Table of Contents

MIDS - w261 Machine Learning At Scale

Course Lead: Dr James G. Shanahan (**email** Jimi via James.Shanahan *AT* gmail.com)

Assignment - HW5

Name: *Your Name Goes Here*
Class: MIDS w261 (Section *Your Section Goes Here*, e.g., Fall 2016 Group 1)
Email: *Your UC Berkeley Email Goes Here*@iSchool.Berkeley.edu
StudentId 26302327 **End of StudentId**
Week: 5

NOTE: please replace `1234567` with your student id above
Due Time: HW is due the Tuesday of the following week by 8AM (West coast time). I.e., Tuesday, Feb 14, 2017 in the case of this homework.

- **HW5 Phase 1** This can be done on a local machine (with a unit test on the cloud such as AltaScale's PaaS or on AWS) and is due Tuesday, Week 6 by 8AM (West coast time). It will primarily focus on building a unit/systems and for pairwise similarity calculations pipeline (for stripe documents)
- **HW5 Phase 2** This will require the Altiscale cluster and will be due Tuesday, Feb 21 by 8AM (West coast time). The focus of HW5 Phase 2 will be to scale up the unit/systems tests to the Google 5 gram corpus.

1 Instructions

MIDS UC Berkeley, Machine Learning at Scale
DATSCIW261 ASSIGNMENT #5

INSTRUCTIONS for SUBMISSIONS

Follow the instructions for submissions carefully.

Each student has a `HW-<user>` repository for all assignments.

Push the following to your HW github repo into the master branch:

- Your local HW5 directory. Your repo file structure should look like this:

```
HW-<user>
    --HW3
       |__MIDS-W261-HW-03-<Student_id>.ipynb
       |__MIDS-W261-HW-03-<Student_id>.pdf
       |__some other hw3 file
    --HW4
       |__MIDS-W261-HW-04-<Student_id>.ipynb
       |__MIDS-W261-HW-04-<Student_id>.pdf
       |__some other hw4 file
    etc..
```

HW Problems

HW5.0 data warehouse; star schema

- What is a data warehouse? What is a Star schema? When is it used?

We will understand this sytem from insurance industry example.

DW

Underwriting system book policies and collects premium from customers. They are transaction systems which gives current status of policies/booking system and also can give total active policies. system is good enough for underwriteres.

For acturial calculation (people who decides the price/rate premium) they would need historical records for individual customers, they will also need data for all claims ,market trend data and competitors data. So decision making people needs data from various sources not just single .

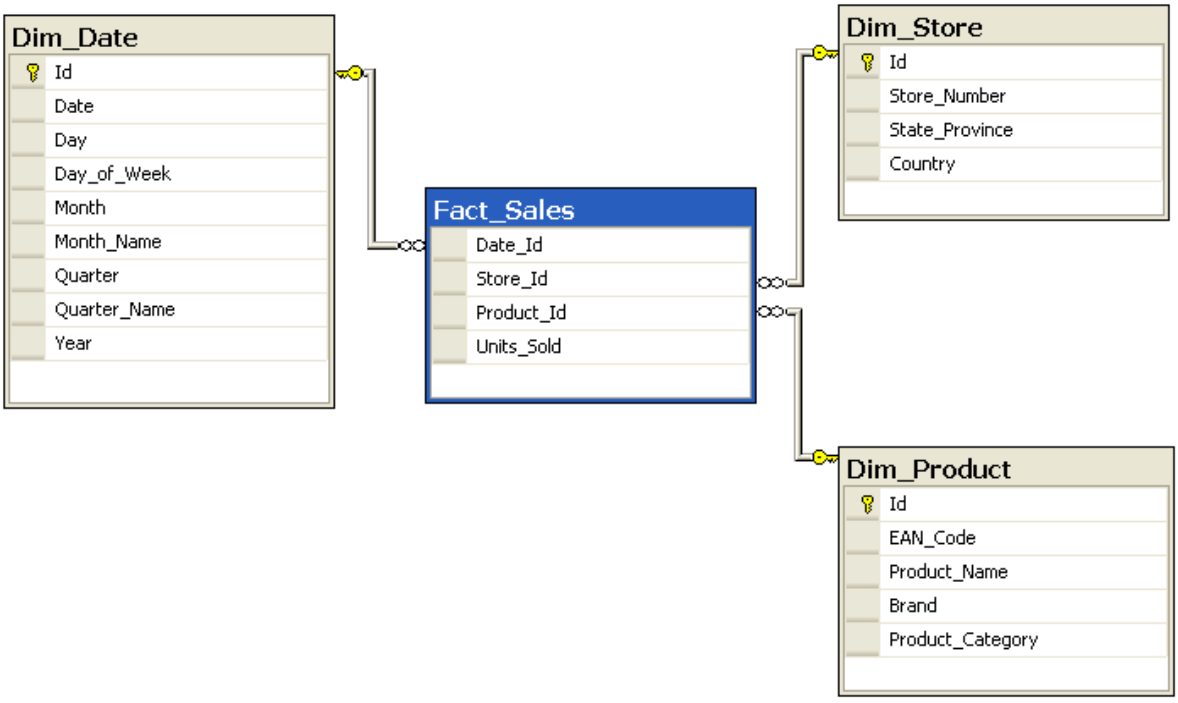
DW system solves this problem. DWs are central repositories of integrated data from one or more disparate sources. They store current and historical data in one single place and are used for creating analytical reports for knowledge workers throughout the enterprise.

Star Schema

Star schema is one of the architecture in DW systems. A single fact record surrounded by multiple dimension records forming a star pattern. The fact record type sits in the middle of the schema at the many end of a number of one-to-many relationships. The dimension records are used to query the fact record.

Related dimensions are grouped as columns in dimension tables, and the facts are stored as columns in a fact table. The star schema gets its name from its appearance: when drawn with the fact table in the center, it looks like a star or asterisk

SAP/Oracle DW system supports this architecture. for SAP BI sytems they are called as infocubes.



when it is used?

Querying Facts

One or more facts are requested, along with the dimensional attributes that provide the desired context. The facts will be summarized in accordance with the dimensions present in the query. Dimension values are also used to limit the scope of the query, serving as the basis for filters or constraints on the data to be fetched and aggregated.

Browsing Dimensions

Browsing is the act of exploring the data within a dimension.Here we interact with dimensional schema.

HW5.1 Databases: 3NF; denormalized

- In the database world What is 3NF? Does machine learning use data in 3NF? If so why?
- In what form does ML consume data?
- Why would one use log files that are denormalized?

In the database world What is 3NF? Does machine learning use data in 3NF? If so why?

Third normal form (3NF) is the third step in normalizing a database and it builds on the first and second normal forms, 1NF and 2NF.

3NF states that all column reference in referenced data that are not dependent on the primary key should be removed. Another way of putting this is that only foreign key columns should be used to reference another table, and no other columns from the parent table should exist in the referenced table.

ML tries to find out model that can be generalized for observed data of various kind. For example, to predict the premium rate for particular insurance product we need premium/claims/market values housing/health and data from variety of sources. Algorithm needs to study all features at same time. With 3NF, we will end up with reading thosands of tables to make single record while training time which is not feasible. Hence not ideal though technically we can make it possible by joining tables at runtime.

In what form does ML consume data?

ML consumes data in denormalized form i.e. in flattened file.

This also helps in hadoop kind of databases where we fetch records in blocks of size 128MB or someitmes gigs. We get all data for each observation for data fetch.

Why would one use log files that are denormalized?

for ML we need each attributes/demension for particular observation. So we will use denormalized log files for that

HW5.2 Memory-backed map-side

Using MRJob, implement a hashside join (memory-backed map-side) for left, right and inner joins. Use the following tables for this HW and join based on the country code (third column of the transactions table and the second column of the Countries table:

```
transactions.dat
Alice Bob|$10|US
Sam Sneed|$1|CA
Jon Sneed|$20|CA
Arnold Wesise|$400|UK
Henry Bob|$2|US
Yo Yo Ma|$2|CA
Jon York|$44|CA
Alex Ball|$5|UK
Jim Davis|$66|JA

Countries.dat
United States|US
Canada|CA
United Kingdom|UK
Italy|IT
```

Justify which table you chose as the Left table in this hashside join.

Please report the number of rows resulting from:

- (1) Left joining Table Left with Table Right
- (2) Right joining Table Left with Table Right
- (3) Inner joining Table Left with Table Right

Left Table

I am choosing transactions table on left side as for left join I will get records only in left table. If I am analyzing the transactions I would prefer to see all transactions though some of them might be missing some data elements.

Even SAP recommends to use left join in their training material for HANA Database. People rarely use right outer join in practice.

We could have implement this using small table as left as well. This would not make a difference though as we are doing all joins here. but if situation would have been only for inner join then smaller table to use as left would make sense.

