13, 34)), ((u'doc2', u'doc5'), (10, 20))]  
  
  
**>>> r22 = r21.map(lambda x:(x[0],(x[1][0]\*100/x[1][1])))  
>>> r22.collect()**

**//Divide**

s[((u'doc2', u'doc4'), 42), ((u'doc3', u'doc5'), 20), ((u'doc2', u'doc6'), 12), ((u'doc1', u'doc3'), 18), ((u'doc4', u'doc6'), 10), ((u'doc1', u'doc5'), 23), ((u'doc5', u'doc6'), 10), ((u'doc3', u'doc4'), 33), ((u'doc1', u'doc2'), 25), ((u'doc2', u'doc3'), 23), ((u'doc4', u'doc5'), 27), ((u'doc3', u'doc6'), 26), ((u'doc1', u'doc4'), 13), ((u'doc1', u'doc6'), 38), ((u'doc2', u'doc5'), 50)]  
>>>

**>>> r22.sortByKey().collect()**[((u'doc1', u'doc2'), 25), ((u'doc1', u'doc3'), 18), ((u'doc1', u'doc4'), 13), ((u'doc1', u'doc5'), 23), ((u'doc1', u'doc6'), 38), ((u'doc2', u'doc3'), 23), ((u'doc2', u'doc4'), 42), ((u'doc2', u'doc5'), 50), ((u'doc2', u'doc6'), 12), ((u'doc3', u'doc4'), 33), ((u'doc3', u'doc5'), 20), ((u'doc3', u'doc6'), 26), ((u'doc4', u'doc5'), 27), ((u'doc4', u'doc6'), 10), ((u'doc5', u'doc6'), 10)]  
>>>

Output for n=3 (Or Trigram)

**>>> r22.sortByKey().collect()**[((u'doc1', u'doc2'), 16), ((u'doc1', u'doc3'), 15), ((u'doc1', u'doc4'), 6), ((u'doc1', u'doc5'), 17), ((u'doc1', u'doc6'), 35), ((u'doc2', u'doc3'), 10), ((u'doc2', u'doc4'), 33), ((u'doc2', u'doc5'), 47), ((u'doc2', u'doc6'), 3), ((u'doc3', u'doc4'), 18), ((u'doc3', u'doc5'), 11), ((u'doc3', u'doc6'), 20), ((u'doc4', u'doc5'), 21), ((u'doc4', u'doc6'), 3), ((u'doc5', u'doc6'), 3)]  
>>>