In [1]:
```
%load_ext autoreload
%autoreload 2
```

In [2]:
```
%%writefile transactions.dat
Alice Bob|$10|US
Sam Sneed|$1|CA
Jon Sneed|$20|CA
Arnold Wesise|$400|UK
Henry Bob|$2|US
Yo Yo Ma|$2|CA
Jon York|$44|CA
Alex Ball|$5|UK
Jim Davis|$66|JA
```

```
Overwriting transactions.dat
```

In [3]:
```
%%writefile Countries.dat
United States|US
Canada|CA
United Kingdom|UK
Italy|IT
```

```
Overwriting Countries.dat
```

In [4]:
```
%%writefile MRjoins.py
#!/usr/bin/env python
#START STUDENT CODE43

from mrjob.job import MRJob
from mrjob.step import MRStep
import re




class MRjoins(MRJob):

#    OUTPUT_PROTOCOL = JSONValueProtocol
    SORT_VALUES = True

    #def __init__(self, *args, **kwargs):
    #    super(MRjoins, self).__init__(*args, **kwargs)
    def configure_options(self):
        super(MRjoins, self).configure_options()
        self.add_passthrough_option('--join-type', type = 'string', default = '
left')

    def mapper_init_leftjoin(self):
        self.country = {}
        with open('Countries.dat', 'r') as country:
            for line in country:
                tokens = line.split('|')
                # the entry is the page id => ( url : count )
                self.country[tokens[1].strip('\n')] = tokens[0]



    def mapper(self, _, line):
        self.increment_counter('Execution Counts', 'mapper calls', 1)

        fields = line.split("|")
        if fields[2].strip('\n') in self.country:
            sline = line + '|'+ self.country[fields[2].strip('\n')]
            yield None,sline

        else:
            yield None,line
    def mapper_innerjoin(self, _, line):
        self.increment_counter('Execution Counts', 'mapper calls', 1)

        fields = line.split("|")
        if fields[2].strip('\n') in self.country:
            sline = line + '|'+ self.country[fields[2].strip('\n')]
            yield None,sline

    def mapper_rightjoin(self, _, line):
        self.increment_counter('Execution Counts', 'mapper calls', 1)
        fields = line.split("|")
        keys =fields[2].strip('\n')
        if keys in self.country:
            sline = line + '|'+ self.country[keys]
            yield None,sline


    def reducer_init(self):
        self.country = {}
        with open('Countries.dat', 'r') as f:
            for line in f:
                tokens = line.split('|')
                # the entry is the page id => ( url : count )
                self.country[tokens[1].strip('\n')] = tokens[0]

    def reducer(self,key, records):
```

```
Overwriting MRjoins.py
```

In [5]:
```
input_folder = 'hdfs://' + '/user/root/tmp/mrjob' + '/' + 'hw522'
```

In [6]:
```
!hdfs dfs -rm -r $input_folder
!hdfs dfs -mkdir $input_folder

!hdfs dfs -copyFromLocal transactions.dat  $input_folder
```

```
Deleted hdfs:///user/root/tmp/mrjob/hw522
```

In [7]:
```
!hdfs dfs -ls $input_folder
```

```
Found 1 items
-rw-r--r--   1 root supergroup        151 2017-06-12 21:28 hdfs:///user/root/t
mp/mrjob/hw522/transactions.dat
```

In [8]:
```
from MRjoins import MRjoins
def do_joins(jointype,hadoop):
    if hadoop == 'X': #not in hadoop mode
        mr_job = MRjoins(args=['transactions.dat',"--file" ,"Countries.dat","--
join-type",jointype])
    else:
        mr_job = MRjoins(args=['-r','hadoop','transactions.dat',"--file" ,"Coun
tries.dat","--join-type",jointype])
    with mr_job.make_runner() as runner:
    # Run MRJob
        runner.run()

    # Write stream_output to file
        count = 0
        for line in runner.stream_output():
            key,value =  mr_job.parse_output_line(line)
            print value
            count += 1
    print "Total number of Records %d"%count
```

In [9]:
```
#Inner join in hadoop mode
do_joins("inner",'')
```

```
No handlers could be found for logger "mrjob.compat"

Yo Yo Ma|$2|CA|Canada
Jon York|$44|CA|Canada
Alex Ball|$5|UK|United Kingdom
Alice Bob|$10|US|United States
Sam Sneed|$1|CA|Canada
Jon Sneed|$20|CA|Canada
Arnold Wesise|$400|UK|United Kingdom
Henry Bob|$2|US|United States
Total number of Records 8
```

In [10]:
```
#lefjoin in hadoop mode
do_joins("left",'')
```

```
Yo Yo Ma|$2|CA|Canada
Jon York|$44|CA|Canada
Alex Ball|$5|UK|United Kingdom
Jim Davis|$66|JA
Alice Bob|$10|US|United States
Sam Sneed|$1|CA|Canada
Jon Sneed|$20|CA|Canada
Arnold Wesise|$400|UK|United Kingdom
Henry Bob|$2|US|United States
Total number of Records 9
```

In [11]:
```
#right in hadoop mode
do_joins("right",'')
```

```
[u'Alex Ball', u'$5', u'UK', u'United Kingdom']
[u'Alice Bob', u'$10', u'US', u'United States']
[u'Arnold Wesise', u'$400', u'UK', u'United Kingdom']
[u'Henry Bob', u'$2', u'US', u'United States']
[u'Jon Sneed', u'$20', u'CA', u'Canada']
[u'Jon York', u'$44', u'CA', u'Canada']
[u'Sam Sneed', u'$1', u'CA', u'Canada']
[u'Yo Yo Ma', u'$2', u'CA', u'Canada']
[None, None, u'IT', u'Italy']
Total number of Records 9
```

In [12]:
```
from MRjoins import MRjoins
#mr_job = MRjoins(args=['-r','hadoop','hdfs:///user/root/tmp/mrjob/hw522/transa
ctions.dat'])
mr_job = MRjoins(args=['transactions.dat',"--file" ,"Countries.dat","--join-typ
e","right"])

with mr_job.make_runner() as runner:
    # Run MRJob
    runner.run()

    # Write stream_output to file
    count = 0
    for line in runner.stream_output():
        key,value =  mr_job.parse_output_line(line)
        print value
        count += 1
    print "Total number of Records %d"%count
```

```
[u'Alex Ball', u'$5', u'UK', u'United Kingdom']
[u'Alice Bob', u'$10', u'US', u'United States']
[u'Arnold Wesise', u'$400', u'UK', u'United Kingdom']
[u'Henry Bob', u'$2', u'US', u'United States']
[u'Jon Sneed', u'$20', u'CA', u'Canada']
[u'Jon York', u'$44', u'CA', u'Canada']
[u'Sam Sneed', u'$1', u'CA', u'Canada']
[u'Yo Yo Ma', u'$2', u'CA', u'Canada']
[None, None, u'IT', u'Italy']
Total number of Records 9
```

HW5.2.1 (OPTIONAL) Almost stateless reducer-side join

The following MRJob code, implements a reduce-side join for an inner join. The reducer is almost stateless, i.e., uses as little memory as possible. Use the tables from HW5.2 for this HW and join based on the country code (third column of the transactions table and the second column of the Countries table perform. Perform an left, right, inner joins using the code provided below and report the number of rows resulting from:

- (1) Left joining Table Left with Table Right
- (2) Right joining Table Left with Table Right
- (3) Inner joining Table Left with Table Right

Again make smart decisions about which table should be the left table (i.e., crosscheck the code).

Some notes on the code Here, the mapper receives its set of input splits either from the transaction table or from the countries table and makes the appropriate transformations: splitting the line into fields, and emitting a key/value. The key is the join key - in this case, the country code field of both sets of records. The mapper knows which file and type of record it is receiving based on the length of the fields. The records it emits contain the join field as the key, which acts as the partitioning key; We use the SORT_VALUES option, which ensures the values are sorted as well. Then, we employ a trick to ensure that for each join key, country records are seen always before transaction records. We achieve this by adding an arbitrary key to the front of the value: 'A' for countries, 'B' for customers. This makes countries sort before customers for each and every join/partition key. After that trick, the join is simply a matter of storing countries ('A' records) and crossing this array with each customer record.

In [13]:
```
%%writefile reducerjoins.py
import sys, os, re
from mrjob.job import MRJob
from mrjob.step import MRStep


class MRJoin(MRJob):

  # Performs secondary sort
  #    OUTPUT_PROTOCOL = JSONValueProtocol
  SORT_VALUES = True

    #def __init__(self, *args, **kwargs):
    #    super(MRjoins, self).__init__(*args, **kwargs)
  def configure_options(self):
        super(MRJoin, self).configure_options()
        self.add_passthrough_option('--join-type', type = 'string', default = '
left')


  def mapper(self, _, line):
    splits = line.rstrip("\n").split("|")

    if len(splits) == 2: # country data
      symbol = 'A' # make country sort before transaction data
      country2digit = splits[1]
      yield country2digit, [symbol, splits]
    else: # person data
      symbol = 'B'
      country2digit = splits[2]
      yield country2digit, [symbol, splits]

  def reducer_inner(self, key, values):
    countries = {}
    for value in values:
      if value[0] == 'A':
        countries[value[1][1]] = value[1][0]
      if value[0] == 'B':
        if  key in countries:
           yield key, countries[key] + value[1][0] + value[1][1] + value[1][2]

  def reducer_left(self, key, values):
    countries = {} # should come first, as they are sorted on artificia key 'A'
    for value in values:
      if value[0] == 'A':
        countries[value[1][1]] = value[1][0]
      if value[0] == 'B':
        if  key in countries:
           yield key, countries[key] + value[1][0] + value[1][1] + value[1][2]
        else:
            yield key,"None"+ value[1][0] + value[1][1] + value[1][2]

  def reducer_init(self):
     self.countries = {}  #this is used in final method to do filtering on righ
t join
     self.passed_countries = set()
  def reducer_right(self, key, values):

    for value in values:
      if value[0] == 'A':
        self.countries[value[1][1]] = value[1][0]
      if value[0] == 'B':
        if  key in self.countries:
           yield key, self.countries[key] + value[1][0] + value[1][1] + value[1
][2]
           self.passed_countries.add(key)
  def reducer_right_final(self):
    for element in self.countries:
            if element not in self.passed_countries:
```

```
Overwriting reducerjoins.py
```

In [14]:
```
from reducerjoins import MRJoin
def right_run(jointype):


    mr_job = MRJoin(args=['transactions.dat',"Countries.dat","--join-type",join
type])

    with mr_job.make_runner() as runner:
    # Run MRJob
        runner.run()

    # Write stream_output to file
        count = 0
        for line in runner.stream_output():
            key,value =  mr_job.parse_output_line(line)
            print value
            count += 1
        print "Total number of Records %d"%count
```

In [15]:
```
right_run("left")
```

```
CanadaJon Sneed$20CA
CanadaJon York$44CA
CanadaSam Sneed$1CA
CanadaYo Yo Ma$2CA
NoneJim Davis$66JA
United KingdomAlex Ball$5UK
United KingdomArnold Wesise$400UK
United StatesAlice Bob$10US
United StatesHenry Bob$2US
Total number of Records 9
```

In [16]:
```
right_run("right")
```

```
CanadaJon Sneed$20CA
CanadaJon York$44CA
CanadaSam Sneed$1CA
CanadaYo Yo Ma$2CA
United KingdomAlex Ball$5UK
United KingdomArnold Wesise$400UK
United StatesAlice Bob$10US
United StatesHenry Bob$2US
ItalyNoneNone
Total number of Records 9
```

In [17]:
```
right_run("inner")
```

```
CanadaJon Sneed$20CA
CanadaJon York$44CA
CanadaSam Sneed$1CA
CanadaYo Yo Ma$2CA
United KingdomAlex Ball$5UK
United KingdomArnold Wesise$400UK
United StatesAlice Bob$10US
United StatesHenry Bob$2US
Total number of Records 8
```

5.3 Pairwise similarity - PHASE 1

In this part of the assignment we will focus on developing methods for detecting synonyms, using the Google 5-grams dataset. To accomplish this you must script two main tasks using MRJob:

(1) Using the systems tests data sets, write mrjob code to build the stripes

(2) Write mrjob code to build an inverted index from the stripes

(3) Using two (symmetric) comparison methods of your choice (e.g., correlations, distances, similarities), pairwise compare all stripes (vectors), and output to a file.

==Design notes for (1)==
For this task you will be able to modify the pattern we used in HW 3.2 (feel free to use the solution as reference). To total the word counts across the n-grams, output the support from the mappers using the total order inversion pattern:

<*word,count>

to ensure that the support arrives before the cooccurrences.

In addition to ensuring the determination of the total word counts, the mapper must also output co-occurrence counts for the pairs of words inside of each n-gram. Treat these words as a basket, as we have in HW 3, but count all stripes or pairs in both orders, i.e., count both orderings: (word1,word2), and (word2,word1), to preserve symmetry in our output for (2).

==Design notes for (3)==
For this task you will have to determine a method of comparison. Here are a few that you might consider:

- Jaccard
- Cosine similarity
- Spearman correlation
- Euclidean distance
- Taxicab (Manhattan) distance
- Shortest path graph distance (a graph, because our data is symmetric!)
- Pearson correlation
- Kendall correlation ...

However, be cautioned that some comparison methods are more difficult to parallelize than others, and do not perform more associations than is necessary, since your choice of association will be symmetric.

Please use the inverted index (discussed in live session #5) based pattern to compute the pairwise (term-by-term) similarity matrix.

In [18]:
```
%%writefile buildStripes.py
#!~/anaconda2/bin/python
# -*- coding: utf-8 -*-

from __future__ import division
import re
import mrjob
import json
from mrjob.protocol import RawProtocol
from mrjob.job import MRJob
from mrjob.step import MRStep
from itertools import  combinations
class MRbuildStripes(MRJob):

  #START SUDENT CODE531_STRIPES
    SORT_VALUES = True

    #def __init__(self, *args, **kwargs):
    #    super(MRjoins, self).__init__(*args, **kwargs)




    def mapper(self, _, recs):
        self.increment_counter('Execution Counts', 'mapper calls', 1)
        fields = recs.split("\t")

        products = fields[0].lower().replace('\n','').split()
        for i, term in enumerate(products):
                # Create a new stripe for each term
                stripe = {}

                for j, token in enumerate(products):
                    # Don't count the term's co-occurrence with itself
                    if i != j:
                        x = stripe.get(token,None)
                        if x == None:
                            stripe[token] = int( fields[1])
                        else:
                            stripe[token] += int(fields[1])

                # Emit the term and the stripe
                yield term, stripe

    def combiner(self, word, stripes):
        yield word, self.combine_stripes(stripes)

    def combine_stripes(self, stripes):
        combined_stripe = {}

        for stripe in stripes:
            for key, value in stripe.iteritems():
                if key in combined_stripe:
                    combined_stripe[key] += int(value)
                else:
                    combined_stripe[key] = int(value)

        return combined_stripe
    def reducer(self,key, records):
        yield key, self.combine_stripes(records)

    def steps(self):  #pipeline of Map-Reduce jobs
        step = MRStep(
                    mapper=self.mapper,       # STEP 1: word count step
                    combiner = self.combiner,
                    reducer=self.reducer
                    )
        return [step]
```

```
Overwriting buildStripes.py
```

In [19]:
```
from buildStripes import MRbuildStripes
#mr_job = MRjoins(args=['-r','hadoop','hdfs:///user/root/tmp/mrjob/hw522/transa
ctions.dat'])
mr_job = MRbuildStripes(args=['googlebooks-eng-all-5gram-20090715-0-filtered-fi
rst-10-lines.txt'])

with mr_job.make_runner() as runner:
    # Run MRJob
    runner.run()

    # Write stream_output to file
    count = 0
    for line in runner.stream_output():
        key,value =  mr_job.parse_output_line(line)
        print key,value
        count += 1
    print "Total number of Records %d"%count
```

```
a {u'limited': 55, u'female': 447, u'general': 92, u'sea': 62, u'in': 1201, u'
religious': 59, u'george': 92, u'biography': 92, u'city': 62, u'for': 59, u'ta
les': 123, u'government': 102, u'the': 124, u'forms': 116, u'wales': 1099, u'c
hristmas': 1099, u"child's": 1099, u'collection': 239, u'by': 62, u'case': 604
, u'circumstantial': 62, u'of': 1011, u'study': 604, u'bill': 59, u'establishi
ng': 59, u'narrative': 62, u'fairy': 123}
bill {u'a': 59, u'religious': 59, u'for': 59, u'establishing': 59}
biography {u'a': 92, u'of': 92, u'george': 92, u'general': 92}
by {u'a': 62, u'city': 62, u'the': 62, u'sea': 62}
case {u'a': 604, u'limited': 55, u'government': 102, u'of': 502, u'study': 604
, u'female': 447, u'in': 102}
child's {u'a': 1099, u'wales': 1099, u'christmas': 1099, u'in': 1099}
christmas {u'a': 1099, u'wales': 1099, u"child's": 1099, u'in': 1099}
circumstantial {u'a': 62, u'of': 62, u'the': 62, u'narrative': 62}
city {u'a': 62, u'the': 62, u'by': 62, u'sea': 62}
collection {u'a': 239, u'of': 355, u'fairy': 123, u'tales': 123, u'forms': 116
}
establishing {u'a': 59, u'bill': 59, u'religious': 59, u'for': 59}
fairy {u'a': 123, u'of': 123, u'tales': 123, u'collection': 123}
female {u'a': 447, u'case': 447, u'study': 447, u'of': 447}
for {u'a': 59, u'bill': 59, u'religious': 59, u'establishing': 59}
forms {u'a': 116, u'of': 232, u'collection': 116}
general {u'a': 92, u'of': 92, u'george': 92, u'biography': 92}
george {u'a': 92, u'of': 92, u'biography': 92, u'general': 92}
government {u'a': 102, u'case': 102, u'study': 102, u'in': 102}
in {u'a': 1201, u'case': 102, u"child's": 1099, u'study': 102, u'government':
102, u'wales': 1099, u'christmas': 1099}
limited {u'a': 55, u'case': 55, u'study': 55, u'of': 55}
narrative {u'a': 62, u'of': 62, u'the': 62, u'circumstantial': 62}
of {u'a': 1011, u'case': 502, u'circumstantial': 62, u'limited': 55, u'of': 23
2, u'tales': 123, u'collection': 355, u'general': 92, u'forms': 232, u'female'
: 447, u'narrative': 62, u'study': 502, u'fairy': 123, u'the': 62, u'george':
92, u'biography': 92}
religious {u'a': 59, u'bill': 59, u'for': 59, u'establishing': 59}
sea {u'a': 62, u'city': 62, u'the': 62, u'by': 62}
study {u'a': 604, u'case': 604, u'limited': 55, u'government': 102, u'of': 502
, u'female': 447, u'in': 102}
tales {u'a': 123, u'of': 123, u'fairy': 123, u'collection': 123}
the {u'a': 124, u'city': 62, u'circumstantial': 62, u'of': 62, u'sea': 62, u'n
arrative': 62, u'by': 62}
wales {u'a': 1099, u"child's": 1099, u'christmas': 1099, u'in': 1099}
Total number of Records 28
```

In [20]:
```
%%writefile invertedIndex.py
#!~/anaconda2/bin/python
# -*- coding: utf-8 -*-


from __future__ import division
import collections
import re
import json
import math
import numpy as np
import itertools
import mrjob
from mrjob.protocol import RawProtocol
from mrjob.job import MRJob
from mrjob.step import MRStep
from mrjob.protocol import JSONProtocol
class MRinvertedIndex(MRJob):
    INPUT_PROTOCOL = JSONProtocol
    SORT_VALUES = True
#START SUDENT CODE531_INV_INDEX
    def mapper_normalize_transpose(self, word, rate_stripe):

        # First compute the magnitude for the vector.

        #magnitude = math.sqrt(sum([value ** 2 for value in rate_stripe.interval
ues()]))

        # Divide each value in the vector by the magnitude to normalize.
        length = len(rate_stripe)
        for key, value in rate_stripe.iteritems():
            #normalized_value = value / magnitude
            yield key, { word: length}
    def combiner_normalize_transpose(self, word, transpose_stripes):
        yield word, self.combine_stripes(transpose_stripes)
    def reducer_normalize_transpose(self, word, transpose_stripes):
        yield word, self.combine_stripes(transpose_stripes)
    def combine_stripes(self, stripes):
        combined_stripe = {}

        for stripe in stripes:
            for key, value in stripe.iteritems():
                if key in combined_stripe:
                    combined_stripe[key] += value
                else:
                    combined_stripe[key] = value

        return combined_stripe


    def steps(self):

        transpose_step = MRStep(
            mapper = self.mapper_normalize_transpose,
            combiner = self.combiner_normalize_transpose,
            reducer = self.reducer_normalize_transpose)
        return [transpose_step]

#END SUDENT CODE531_INV_INDEX

if __name__ == '__main__':
    MRinvertedIndex.run()
```

```
Overwriting invertedIndex.py
```

In [21]:
```
from invertedIndex import MRinvertedIndex
#mr_job = MRjoins(args=['-r','hadoop','hdfs:///user/root/tmp/mrjob/hw522/transa
ctions.dat'])
mr_job = MRinvertedIndex(args=['systems_test_stripes_3'])

with mr_job.make_runner() as runner:
    # Run MRJob
    runner.run()

    # Write stream_output to file
    count = 0
    for line in runner.stream_output():
        key,value =  mr_job.parse_output_line(line)
        print key,value
        count += 1
    print "Total number of Records %d"%count
```

```
M {u'DocC': 4}
N {u'DocC': 4}
X {u'DocB': 2, u'DocA': 3}
Y {u'DocB': 2, u'DocC': 4, u'DocA': 3}
Z {u'DocC': 4, u'DocA': 3}
Total number of Records 5
```

In [22]:
```
%%writefile similarity.py
#!~/anaconda2/bin/python
# -*- coding: utf-8 -*-

from __future__ import division
import collections
import re
import json
import math
import numpy as np
import itertools
import mrjob
from mrjob.protocol import RawProtocol
from mrjob.job import MRJob
from mrjob.step import MRStep
from mrjob.protocol import JSONProtocol

class MRsimilarity(MRJob):
    INPUT_PROTOCOL = JSONProtocol
    SORT_VALUES = True

#START SUDENT CODE531_SIMILARITY
    def mapper_jaccard(self, word, rate_stripe):
        #get all words and lengths
#We will emit stripes for each word vector here.  These stripes will
#be used in combiner to find the common length i.e. words occuring together
        nonzero_keys = [key for key, value in rate_stripe.iteritems() if value
!= 0]



        sorted_keys = sorted(nonzero_keys)
        # N * N  complexity matrix calculation
        #We are going over each record to find out the common occureances
        for i in range(0, len(sorted_keys)):
            left_label = sorted_keys[i]

            stripe = {}

            for j in range(i + 1, len(sorted_keys)):
                right_label = sorted_keys[j]
                stripe[right_label] = 1

            yield left_label, stripe



        for key in sorted_keys:
            yield '*',{key:1}
            #{u'DocC': 1}
        #yield '*', { key: 1 for key in sorted_keys }
    def combiner_jaccard(self, left_label, partial_stripes):
        yield left_label, self.combine_stripes(partial_stripes)

#find out the jaccard values.
    def reducer_jaccard(self, left_label, partial_stripes):
        total_stripe = self.combine_stripes(partial_stripes)
        #this stores the total length of each word Vector
        if left_label == '*':
            self.total_counts = total_stripe
            return

        for right_label, intersection_size in total_stripe.iteritems():
            coordinate = (left_label, right_label)
            union_size = self.total_counts[left_label] + self.total_counts[righ
t_label]


            jaccard_distance = float(intersection_size)/float(union_size - inte
```

```
Overwriting similarity.py
```

In [23]:
```
from invertedIndex import MRinvertedIndex
#mr_job = MRjoins(args=['-r','hadoop','hdfs:///user/root/tmp/mrjob/hw522/transa
ctions.dat'])
mr_job = MRinvertedIndex(args=['systems_test_stripes_3'])

with mr_job.make_runner() as runner:
    # Run MRJob
    runner.run()

    # Write stream_output to file
    count = 0
    for line in runner.stream_output():
        key,value =  mr_job.parse_output_line(line)
        print key,value
        count += 1
    print "Total number of Records %d"%count
```

```
M {u'DocC': 4}
N {u'DocC': 4}
X {u'DocB': 2, u'DocA': 3}
Y {u'DocB': 2, u'DocC': 4, u'DocA': 3}
Z {u'DocC': 4, u'DocA': 3}
Total number of Records 5
```

In [24]:
```
from similarity import MRsimilarity
#mr_job = MRjoins(args=['-r','hadoop','hdfs:///user/root/tmp/mrjob/hw522/transa
ctions.dat'])
mr_job = MRsimilarity(args=['systems_test_index_3'])

with mr_job.make_runner() as runner:
    # Run MRJob
    runner.run()

    # Write stream_output to file
    count = 0
    for line in runner.stream_output():
        key,value =  mr_job.parse_output_line(line)
        print key,value
        count += 1
    print "Total number of Records %d"%count
```

```
[u'DocA', u'DocB'] {u'dice': 0.8, u'jaccard': 0.6666666667000001}
[u'DocA', u'DocC'] {u'dice': 0.5714285714, u'jaccard': 0.4}
[u'DocB', u'DocC'] {u'dice': 0.33333333330000003, u'jaccard': 0.2}
Total number of Records 3
```

HW5.3.1 Run Systems tests locally on small datasets (PHASE1)

Complete 5.3 and systems test using the below test datasets. Phase 2 will focus on the entire Ngram dataset.

To help you through these tasks please verify that your code gives the results below (for stripes, inverted index, and pairwise similarities).

Test datasets:

- googlebooks-eng-all-5gram-20090715-0-filtered.txt [see below]
- atlas-boon-test [see below]
- stripe-docs-test [see below]

A large subset of the Google n-grams dataset

https://aws.amazon.com/datasets/google-books-ngrams/ (https://aws.amazon.com/datasets/google-books-ngrams/)

which we have placed in a bucket/folder on Dropbox and on s3:

https://www.dropbox.com/sh/tmqpc4o0xswhkvz/AACUifrl6wrMrlK6a3X3lZ9Ea?dl=0 (https://www.dropbox.com/sh/tmqpc4o0xswhkvz/AACUifrl6wrMrlK6a3X3lZ9Ea?dl=0)

s3://filtered-5grams/

In particular, this bucket contains (~200) files (10Meg each) in the format:

```
(ngram) \t (count) \t (pages_count) \t (books_count)
```

The next cell shows the first 10 lines of the googlebooks-eng-all-5gram-20090715-0-filtered.txt file.

DISCLAIMER: Each record is already a 5-gram. In real life, we would calculate the stripes cooccurrence data from the raw text by windowing over the raw text and not from the 5-gram preprocessed data (as we are doing here). Calculatating pairs on this 5-gram is a little corrupt as we will be double counting cooccurences. Having said that this exercise can still pull out some simialr terms.

1: unit/systems first-10-lines

In [25]:
```
%%writefile googlebooks-eng-all-5gram-20090715-0-filtered-first-10-lines.txt
A BILL FOR ESTABLISHING RELIGIOUS       59      59      54
A Biography of General George   92      90      74
A Case Study in Government      102     102     78
A Case Study of Female  447     447     327
A Case Study of Limited 55      55      43
A Child's Christmas in Wales    1099    1061    866
A Circumstantial Narrative of the       62      62      50
A City by the Sea       62      60      49
A Collection of Fairy Tales     123     117     80
A Collection of Forms of        116     103     82
```

```
Overwriting googlebooks-eng-all-5gram-20090715-0-filtered-first-10-lines.txt
```

2: unit/systems atlas-boon

In [26]:
```
%%writefile atlas-boon-systems-test.txt
atlas boon      50      50      50
boon cava dipped        10      10      10
atlas dipped    15      15      15
```

```
Overwriting atlas-boon-systems-test.txt
```

3: unit/systems stripe-docs-test

Three terms, A,B,C and their corresponding stripe-docs of co-occurring terms

- DocA {X:20, Y:30, Z:5}
- DocB {X:100, Y:20}
- DocC {M:5, N:20, Z:5}

(1) build stripes for all the test data sets - run the commands and insure that your output matches the output below

In [27]:
```
##############################################################################
# Make Stripes from ngrams for systems test 1
##############################################################################

#!hdfs dfs rm --recursive systems_test_stripes_1
!python buildStripes.py -r local googlebooks-eng-all-5gram-20090715-0-filtered-first-10-lines.txt > systems_test_stripes_1
```

```
No configs found; falling back on auto-configuration
ignoring partitioner keyword arg (requires real Hadoop): 'org.apache.hadoop.mapred.lib.KeyFieldBasedPartitioner'
Creating temp directory /tmp/buildStripes.root.20170612.213235.707741
Running step 1 of 1...
Counters: 1
        Execution Counts
                mapper calls=10
Counters: 1
        Execution Counts
                mapper calls=10
Streaming final output from /tmp/buildStripes.root.20170612.213235.707741/output...
Removing temp directory /tmp/buildStripes.root.20170612.213235.707741...
```

In [28]:
```
!cat systems_test_stripes_1
```

```
"a"     {"limited":55,"sea":62,"general":92,"female":447,"in":1201,"religious"
:59,"george":92,"biography":92,"city":62,"for":59,"tales":123,"child's":1099,"
forms":116,"wales":1099,"christmas":1099,"government":102,"collection":239,"by
":62,"case":604,"circumstantial":62,"fairy":123,"of":1011,"study":604,"bill":5
9,"establishing":59,"narrative":62,"the":124}
"bill"  {"a":59,"religious":59,"for":59,"establishing":59}
"biography"     {"a":92,"of":92,"george":92,"general":92}
"by"    {"a":62,"city":62,"the":62,"sea":62}
"case"  {"a":604,"limited":55,"government":102,"of":502,"study":604,"female":4
47,"in":102}
"child's"       {"a":1099,"wales":1099,"christmas":1099,"in":1099}
"christmas"     {"a":1099,"wales":1099,"in":1099,"child's":1099}
"circumstantial"        {"a":62,"of":62,"the":62,"narrative":62}
"city"  {"a":62,"the":62,"by":62,"sea":62}
"collection"    {"a":239,"of":355,"fairy":123,"tales":123,"forms":116}
"establishing"  {"a":59,"bill":59,"religious":59,"for":59}
"fairy" {"a":123,"of":123,"tales":123,"collection":123}
"female"        {"a":447,"case":447,"study":447,"of":447}
"for"   {"a":59,"bill":59,"religious":59,"establishing":59}
"forms" {"a":116,"of":232,"collection":116}
"general"       {"a":92,"of":92,"george":92,"biography":92}
"george"        {"a":92,"of":92,"biography":92,"general":92}
"government"    {"a":102,"case":102,"study":102,"in":102}
"in"    {"a":1201,"case":102,"government":102,"study":102,"child's":1099,"wale
s":1099,"christmas":1099}
"limited"       {"a":55,"case":55,"study":55,"of":55}
"narrative"     {"a":62,"of":62,"the":62,"circumstantial":62}
"of"    {"a":1011,"case":502,"circumstantial":62,"george":92,"limited":55,"of"
:232,"tales":123,"collection":355,"general":92,"forms":232,"female":447,"narra
tive":62,"study":502,"fairy":123,"the":62,"biography":92}
"religious"     {"a":59,"bill":59,"for":59,"establishing":59}
"sea"   {"a":62,"city":62,"the":62,"by":62}
"study" {"a":604,"case":604,"limited":55,"government":102,"of":502,"female":44
7,"in":102}
"tales" {"a":123,"of":123,"fairy":123,"collection":123}
"the"   {"a":124,"city":62,"circumstantial":62,"of":62,"sea":62,"narrative":62
,"by":62}
"wales" {"a":1099,"in":1099,"christmas":1099,"child's":1099}
```

In [29]:
```
!cat systems_test_stripes_1
```

```
"a"     {"limited":55,"sea":62,"general":92,"female":447,"in":1201,"religious"
:59,"george":92,"biography":92,"city":62,"for":59,"tales":123,"child's":1099,"
forms":116,"wales":1099,"christmas":1099,"government":102,"collection":239,"by
":62,"case":604,"circumstantial":62,"fairy":123,"of":1011,"study":604,"bill":5
9,"establishing":59,"narrative":62,"the":124}
"bill"  {"a":59,"religious":59,"for":59,"establishing":59}
"biography"     {"a":92,"of":92,"george":92,"general":92}
"by"    {"a":62,"city":62,"the":62,"sea":62}
"case"  {"a":604,"limited":55,"government":102,"of":502,"study":604,"female":4
47,"in":102}
"child's"       {"a":1099,"wales":1099,"christmas":1099,"in":1099}
"christmas"     {"a":1099,"wales":1099,"in":1099,"child's":1099}
"circumstantial"        {"a":62,"of":62,"the":62,"narrative":62}
"city"  {"a":62,"the":62,"by":62,"sea":62}
"collection"    {"a":239,"of":355,"fairy":123,"tales":123,"forms":116}
"establishing"  {"a":59,"bill":59,"religious":59,"for":59}
"fairy" {"a":123,"of":123,"tales":123,"collection":123}
"female"        {"a":447,"case":447,"study":447,"of":447}
"for"   {"a":59,"bill":59,"religious":59,"establishing":59}
"forms" {"a":116,"of":232,"collection":116}
"general"       {"a":92,"of":92,"george":92,"biography":92}
"george"        {"a":92,"of":92,"biography":92,"general":92}
"government"    {"a":102,"case":102,"study":102,"in":102}
"in"    {"a":1201,"case":102,"government":102,"study":102,"child's":1099,"wale
s":1099,"christmas":1099}
"limited"       {"a":55,"case":55,"study":55,"of":55}
"narrative"     {"a":62,"of":62,"the":62,"circumstantial":62}
"of"    {"a":1011,"case":502,"circumstantial":62,"george":92,"limited":55,"of"
:232,"tales":123,"collection":355,"general":92,"forms":232,"female":447,"narra
tive":62,"study":502,"fairy":123,"the":62,"biography":92}
"religious"     {"a":59,"bill":59,"for":59,"establishing":59}
"sea"   {"a":62,"city":62,"the":62,"by":62}
"study" {"a":604,"case":604,"limited":55,"government":102,"of":502,"female":44
7,"in":102}
"tales" {"a":123,"of":123,"fairy":123,"collection":123}
"the"   {"a":124,"city":62,"circumstantial":62,"of":62,"sea":62,"narrative":62
,"by":62}
"wales" {"a":1099,"in":1099,"christmas":1099,"child's":1099}
```

"a" {"limited": 55, "sea": 62, "general": 92, "female": 447, "in": 1201, "religi
ous": 59, "george": 92, "biography": 92, "city": 62, "for": 59, "tales": 123, "chil
d's": 1099, "forms": 116, "wales": 1099, "christmas": 1099, "government": 102, "col
lection": 239, "by": 62, "case": 604, "circumstantial": 62, "fairy": 123, "of": 101
1, "study": 604, "bill": 59, "establishing": 59, "narrative": 62, "the": 124}
"bill" {"a": 59, "religious": 59, "for": 59, "establishing": 59}
"biography" {"a": 92, "of": 92, "george": 92, "general": 92}
"by" {"a": 62, "city": 62, "the": 62, "sea": 62}
"case" {"a": 604, "limited": 55, "government": 102, "of": 502, "study": 604, "fe
male": 447, "in": 102}
"child's" {"a": 1099, "wales": 1099, "christmas": 1099, "in": 1099}
"christmas" {"a": 1099, "wales": 1099, "in": 1099, "child's": 1099}
"circumstantial" {"a": 62, "of": 62, "the": 62, "narrative": 62}
"city" {"a": 62, "the": 62, "by": 62, "sea": 62}
"collection" {"a": 239, "of": 355, "fairy": 123, "tales": 123, "forms": 116}
"establishing" {"a": 59, "bill": 59, "religious": 59, "for": 59}
"fairy" {"a": 123, "of": 123, "tales": 123, "collection": 123}
"female" {"a": 447, "case": 447, "study": 447, "of": 447}
"for" {"a": 59, "bill": 59, "religious": 59, "establishing": 59}
"forms" {"a": 116, "of": 232, "collection": 116}
"general" {"a": 92, "of": 92, "george": 92, "biography": 92}
"george" {"a": 92, "of": 92, "biography": 92, "general": 92}
"government" {"a": 102, "case": 102, "study": 102, "in": 102}
"in" {"a": 1201, "case": 102, "government": 102, "study": 102, "child's": 1099,
"wales": 1099, "christmas": 1099}
"limited" {"a": 55, "case": 55, "study": 55, "of": 55}
"narrative" {"a": 62, "of": 62, "the": 62, "circumstantial": 62}
"of" {"a": 1127, "case": 502, "circumstantial": 62, "george": 92, "limited": 55,
"tales": 123, "collection": 471, "general": 92, "forms": 348, "female": 447, "narra
tive": 62, "study": 502, "fairy": 123, "the": 62, "biography": 92}
"religious" {"a": 59, "bill": 59, "for": 59, "establishing": 59}
"sea" {"a": 62, "city": 62, "the": 62, "by": 62}
"study" {"a": 604, "case": 604, "limited": 55, "government": 102, "of": 502, "fe
male": 447, "in": 102}
"tales" {"a": 123, "of": 123, "fairy": 123, "collection": 123}
"the" {"a": 124, "city": 62, "circumstantial": 62, "of": 62, "sea": 62, "narrati
ve": 62, "by": 62}
"wales" {"a": 1099, "in": 1099, "christmas": 1099, "child's": 1099}

In [30]:
```
##############################################################################
# Make Stripes from ngrams for systems test 2
##############################################################################

!hdfs dfs rm --recursive systems_test_stripes_2
!python buildStripes.py -r local atlas-boon-systems-test.txt > systems_test_str
ipes_2
```

```
rm: Unknown command
Did you mean -rm?  This command begins with a dash.
No configs found; falling back on auto-configuration
ignoring partitioner keyword arg (requires real Hadoop): 'org.apache.hadoop.ma
pred.lib.KeyFieldBasedPartitioner'
Creating temp directory /tmp/buildStripes.root.20170612.213240.421389
Running step 1 of 1...
Counters: 1
        Execution Counts
                mapper calls=3
Counters: 1
        Execution Counts
                mapper calls=3
Streaming final output from /tmp/buildStripes.root.20170612.213240.421389/outp
ut...
Removing temp directory /tmp/buildStripes.root.20170612.213240.421389...
```

In [31]:
```
!cat systems_test_stripes_2
```

```
"atlas" {"dipped":15,"boon":50}
"boon"  {"atlas":50,"dipped":10,"cava":10}
"cava"  {"dipped":10,"boon":10}
"dipped"        {"atlas":15,"boon":10,"cava":10}
```

In [32]:
```
!cat systems_test_stripes_2
```

```
"atlas" {"dipped":15,"boon":50}
"boon"  {"atlas":50,"dipped":10,"cava":10}
"cava"  {"dipped":10,"boon":10}
"dipped"        {"atlas":15,"boon":10,"cava":10}
```

```
"atlas"   {"dipped": 15, "boon": 50}
"boon"    {"atlas": 50, "dipped": 10, "cava": 10}
"cava"    {"dipped": 10, "boon": 10}
"dipped"  {"atlas": 15, "boon": 10, "cava": 10}
```

In [33]:
```
##############################################################################
# Stripes for systems test 3 (given, no need to build stripes)
##############################################################################

with open("systems_test_stripes_3", "w") as f:
    f.writelines([
        '"DocA"\t{"X":20, "Y":30, "Z":5}\n',
        '"DocB"\t{"X":100, "Y":20}\n',
        '"DocC"\t{"M":5, "N":20, "Z":5, "Y":1}\n'
    ])
!cat systems_test_stripes_3
```

```
"DocA"  {"X":20, "Y":30, "Z":5}
"DocB"  {"X":100, "Y":20}
"DocC"  {"M":5, "N":20, "Z":5, "Y":1}
```

(2) Build Inverted Index - run the commands and insure that your output matches the output below

In [34]:
```
!python invertedIndex.py -r local systems_test_stripes_1 > systems_test_index_1
```

```
No configs found; falling back on auto-configuration
ignoring partitioner keyword arg (requires real Hadoop): 'org.apache.hadoop.ma
pred.lib.KeyFieldBasedPartitioner'
Creating temp directory /tmp/invertedIndex.root.20170612.213247.264113
Running step 1 of 1...
Streaming final output from /tmp/invertedIndex.root.20170612.213247.264113/out
put...
Removing temp directory /tmp/invertedIndex.root.20170612.213247.264113...
```

In [35]:
```
!python invertedIndex.py -r local systems_test_stripes_2 > systems_test_index_2
```

```
No configs found; falling back on auto-configuration
ignoring partitioner keyword arg (requires real Hadoop): 'org.apache.hadoop.ma
pred.lib.KeyFieldBasedPartitioner'
Creating temp directory /tmp/invertedIndex.root.20170612.213250.924138
Running step 1 of 1...
Streaming final output from /tmp/invertedIndex.root.20170612.213250.924138/out
put...
Removing temp directory /tmp/invertedIndex.root.20170612.213250.924138...
```

In [36]:
```
!python invertedIndex.py -r local systems_test_stripes_3 > systems_test_index_3
```

```
No configs found; falling back on auto-configuration
ignoring partitioner keyword arg (requires real Hadoop): 'org.apache.hadoop.ma
pred.lib.KeyFieldBasedPartitioner'
Creating temp directory /tmp/invertedIndex.root.20170612.213254.206656
Running step 1 of 1...
Streaming final output from /tmp/invertedIndex.root.20170612.213254.206656/out
put...
Removing temp directory /tmp/invertedIndex.root.20170612.213254.206656...
```

In [37]:
```
##############################################################
# Pretty print systems tests for generating Inverted Index
##############################################################

import json

for i in range(1,4):
    print "-"*100
    print "Systems test ",i," - Inverted Index"
    print "-"*100
    with open("systems_test_index_"+str(i),"r") as f:
        lines = sorted(f.readlines())
        for line in lines:
            line = line.strip()
            word, doc_list = line.split("\t")
            doc_dict = json.loads(doc_list)
            stripe=[]
            for doc in doc_dict:
                stripe.append([doc, doc_dict[doc]])
            stripe=sorted(stripe)
            stripe.extend([["",""] for _ in xrange(3 - len(stripe))])

            print "{0:>16} |{1:>16} |{2:>16} |{3:>16}".format(
              (word), stripe[0][0]+" "+str(stripe[0][1]), stripe[1][0]+" "+str(
stripe[1][1]), stripe[2][0]+" "+str(stripe[2][1]))
```

```
_____________________
Systems test  1  - Inverted Index
_____________________
_____________________
             "a" |          bill 4 |     biography 4 |            by 4
          "bill" |             a 27 |  establishing 4 |           for 4
     "biography" |             a 27 |       general 4 |        george 4
            "by" |             a 27 |          city 4 |           sea 4
          "case" |             a 27 |        female 4 |    government 4
       "child's" |             a 27 |     christmas 4 |            in 7
     "christmas" |             a 27 |       child's 4 |            in 7
"circumstantial" |             a 27 |     narrative 4 |           of 16
          "city" |             a 27 |            by 4 |           sea 4
    "collection" |             a 27 |         fairy 4 |         forms 3
  "establishing" |             a 27 |          bill 4 |           for 4
         "fairy" |             a 27 |    collection 5 |           of 16
        "female" |             a 27 |          case 7 |           of 16
           "for" |             a 27 |          bill 4 |  establishing 4
         "forms" |             a 27 |    collection 5 |           of 16
       "general" |             a 27 |     biography 4 |        george 4
        "george" |             a 27 |     biography 4 |       general 4
    "government" |             a 27 |          case 7 |            in 7
            "in" |             a 27 |          case 7 |       child's 4
       "limited" |             a 27 |          case 7 |           of 16
     "narrative" |             a 27 |circumstantial 4 |           of 16
            "of" |             a 27 |     biography 4 |          case 7
     "religious" |             a 27 |          bill 4 |  establishing 4
           "sea" |             a 27 |            by 4 |          city 4
         "study" |             a 27 |          case 7 |        female 4
         "tales" |             a 27 |    collection 5 |         fairy 4
           "the" |             a 27 |            by 4 |circumstantial 4
         "wales" |             a 27 |       child's 4 |     christmas 4
_____________________
_____________________
Systems test  2  - Inverted Index
_____________________
_____________________
         "atlas" |          boon 3 |        dipped 3 |
          "boon" |         atlas 2 |          cava 2 |        dipped 3
          "cava" |          boon 3 |        dipped 3 |
        "dipped" |         atlas 2 |          boon 3 |          cava 2
_____________________
_____________________
Systems test  3  - Inverted Index
_____________________
_____________________
             "M" |          DocC 4 |                 |
             "N" |          DocC 4 |                 |
             "X" |          DocA 3 |          DocB 2 |
             "Y" |          DocA 3 |          DocB 2 |          DocC 4
             "Z" |          DocA 3 |          DocC 4 |
```

#

Pretty print systems tests for generating Inverted Index

#

import json

for i in range(1,4): print "—"*100 print "Systems test ",i," - Inverted Index" print "—"*100
with open("systems_test*index*"+str(i),"r") as f: lines = f.readlines() for line in lines: line = line.str word,stripe = line.split("\t")
stripe = json.loads(stripe) stripe.extend([["",""] for _ in xrange(3 - len(stripe))])

```
        print "{0:>16} |{1:>16} |{2:>16} |{3:>16}".format(
            (word), stripe[0][0]+" "+str(stripe[0][1]), stripe[1][0]+" "+str(stripe[1
    ][1]), stripe[2][0]+" "+str(stripe[2][1]))
```

Inverted Index

```
————————————————————————————————————————————————————————————————————————————————
Systems test 1 - Inverted Index
————————————————————————————————————————————————————————————————————————————————
"a" | bill 4 | biography 4 | by 4 "bill" | a 27 | establishing 4 | for 4 "biography" | a 27 | general 4 | george 4 "by" | a 27 | city 4
| sea 4 "case" | a 27 | female 4 | government 4 "child's" | a 27 | christmas 4 | in 7 "christmas" | a 27 | child's 4 | in 7
"circumstantial" | a 27 | narrative 4 | of 15 "city" | a 27 | by 4 | sea 4 "collection" | a 27 | fairy 4 | forms 3 "establishing" | a
27 | bill 4 | for 4 "fairy" | a 27 | collection 5 | of 15 "female" | a 27 | case 7 | of 15 "for" | a 27 | bill 4 | establishing 4 "forms"
| a 27 | collection 5 | of 15 "general" | a 27 | biography 4 | george 4 "george" | a 27 | biography 4 | general 4 "government"
| a 27 | case 7 | in 7 "in" | a 27 | case 7 | child's 4 "limited" | a 27 | case 7 | of 15 "narrative" | a 27 |circumstantial 4 | of 15
"of" | a 27 | biography 4 | case 7 "religious" | a 27 | bill 4 | establishing 4 "sea" | a 27 | by 4 | city 4 "study" | a 27 | case 7 |
female 4 "tales" | a 27 | collection 5 | fairy 4 "the" | a 27 | by 4 |circumstantial 4 "wales" | a 27 | child's 4 | christmas 4
————————————————————————————————————————————————————————————————————————————————
Systems test 2 - Inverted Index
————————————————————————————————————————————————————————————————————————————————
"atlas" | boon 3 | dipped 3 |
"boon" | atlas 2 | cava 2 | dipped 3 "cava" | boon 3 | dipped 3 |
"dipped" | atlas 2 | boon 3 | cava 2
————————————————————————————————————————————————————————————————————————————————
Systems test 3 - Inverted Index
————————————————————————————————————————————————————————————————————————————————
"M" | DocC 4 | |
"N" | DocC 4 | |
"X" | DocA 3 | DocB 2 |
"Y" | DocA 3 | DocB 2 | DocC 4 "Z" | DocA 3 | DocC 4 |
```

(3) Calculate similarities - run the commands and insure that your output matches the output below

NOTE: you must run in hadoop mode to generate sorted similarities

In [38]:
```
!python similarity.py -r hadoop systems_test_index_1 --cmdenv PATH=/opt/anaconda/bin:$PATH  > systems_test_similarities_1
```

```
No configs found; falling back on auto-configuration
Looking for hadoop binary in $PATH...
Found hadoop binary: /usr/bin/hadoop
Using Hadoop version 2.6.0
Looking for Hadoop streaming jar in /home/hadoop/contrib...
Looking for Hadoop streaming jar in /usr/lib/hadoop-mapreduce...
Found Hadoop streaming jar: /usr/lib/hadoop-mapreduce/hadoop-streaming.jar
Creating temp directory /tmp/similarity.root.20170612.213258.131085
Copying local files to hdfs:///user/root/tmp/mrjob/similarity.root.20170612.21
3258.131085/files/...
Detected hadoop configuration property names that do not match hadoop version
2.6.0:
The have been translated as follows
 mapred.output.key.comparator.class: mapreduce.job.output.key.comparator.class
mapred.text.key.comparator.options: mapreduce.partition.keycomparator.options
mapred.text.key.partitioner.options: mapreduce.partition.keypartitioner.option
s
Running step 1 of 1...
  mapred.text.key.partitioner.options is deprecated. Instead, use mapreduce.pa
rtition.keypartitioner.options
  packageJobJar: [] [/usr/jars/hadoop-streaming-2.6.0-cdh5.7.0.jar] /tmp/strea
mjob3718002972068040596.jar tmpDir=null
  Connecting to ResourceManager at /0.0.0.0:8032
  Connecting to ResourceManager at /0.0.0.0:8032
  Total input paths to process : 1
  number of splits:2
  Submitting tokens for job: job_1497275441269_0026
  Submitted application application_1497275441269_0026
  The url to track the job: http://quickstart.cloudera:8088/proxy/application_
1497275441269_0026/
  Running job: job_1497275441269_0026
  Job job_1497275441269_0026 running in uber mode : false
   map 0% reduce 0%
   map 50% reduce 0%
   map 100% reduce 0%
   map 100% reduce 100%
  Job job_1497275441269_0026 completed successfully
  Output directory: hdfs:///user/root/tmp/mrjob/similarity.root.20170612.21325
8.131085/output
Counters: 49
        File Input Format Counters
                Bytes Read=2868
        File Output Format Counters
                Bytes Written=21828
        File System Counters
                FILE: Number of bytes read=7280
                FILE: Number of bytes written=371199
                FILE: Number of large read operations=0
                FILE: Number of read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=3206
                HDFS: Number of bytes written=21828
                HDFS: Number of large read operations=0
                HDFS: Number of read operations=9
                HDFS: Number of write operations=2
        Job Counters
                Data-local map tasks=2
                Launched map tasks=2
                Launched reduce tasks=1
                Total megabyte-seconds taken by all map tasks=12429312
                Total megabyte-seconds taken by all reduce tasks=5079040
                Total time spent by all map tasks (ms)=12138
                Total time spent by all maps in occupied slots (ms)=12138
                Total time spent by all reduce tasks (ms)=4960
                Total time spent by all reduces in occupied slots (ms)=4960
                Total vcore-seconds taken by all map tasks=12138
                Total vcore-seconds taken by all reduce tasks=4960
        Map-Reduce Framework
                CPU time spent (ms)=2480
                Combine input records=318
```

In [39]:
```
!python similarity.py -r hadoop systems_test_index_2 --cmdenv PATH=/opt/anacond
a/bin:$PATH > systems_test_similarities_2
```

```
No configs found; falling back on auto-configuration
Looking for hadoop binary in $PATH...
Found hadoop binary: /usr/bin/hadoop
Using Hadoop version 2.6.0
Looking for Hadoop streaming jar in /home/hadoop/contrib...
Looking for Hadoop streaming jar in /usr/lib/hadoop-mapreduce...
Found Hadoop streaming jar: /usr/lib/hadoop-mapreduce/hadoop-streaming.jar
Creating temp directory /tmp/similarity.root.20170612.213401.115261
Copying local files to hdfs:///user/root/tmp/mrjob/similarity.root.20170612.21
3401.115261/files/...
Detected hadoop configuration property names that do not match hadoop version
2.6.0:
The have been translated as follows
 mapred.output.key.comparator.class: mapreduce.job.output.key.comparator.class
mapred.text.key.comparator.options: mapreduce.partition.keycomparator.options
mapred.text.key.partitioner.options: mapreduce.partition.keypartitioner.option
s
Running step 1 of 1...
  mapred.text.key.partitioner.options is deprecated. Instead, use mapreduce.pa
rtition.keypartitioner.options
  packageJobJar: [] [/usr/jars/hadoop-streaming-2.6.0-cdh5.7.0.jar] /tmp/strea
mjob8473411775053507457.jar tmpDir=null
  Connecting to ResourceManager at /0.0.0.0:8032
  Connecting to ResourceManager at /0.0.0.0:8032
  Total input paths to process : 1
  number of splits:2
  Submitting tokens for job: job_1497275441269_0027
  Submitted application application_1497275441269_0027
  The url to track the job: http://quickstart.cloudera:8088/proxy/application_
1497275441269_0027/
  Running job: job_1497275441269_0027
  Job job_1497275441269_0027 running in uber mode : false
   map 0% reduce 0%
   map 50% reduce 0%
   map 100% reduce 0%
   map 100% reduce 100%
  Job job_1497275441269_0027 completed successfully
  Output directory: hdfs:///user/root/tmp/mrjob/similarity.root.20170612.21340
1.115261/output
Counters: 49
        File Input Format Counters
                Bytes Read=206
        File Output Format Counters
                Bytes Written=280
        File System Counters
                FILE: Number of bytes read=277
                FILE: Number of bytes written=357193
                FILE: Number of large read operations=0
                FILE: Number of read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=544
                HDFS: Number of bytes written=280
                HDFS: Number of large read operations=0
                HDFS: Number of read operations=9
                HDFS: Number of write operations=2
        Job Counters
                Data-local map tasks=2
                Launched map tasks=2
                Launched reduce tasks=1
                Total megabyte-seconds taken by all map tasks=13785088
                Total megabyte-seconds taken by all reduce tasks=5533696
                Total time spent by all map tasks (ms)=13462
                Total time spent by all maps in occupied slots (ms)=13462
                Total time spent by all reduce tasks (ms)=5404
                Total time spent by all reduces in occupied slots (ms)=5404
                Total vcore-seconds taken by all map tasks=13462
                Total vcore-seconds taken by all reduce tasks=5404
        Map-Reduce Framework
                CPU time spent (ms)=2470
                Combine input records=20
```

In [41]:
```
!python similarity.py -r hadoop systems_test_index_3 --cmdenv PATH=/opt/anacond
a/bin:$PATH > systems_test_similarities_3
```

```
No configs found; falling back on auto-configuration
Looking for hadoop binary in $PATH...
Found hadoop binary: /usr/bin/hadoop
Using Hadoop version 2.6.0
Looking for Hadoop streaming jar in /home/hadoop/contrib...
Looking for Hadoop streaming jar in /usr/lib/hadoop-mapreduce...
Found Hadoop streaming jar: /usr/lib/hadoop-mapreduce/hadoop-streaming.jar
Creating temp directory /tmp/similarity.root.20170612.213508.518576
Copying local files to hdfs:///user/root/tmp/mrjob/similarity.root.20170612.21
3508.518576/files/...
Detected hadoop configuration property names that do not match hadoop version
2.6.0:
The have been translated as follows
 mapred.output.key.comparator.class: mapreduce.job.output.key.comparator.class
mapred.text.key.comparator.options: mapreduce.partition.keycomparator.options
mapred.text.key.partitioner.options: mapreduce.partition.keypartitioner.option
s
Running step 1 of 1...
  mapred.text.key.partitioner.options is deprecated. Instead, use mapreduce.pa
rtition.keypartitioner.options
  packageJobJar: [] [/usr/jars/hadoop-streaming-2.6.0-cdh5.7.0.jar] /tmp/strea
mjob3333690816835545755.jar tmpDir=null
  Connecting to ResourceManager at /0.0.0.0:8032
  Connecting to ResourceManager at /0.0.0.0:8032
  Total input paths to process : 1
  number of splits:2
  Submitting tokens for job: job_1497275441269_0028
  Submitted application application_1497275441269_0028
  The url to track the job: http://quickstart.cloudera:8088/proxy/application_
1497275441269_0028/
  Running job: job_1497275441269_0028
  Job job_1497275441269_0028 running in uber mode : false
   map 0% reduce 0%
   map 100% reduce 0%
   map 100% reduce 100%
  Job job_1497275441269_0028 completed successfully
  Output directory: hdfs:///user/root/tmp/mrjob/similarity.root.20170612.21350
8.518576/output
Counters: 49
        File Input Format Counters
                Bytes Read=167
        File Output Format Counters
                Bytes Written=156
        File System Counters
                FILE: Number of bytes read=160
                FILE: Number of bytes written=356959
                FILE: Number of large read operations=0
                FILE: Number of read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=505
                HDFS: Number of bytes written=156
                HDFS: Number of large read operations=0
                HDFS: Number of read operations=9
                HDFS: Number of write operations=2
        Job Counters
                Data-local map tasks=2
                Launched map tasks=2
                Launched reduce tasks=1
                Total megabyte-seconds taken by all map tasks=24995840
                Total megabyte-seconds taken by all reduce tasks=7512064
                Total time spent by all map tasks (ms)=24410
                Total time spent by all maps in occupied slots (ms)=24410
                Total time spent by all reduce tasks (ms)=7336
                Total time spent by all reduces in occupied slots (ms)=7336
                Total vcore-seconds taken by all map tasks=24410
                Total vcore-seconds taken by all reduce tasks=7336
        Map-Reduce Framework
                CPU time spent (ms)=2780
                Combine input records=18
                Combine output records=7
```

In [42]:
```
##########################################
# Pretty print systems tests
##########################################

import json
for i in range(3,0,-1):
  print '—'*110
  print "Systems test ",i," - Similarity measures"
  print '—'*110
  print "{0:>15} |{1:>15}|{2:>15}".format(
            "pair",  "jaccard","Dice")
  print '-'*110

  with open("systems_test_similarities_"+str(i),"r") as f:
      lines = f.readlines()
      for line in lines:
          line = line.strip()
          pair,stripe = line.split("\t")
          stripe = json.loads(stripe)

          print "{0:>15} |{1:>15f}|{2:>15f}".format(
              pair,float(stripe['jaccard']),float(stripe['dice'] ))
#          print "{0:>15f} |{1:>15} |{2:>15f} |{3:>15f} |{4:>15f} |{5:>15f}".fo
rmat(
#                float(avg), stripe[0], float(stripe[1]), float(stripe[2]), floa
t(stripe[3]), float(stripe[4]))
          #print
```

```
_____________________________________________________________________________________
_____________________________
Systems test  3  - Similarity measures
_____________________________________________________________________________________
_____________________________
           pair |        jaccard|           Dice
-------------------------------------------------------------------------------------
--------------------------------
["DocA","DocB"] |       0.666667|       0.800000
["DocA","DocC"] |       0.400000|       0.571429
["DocB","DocC"] |       0.200000|       0.333333
_____________________________________________________________________________________
_____________________________
Systems test  2  - Similarity measures
_____________________________________________________________________________________
_____________________________
           pair |        jaccard|           Dice
-------------------------------------------------------------------------------------
--------------------------------
["atlas","dipped"] |       0.250000|       0.400000
["atlas","boon"] |       0.250000|       0.400000
["atlas","cava"] |       1.000000|       1.000000
["boon","dipped"] |       0.500000|       0.666667
["boon","cava"] |       0.250000|       0.400000
["cava","dipped"] |       0.250000|       0.400000
_____________________________________________________________________________________
_____________________________
Systems test  1  - Similarity measures
_____________________________________________________________________________________
_____________________________
           pair |        jaccard|           Dice
-------------------------------------------------------------------------------------
--------------------------------
["a","limited"] |       0.107143|       0.193548
    ["a","sea"] |       0.107143|       0.193548
["a","general"] |       0.107143|       0.193548
 ["a","female"] |       0.107143|       0.193548
     ["a","in"] |       0.214286|       0.352941
["a","religious"] |       0.107143|       0.193548
 ["a","george"] |       0.107143|       0.193548
["a","biography"] |       0.107143|       0.193548
   ["a","city"] |       0.107143|       0.193548
    ["a","for"] |       0.107143|       0.193548
  ["a","tales"] |       0.107143|       0.193548
["a","child's"] |       0.107143|       0.193548
  ["a","forms"] |       0.071429|       0.133333
  ["a","wales"] |       0.107143|       0.193548
["a","christmas"] |       0.107143|       0.193548
["a","government"] |       0.107143|       0.193548
["a","collection"] |       0.142857|       0.250000
     ["a","by"] |       0.107143|       0.193548
   ["a","case"] |       0.214286|       0.352941
["a","circumstantial"] |       0.107143|       0.193548
  ["a","fairy"] |       0.107143|       0.193548
     ["a","of"] |       0.535714|       0.697674
  ["a","study"] |       0.214286|       0.352941
   ["a","bill"] |       0.107143|       0.193548
["a","establishing"] |       0.107143|       0.193548
["a","narrative"] |       0.107143|       0.193548
    ["a","the"] |       0.214286|       0.352941
["bill","limited"] |       0.142857|       0.250000
["bill","female"] |       0.142857|       0.250000
["bill","general"] |       0.142857|       0.250000
 ["bill","sea"] |       0.142857|       0.250000
  ["bill","in"] |       0.100000|       0.181818
["bill","religious"] |       0.600000|       0.750000
["bill","george"] |       0.142857|       0.250000
["bill","biography"] |       0.142857|       0.250000
["bill","city"] |       0.142857|       0.250000
 ["bill","for"] |       0.600000|       0.750000
```

In [ ]:
```
##########################################
# Pretty print systems tests
##########################################

import json
for i in range(1,4):
  print '-'*110
  print "Systems test ",i," - Similarity measures"
  print '-'*110
  print "{0:>15} |{1:>15} |{2:>15} |{3:>15} |{4:>15} |{5:>15}".format(
          "average", "pair", "cosine", "jaccard", "overlap", "dice")
  print '-'*110

  with open("systems_test_similarities_"+str(i),"r") as f:
      lines = f.readlines()
      for line in lines:
          line = line.strip()
          avg,stripe = line.split("\t")
          stripe = json.loads(stripe)
          print avg,stripe
          print "{0:>15} |{1:>15} |{2:>15} |{3:>15} |{4:>15} |{5:>15}".format(
              stripe, avg, dice")
#          print "{0:>15f} |{1:>15} |{2:>15f} |{3:>15f} |{4:>15f} |{5:>15f}".fo
rmat(
#                float(avg), stripe[0], float(stripe[1]), float(stripe[2]), floa
t(stripe[3]), float(stripe[4]))
          #print
```

Pairwise Similairity

Systems test 1 - Similarity measures

---

average | pair | cosine | jaccard | overlap | dice

1.000000 | female - limited | 1.000000 | 1.000000 | 1.000000 | 1.000000 0.868292 | fairy - forms | 0.866025 | 0.750000 | 1.000000 | 0.857143 0.868292 | forms - tales | 0.866025 | 0.750000 | 1.000000 | 0.857143 0.830357 | case - study | 0.857143 | 0.750000 | 0.857143 | 0.857143 0.712500 | bill - establishing | 0.750000 | 0.600000 | 0.750000 | 0.750000 0.712500 | christmas - wales | 0.750000 | 0.600000 | 0.750000 | 0.750000 0.712500 |circumstantial - narrative | 0.750000 | 0.600000 | 0.750000 | 0.750000 0.712500 | by - sea | 0.750000 | 0.600000 | 0.750000 | 0.750000 0.712500 | by - city | 0.750000 | 0.600000 | 0.750000 | 0.750000 0.712500 | child's - wales | 0.750000 | 0.600000 | 0.750000 | 0.750000 0.712500 | biography - george | 0.750000 | 0.600000 | 0.750000 | 0.750000 0.712500 | child's - christmas | 0.750000 | 0.600000 | 0.750000 | 0.750000 ...

Systems test 2 - Similarity measures

---

average | pair | cosine | jaccard | overlap | dice

1.000000 | atlas - cava | 1.000000 | 1.000000 | 1.000000 | 1.000000 0.625000 | boon - dipped | 0.666667 | 0.500000 | 0.666667 | 0.666667 0.389562 | cava - dipped | 0.408248 | 0.250000 | 0.500000 | 0.400000 0.389562 | boon - cava | 0.408248 | 0.250000 | 0.500000 | 0.400000 0.389562 | atlas - dipped | 0.408248 | 0.250000 | 0.500000 | 0.400000 0.389562 | atlas - boon | 0.408248 | 0.250000 | 0.500000 | 0.400000

Systems test 3 - Similarity measures

---

average | pair | cosine | jaccard | overlap | dice

0.820791 | DocA - DocB | 0.816497 | 0.666667 | 1.000000 | 0.800000 0.553861 | DocA - DocC | 0.577350 | 0.400000 | 0.666667 | 0.571429 0.346722 | DocB - DocC | 0.353553 | 0.200000 | 0.500000 | 0.333333

=== END OF PHASE 1 ===

In [ ]: