DATASCI W261: Machine Learning at Scale

Group: 10
Names:
Prabhakar Gundugola,
Yi Jin,
Jaime Villalpando

Emails:
prabhakar@berkeley.edu,
yjin@ischool.berkeley.edu,
jaimegvl@ischool.berkeley.edu

Time of Initial Submission: Feb 15, 2016
Week 5: Homework 5
Date: February 19, 2016
Time of Submission: 03:00 AM PT

HW 5.0.

What is a data warehouse?
A data warehouse is a central repository that integrates copies of transaction data from disparate source systems and provisions them for analytical purposes. It accomodates data storage for any number of applications and is optimized for efficiently reading or retrieving large data sets and for aggregating data.

What is a Star schema? When is it used?
A Star schema is a style of data mart schema that consists of one or more fact tables referencing any number of dimension tables. The star schema is very effective for handling simpler queries.

The star schema gets its name from the physical model's resemblance to a star shape with a fact table at its center and the dimension tables surrounding it representing the star's points.

Sample Star Schema

HW 5.1

In the database world What is 3NF?
Third normal form is a normal form that is used in normalizing a database design to reduce the duplication of data and ensure referential integrity by ensuring that:
(1) the entity is in second normal form, and
(2) all the attributes in a table are determined only by the candidate keys of that table and not by any non-prime attributes. 3NF was designed to improve data base processing while minimizing storage costs.

3NF data modeling was ideal for online transaction processing (OLTP) applications with heavy order entry type of needs.

Does machine learning use data in 3NF? If so why?
Machine Learning does often use 3NF as the data required for Machine Learning algorithms are often in a very structured format.

In what form does ML consume data?
Machine Learning algorithms can consume data in a variety of formats including unstructured data. Machine learning algorithms perform much better when the data is normalized to the highest order that is 3NF.

Why would one use log files that are denormalized?
Denormalization of log files provides comprehensive view of all the log details and doesn't require the algorithm to perform complex joins. This is particularly helpful while retrieving the full log details of a particular transaction as it doesn't require grouping.

HW 5.2

Using MRJob, implement a hashside join (memory-backed map-side) for left, right and inner joins. Run your code on the data used in HW 4.4: (Recall HW 4.4: Find the most frequent visitor of each page using mrjob and the output of 4.2 (i.e., transfromed log file). In this output please include the webpage URL, webpageID and Visitor ID.) :

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

In [4]:
```
%%writefile preprocess52.py
#!/usr/bin/python
# preprocess52.py
# Author: Prabhakar Gundugola
# Description: Code for preprocess52.py

import sys

page_fp = open('pages.txt', 'w')
customer_fp = open('customers.txt', 'w')

visitor_id = None

with open('anonymous-msweb.data', 'r') as fp:
    for line in fp.readlines():
        tokens = line.strip().split(",")
        if tokens[0] == 'A':
            page_fp.write(line)
        elif tokens[0] == 'C':
            visitor_id = tokens[2]
        elif tokens[0] == 'V':
            line = line.strip() + ',C,' + visitor_id
            customer_fp.write(line + '\n')

page_fp.close()
customer_fp.close()
```

```
Overwriting preprocess52.py
```

In [5]:
```
!preprocess52.py
```

In [29]:
```
%%writefile mrfreqvisitor44.py
#!/usr/bin/python
# mrfreqvisitor44.py
# Author: Prabhakar Gundugola
# Description: Code for finding top visitors for each page

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

class MRFreqVisitor(MRJob):
    #def configure_options(self):
    #    super(MRFreqVisitor,self).configure_options()
    #    self.add_passthrough_option('--joinType', type='str', default
='inner')

    def mapper_freq(self, _, line):
        tokens = line.strip().split(',')
        page_id = tokens[1]
        visitor_id = tokens[4]
        yield (page_id, visitor_id), 1

    def reducer_freq(self, key, counts):
        yield (key[0], key[1]), sum(counts)



    def mapper_sort(self, key, count):
        yield key[0], (count, key[1])

    def reducer_sort(self, page, counts):
        yield page, max(counts)

    def steps(self):
        return [
            MRStep(mapper=self.mapper_freq,
                   reducer=self.reducer_freq
                  ),
            MRStep(mapper=self.mapper_sort,
                  reducer=self.reducer_sort)
        ]

if __name__ == "__main__":
    MRFreqVisitor.run()
```

```
Overwriting mrfreqvisitor44.py
```

In [30]:
```
%load_ext autoreload
%autoreload 2

from mrfreqvisitor44 import MRFreqVisitor
mr_job = MRFreqVisitor(args=['customers.txt'])

output_data = []

with mr_job.make_runner() as runner:
    runner.run()

    for line in runner.stream_output():
        output_data.append(mr_job.parse_output_line(line))
```

```
WARNING:mrjob.runner:
WARNING:mrjob.runner:PLEASE NOTE: Starting in mrjob v0.5.0, protocols w
ill be strict by default. It's recommended you run your job with --stri
ct-protocols or set up mrjob.conf as described at https://pythonhosted.
org/mrjob/whats-new.html#ready-for-strict-protocols
WARNING:mrjob.runner:

The autoreload extension is already loaded. To reload it, use:
  %reload_ext autoreload
```

In [36]:
```
with open('topvisitors.txt', 'w') as fp:
    for i in range(len(output_data)):
        line = str(output_data[i][0]) + ',' + str(output_data[i][1][1])
+ ',' + str(output_data[i][1][0]) + '\n'
        fp.write(line)
```

Tables

- Pages
- Visitors

(1) Left joining Table Left with Table Right

In [129]:

```
%%writefile hashjoin52.py
#!/usr/bin/python
# hashjoin52.py
# Author: Prabhakar Gundugola
# Description: Code for MRJob - Hash Join

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

class MRHashJoin(MRJob):

    # Setup of options configuration
    def configure_options(self):
        super(MRHashJoin, self).configure_options()
        self.add_passthrough_option('--jointype', type='str', default='i
nner')

    def load_options(self, args):
        super(MRHashJoin, self).load_options(args)
        self.jointype = self.options.jointype

    # Load the page URL's
    def mapper_join_init(self):
        self.seenpages = set()
        self.pages = {}

        with open('pages.txt', 'r') as f:
            #print 'Opened pages.txt'
            count = 0
            for line in f.readlines():
                count += 1
                tokens = line.strip().replace('"', '').split(',')
                self.pages[tokens[1]] = (tokens[2], tokens[3], tokens
[4])
            #print count

    def mapper_join(self, _, line):
        self.increment_counter('Counter', 'Mapper', 1)
        tokens = line.strip().split(",")
        self.seenpages.add(tokens[0])

        # Left join
        if tokens[0] in self.pages:
            page_attributes = self.pages[tokens[0]]
            #print tokens[0], page_attributes[2], tokens[1], int(tokens
[2])
            yield (tokens[0], page_attributes[2], tokens[1]), int(tokens
[2])
        else:
            if self.jointype != 'inner':
                #print tokens[0], None, tokens[1], int(tokens[2])
                yield (tokens[0], None, tokens[1]), int(tokens[2])
```

```
    def mapper_join_final(self):
        if self.jointype == 'right':
            for page in self.pages:
                if page not in self.seenpages:
                    yield (page, self.pages[page][2], str(0)), 1

    def reducer_join(self, key, values):
        #print key, sum(values)
        yield (key[0], key[1], key[2]), sum(values)

    def steps(self):
        if self.jointype != 'right':
            return [
                MRStep(mapper_init=self.mapper_join_init,
                       mapper=self.mapper_join,
                       reducer=self.reducer_join
                      )
            ]
        else:
            return [
                MRStep(mapper_init=self.mapper_join_init,
                       mapper=self.mapper_join,
                       mapper_final=self.mapper_join_final,
                       reducer=self.reducer_join
                      )
            ]

if __name__ == "__main__":
    MRHashJoin.run()
```

```
Overwriting hashjoin52.py
```

In [130]:
```
%load_ext autoreload
%autoreload 2
from hashjoin52 import MRHashJoin

def run_mrjob(jointype):
    mr_job = MRHashJoin(args=['topvisitors.txt', '--file', 'pages.txt',
'--jointype', jointype])
    output_data = []

    with mr_job.make_runner() as runner:
        runner.run()

        for line in runner.stream_output():
            output_data.append(mr_job.parse_output_line(line))

    return output_data

leftjoin_data = run_mrjob('left')
rightjoin_data = run_mrjob('right')
innerjoin_data = run_mrjob('inner')
print 'Left Join: ' + str(len(leftjoin_data))
print 'Right Join: ' + str(len(rightjoin_data))
print 'Inner Join: ' + str(len(innerjoin_data))
```

```
WARNING:mrjob.runner:
WARNING:mrjob.runner:PLEASE NOTE: Starting in mrjob v0.5.0, protocols w
ill be strict by default. It's recommended you run your job with --stri
ct-protocols or set up mrjob.conf as described at https://pythonhosted.
org/mrjob/whats-new.html#ready-for-strict-protocols
WARNING:mrjob.runner:
WARNING:mrjob.runner:
WARNING:mrjob.runner:PLEASE NOTE: Starting in mrjob v0.5.0, protocols w
ill be strict by default. It's recommended you run your job with --stri
ct-protocols or set up mrjob.conf as described at https://pythonhosted.
org/mrjob/whats-new.html#ready-for-strict-protocols
WARNING:mrjob.runner:
WARNING:mrjob.runner:
WARNING:mrjob.runner:PLEASE NOTE: Starting in mrjob v0.5.0, protocols w
ill be strict by default. It's recommended you run your job with --stri
ct-protocols or set up mrjob.conf as described at https://pythonhosted.
org/mrjob/whats-new.html#ready-for-strict-protocols
WARNING:mrjob.runner:
```

```
The autoreload extension is already loaded. To reload it, use:
  %reload_ext autoreload
Left Join: 285
Right Join: 294
Inner Join: 285
```

HW 5.3

For the remainder of this assignment you will work with two datasets:

1: unit/systems test data set: SYSTEMS TEST DATASET

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

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

In [141]:
```
import csv

data = {}
data['A'] = ['X']*20
data['A'].extend(['Y']*30)
data['A'].extend(['Z']*5)
data['B'] = ['X']*100
data['B'].extend(['Y']*20)
data['C'] = ['M']*5
data['C'].extend(['N']*20)
data['C'].extend(['Z']*5)

with open('testdata.csv', 'w') as f:
    out = csv.writer(f)
    for doc in data:
        row = [doc]
        row.extend(data[doc])
        out.writerow(row)
```

2: 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 on s3:

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

s3://filtered-5grams/

For each HW 5.3 -5.5 Please unit test and system test your code with with SYSTEMS TEST DATASET and show the results. Please compute the expected answer by hand and show your hand calculations. Then show the results you get with you system. Final show your results on the Google n-grams dataset

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

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

Do some EDA on this dataset using mrjob, e.g.,

- Longest 5-gram (number of characters)
- Top 10 most frequent words (please use the count information), i.e., unigrams
- Most/Least densely appearing words (count/pages_count) sorted in decreasing order of relative frequency (Hint: save to PART-000* and take the head -n 1000)
- Distribution of 5-gram sizes (using counts info.) sorted in decreasing order of relative frequency. OPTIONAL Question:
- Plot the log-log plot of the frequency distributuion of unigrams. Does it follow power law distribution?

For more background see: https://en.wikipedia.org/wiki/Log%E2%80%93log_plot (https://en.wikipedia.org/wiki/Log%E2%80%93log_plot) https://en.wikipedia.org/wiki/Power_law (https://en.wikipedia.org/wiki/Power_law)

In [259]:

```
%%writefile mrgooglegram.py
#!/usr/bin/python
# mrgooglegram.py
# Author: Prabhakar Gundugola
# Description: Google n-grams dataset

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

class MRGoogleGram(MRJob):

    # Configure options
    def configure_options(self):
        super(MRGoogleGram, self).configure_options()
        self.add_passthrough_option('--otype', type='str')

    # Load options
    def load_options(self, args):
        super(MRGoogleGram, self).load_options(args)
        self.otype = self.options.otype

    '''
    Q.1. Longest 5-gram (number of characters)
    '''

    # Mapper Init
    def mapper_longest_5gram_init(self):
        self.longest = 0
        self.longest_5gram = None
    # Mapper
    def mapper_longest_5gram(self, _, line):
        tokens = line.strip().split('\t')
        if int(len(tokens[0])) > self.longest:
            self.longest = int(len(tokens[0]))
            self.longest_5gram = tokens[0]

    # Emit the longest 5gram of mapper
    def mapper_longest_5gram_final(self):
        yield self.longest_5gram, self.longest

    # Reducer init
    def reducer_longest_5gram_init(self):
        self.longest = 0
        self.longest_5gram = None

    # Reducer
    def reducer_longest_5gram(self, key, values):
        for v in values:
            if v > self.longest:
                self.longest = v
                self.longest_5gram = key

    # Reducer final
```

```
    def reducer_longest_5gram_final(self):
        yield self.longest_5gram, self.longest

    '''
    Q.2. Top most frequent words (please use the count information)
    '''

    def mapper_word_freq(self, _, line):
        tokens = line.strip().split('\t')

        for word in tokens[0].lower().split():
            yield word, int(tokens[1])

    def combiner_word_freq(self, word, counts):
        yield word, sum(counts)

    def reducer_word_freq(self, word, counts):
        yield word, sum(counts)

    def reducer_sort_freq_init(self):
        self.count = 0
    def reducer_sort_freq_final(self, word, counts):
        if self.count >= 0:
            self.count += 1
            yield word, counts.next()

    '''
    Q.3. Most/Least densely appearing words (count/pages_count) sorted i
n decreasing order of relative frequency (Hint: save to PART-000* and ta
ke the head -n 1000)
    '''
    def mapper_word_density(self, _, line):
        tokens = line.strip().lower().split('\t')
        count = int(tokens[1])
        pages_count = int(tokens[2])
        for word in tokens[0].split():
            yield word, (count, pages_count)

    def combiner_word_density(self, word, values):
        c, pc = 0.0, 0.0
        for val in values:
            c += val[0]
            pc += val[1]
        yield word, (c, pc)

    def reducer_word_density(self, word, values):
        c, pc = 0.0, 0.0
        for val in values:
            c += val[0]
            pc += val[1]

        yield word, (c/pc)
```

```
    '''
    Q.4. Distribution of 5-gram sizes (using counts info.) sorted in dec
reasing order of relative frequency.
    '''
    def mapper_5gram_size(self, _, line):
        tokens = line.strip().lower().split('\t')
        yield len(tokens[0]), float(tokens[1])

    def combiner_5gram_size(self, key, values):
        yield key, sum(values)

    def reducer_5gram_size(self, key, values):
        yield key, sum(values)

    '''
    Steps to run MRJob
    '''
    def steps(self):
        if self.otype == '1':
            return [
                MRStep(mapper_init=self.mapper_longest_5gram_init,
                       mapper=self.mapper_longest_5gram,
                       mapper_final=self.mapper_longest_5gram_final,
                       reducer_init=self.reducer_longest_5gram_init,
                       reducer=self.reducer_longest_5gram,
                       reducer_final=self.reducer_longest_5gram_final)
                ]
        elif self.otype == '2':
            return [
                MRStep(mapper=self.mapper_word_freq,
                       combiner=self.combiner_word_freq,
                       reducer=self.reducer_word_freq),
                MRStep(reducer_init=self.reducer_sort_freq_init,
                       reducer=self.reducer_sort_freq_final,
                       jobconf={
                        'stream.num.map.output.key.fields':2,
                        'mapred.output.key.comparator.class':'org.apach
e.hadoop.mapred.lib.KeyFieldBasedComparator',
                        'mapred.text.key.comparator.options':'-k2,2nr',
                        'mapred.reduce.tasks':1
                    })
            ]
        elif self.otype == '3':
            return [
                MRStep(mapper=self.mapper_word_density,
                       combiner=self.combiner_word_density,
                       reducer=self.reducer_word_density,
                       jobconf={
                        'stream.num.map.output.key.fields':2,
                        'mapred.output.key.comparator.class':'org.apach
e.hadoop.mapred.lib.KeyFieldBasedComparator',
                        'mapred.text.key.comparator.options':'-k2,2nr',
                        'mapred.reduce.tasks':1
```

```
                    })
            ]
        elif self.otype == '4':
            return [
                MRStep(mapper=self.mapper_5gram_size,
                       combiner=self.combiner_5gram_size,
                       reducer=self.reducer_5gram_size,
                       jobconf={
                        'stream.num.map.output.key.fields':2,
                        'mapred.output.key.comparator.class':'org.apach
e.hadoop.mapred.lib.KeyFieldBasedComparator',
                        'mapred.text.key.comparator.options':'-k2,2nr',
                        'mapred.reduce.tasks':1
                    })
            ]

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

```
Overwriting mrgooglegram.py
```

In [258]:
```
%load_ext autoreload
%autoreload 2
from mrgooglegram import MRGoogleGram

def run_google_mrjob(otype):
    output_data = []

    mr_job = MRGoogleGram(args=['filtered-5Grams\googlebooks-eng-all-5gr
am-20090715-0-filtered.txt', '--otype', str(otype)])

    with mr_job.make_runner() as runner:
        runner.run()

        for line in runner.stream_output():
            output_data.append(mr_job.parse_output_line(line))

    return output_data

run_google_mrjob('4')
```

```
WARNING:mrjob.runner:
WARNING:mrjob.runner:PLEASE NOTE: Starting in mrjob v0.5.0, protocols w
ill be strict by default. It's recommended you run your job with --stri
ct-protocols or set up mrjob.conf as described at https://pythonhosted.
org/mrjob/whats-new.html#ready-for-strict-protocols
WARNING:mrjob.runner:
```

```
The autoreload extension is already loaded. To reload it, use:
  %reload_ext autoreload
```

```
Out[258]: [(10, 140.0),
           (11, 154.0),
           (12, 461.0),
           (13, 2850.0),
           (14, 36515.0),
           (15, 104455.0),
           (16, 331008.0),
           (17, 953963.0),
           (18, 1628712.0),
           (19, 2540154.0),
           (20, 3502503.0),
           (21, 4318999.0),
           (22, 4815021.0),
           (23, 4737598.0),
           (24, 4478362.0),
           (25, 4331008.0),
           (26, 3768905.0),
           (27, 3299934.0),
           (28, 2879717.0),
           (29, 2223998.0),
           (30, 1793666.0),
           (31, 1365446.0),
           (32, 1124513.0),
           (33, 829152.0),
           (34, 590094.0),
           (35, 470756.0),
           (36, 307705.0),
           (37, 234615.0),
           (38, 158253.0),
           (39, 123170.0),
           (40, 85315.0),
           (41, 52853.0),
           (42, 47785.0),
           (43, 31902.0),
           (44, 21210.0),
           (45, 12983.0),
           (46, 8923.0),
           (47, 4990.0),
           (48, 4746.0),
           (49, 3262.0),
           (50, 941.0),
           (51, 725.0),
           (52, 1926.0),
           (53, 977.0),
           (54, 166.0),
           (55, 680.0),
           (57, 142.0),
           (58, 95.0),
           (9, 104.0)]
```

root@prabhakar:~/hw5/HW5-Questions# python mrgooglegram.py -r emr --conf-path w261-mrhw5.conf s3://w261-pg-hw5/input/ --output-dir=s3://w261-pg-hw5/output/a10/ --otype 1 Got unexpected keyword arguments: ssh_tunnel using existing scratch bucket mrjob-7ef13761ff663fc8 using s3://mrjob-

7ef13761ff663fc8/tmp/ as our scratch dir on S3 creating tmp directory /tmp/mrgooglegram.root.20160219.091240.890266 writing master bootstrap script to /tmp/mrgooglegram.root.20160219.091240.890266/b.py PLEASE NOTE: Starting in mrjob v0.5.0, protocols will be strict by default. It's recommended you run your job with --strict-protocols or set up mrjob.conf as described at https://pythonhosted.org/mrjob/whats-new.html#ready-for-strict-protocols Copying non-input files into s3://mrjob-7ef13761ff663fc8/tmp/mrgooglegram.root.20160219.091240.890266/files/ Waiting 5.0s for S3 eventual consistency Creating Elastic MapReduce job flow Job flow created with ID: j-SX04QE6Y6PY8 Created new job flow j-SX04QE6Y6PY8 Job launched 30.7s ago, status STARTING: Provisioning Amazon EC2 capacity Job launched 61.5s ago, status STARTING: Provisioning Amazon EC2 capacity Job launched 92.4s ago, status STARTING: Provisioning Amazon EC2 capacity Job launched 123.1s ago, status STARTING: Provisioning Amazon EC2 capacity Job launched 153.9s ago, status STARTING: Provisioning Amazon EC2 capacity Job launched 184.7s ago, status STARTING: Configuring cluster software Job launched 215.5s ago, status STARTING: Configuring cluster software Job launched 246.2s ago, status BOOTSTRAPPING: Running bootstrap actions Job launched 277.0s ago, status RUNNING: Running step (mrgooglegram.root.20160219.091240.890266: Step 1 of 1) Opening ssh tunnel to Hadoop job tracker Connect to job tracker at: http://localhost:40541/jobtracker.jsp Job launched 308.8s ago, status RUNNING: Running step (mrgooglegram.root.20160219.091240.890266: Step 1 of 1) Unable to load progress from job tracker Job launched 344.6s ago, status RUNNING: Running step (mrgooglegram.root.20160219.091240.890266: Step 1 of 1) Job launched 375.4s ago, status RUNNING: Running step (mrgooglegram.root.20160219.091240.890266: Step 1 of 1) Job launched 406.1s ago, status RUNNING: Running step (mrgooglegram.root.20160219.091240.890266: Step 1 of 1) Oops, ssh subprocess exited with return code 255, restarting... Opening ssh tunnel to Hadoop job tracker Connect to job tracker at: http://localhost:40541/jobtracker.jsp Job completed. Running time was 155.0s (not counting time spent waiting for the EC2 instances) Fetching counters from S3... Waiting 5.0s for S3 eventual consistency Counters from step 1: File Input Format Counters : Bytes Read: 2156069116 File Output Format Counters : Bytes Written: 166 FileSystemCounters: FILE_BYTES_READ: 9068 FILE_BYTES_WRITTEN: 5046376 HDFS_BYTES_READ: 23769 S3_BYTES_READ: 2156069116 S3_BYTES_WRITTEN: 166 Job Counters : Launched map tasks: 191 Launched reduce tasks: 1 Rack-local map tasks: 191 SLOTS_MILLIS_MAPS: 1445212 SLOTS_MILLIS_REDUCES: 98720 Total time spent by all maps waiting after reserving slots (ms): 0 Total time spent by all reduces waiting after reserving slots (ms): 0 Map-Reduce Framework: CPU time spent (ms): 177280 Combine input records: 0 Combine output records: 0 Map input bytes: 2156069116 Map input records: 58682266 Map output bytes: 13528 Map output materialized bytes: 16361 Map output records: 188 Physical memory (bytes) snapshot: 127703691264 Reduce input groups: 188 Reduce input records: 188 Reduce output records: 1 Reduce shuffle bytes: 16361 SPLIT_RAW_BYTES: 23769 Spilled Records: 376 Total committed heap usage (bytes): 135643791360 Virtual memory (bytes) snapshot: 369885208576 Streaming final output from s3://w261-pg-hw5/output/a10/ "AIOPJUMRXUYVASLYHYPSIBEMAPODIKR UFRYDIUUOLBIGASUAURUSREXLISNAYE RNOONDQSRUNSUBUNOUGRABBERYAIRTC UTAHRAPTOREDILEIPMILBDUMMYUVERI SYEVRAHVELOCYALLOSAURUSLINROTSR" 159 removing tmp directory /tmp/mrgooglegram.root.20160219.091240.890266 Removing all files in s3://mrjob-7ef13761ff663fc8/tmp/mrgooglegram.root.20160219.091240.890266/ Removing all files in s3://mrjob-7ef13761ff663fc8/tmp/logs/j-SX04QE6Y6PY8/ Killing our SSH tunnel (pid 18079) Terminating job flow: j-SX04QE6Y6PY8 root@prabhakar:~/hw5/HW5-Questions#

The longest 5gram is:
"AIOPJUMRXUYVASLYHYPSIBEMAPODIKR UFRYDIUUOLBIGASUAURUSREXLISNAYE RNOONDQSRUNSUBUNOUGRABBERYAIRTC UTAHRAPTOREDILEIPMILBDUMMYUVERI SYEVRAHVELOCYALLOSAURUSLINROTSR" 159

Q.2. Top most frequent words (please use the count information)

Job launched 990.5s ago, status RUNNING: Running step (mrgooglegram.root.20160219.091941.123619: Step 1 of 2) root@prabhakar:~/hw5/HW5-Questions# python mrgooglegram.py -r emr --conf-path w261-mrhw5.conf s3://w261-pg-hw5/input/ --output-dir=s3://w261-pg-hw5/output/a11/ --otype 2 Got unexpected keyword arguments: ssh_tunnel using existing scratch bucket mrjob-7ef13761ff663fc8 using s3://mrjob-7ef13761ff663fc8/tmp/ as our scratch dir on S3 creating tmp directory /tmp/mrgooglegram.root.20160219.091941.123619 writing master bootstrap script to /tmp/mrgooglegram.root.20160219.091941.123619/b.py PLEASE NOTE: Starting in mrjob v0.5.0, protocols will be strict by default. It's recommended you run your job with --strict-protocols or set up mrjob.conf as described at https://pythonhosted.org/mrjob/whats-new.html#ready-for-strict-protocols Copying non-input files into s3://mrjob-7ef13761ff663fc8/tmp/mrgooglegram.root.20160219.091941.123619/files/ Waiting 5.0s for S3 eventual consistency Creating Elastic MapReduce job flow Job flow created with ID: j-2P9KKB48E4HHI Created new job flow j-2P9KKB48E4HHI Job launched 30.7s ago, status STARTING: Provisioning Amazon EC2 capacity Job launched 61.5s ago, status STARTING: Provisioning Amazon EC2 capacity Job launched 92.3s ago, status STARTING: Provisioning Amazon EC2 capacity Job launched 123.1s ago, status STARTING: Provisioning Amazon EC2 capacity Job launched 154.0s ago, status STARTING: Provisioning Amazon EC2 capacity Job launched 184.8s ago, status STARTING: Configuring cluster software Job launched 215.6s ago, status BOOTSTRAPPING: Running bootstrap actions Job launched 246.4s ago, status RUNNING: Running step (mrgooglegram.root.20160219.091941.123619: Step 1 of 2) Opening ssh tunnel to Hadoop job tracker Connect to job tracker at: http://localhost:40083/jobtracker.jsp Job launched 278.1s ago, status RUNNING: Running step (mrgooglegram.root.20160219.091941.123619: Step 1 of 2) Unable to load progress from job tracker Job launched 308.9s ago, status RUNNING: Running step (mrgooglegram.root.20160219.091941.123619: Step 1 of 2) Job launched 339.7s ago, status RUNNING: Running step (mrgooglegram.root.20160219.091941.123619: Step 1 of 2) Job launched 370.4s ago, status RUNNING: Running step (mrgooglegram.root.20160219.091941.123619: Step 1 of 2) Job launched 401.2s ago, status RUNNING: Running step (mrgooglegram.root.20160219.091941.123619: Step 1 of 2) Oops, ssh subprocess exited with return code 255, restarting... Opening ssh tunnel to Hadoop job tracker Connect to job tracker at: http://localhost:40083/jobtracker.jsp Job launched 433.0s ago, status RUNNING: Running step (mrgooglegram.root.20160219.091941.123619: Step 1 of 2) Unable to load progress from job tracker Job launched 463.8s ago, status RUNNING: Running step (mrgooglegram.root.20160219.091941.123619: Step 1 of 2) Job launched 494.5s ago, status RUNNING: Running step (mrgooglegram.root.20160219.091941.123619: Step 1 of 2) Job launched 525.3s ago, status RUNNING: Running step (mrgooglegram.root.20160219.091941.123619: Step 1 of 2) Job launched 556.0s ago, status RUNNING: Running step (mrgooglegram.root.20160219.091941.123619: Step 1 of 2) Oops, ssh subprocess exited with return code 255, restarting... Opening ssh tunnel to Hadoop job tracker Connect to job tracker at: http://localhost:40083/jobtracker.jsp Job launched 587.9s ago, status RUNNING:

Running step (mrgooglegram.root.20160219.091941.123619: Step 1 of 2) Unable to load progress from job tracker Job launched 618.7s ago, status RUNNING: Running step (mrgooglegram.root.20160219.091941.123619: Step 1 of 2) Job launched 649.5s ago, status RUNNING: Running step (mrgooglegram.root.20160219.091941.123619: Step 1 of 2) Job launched 680.3s ago, status RUNNING: Running step (mrgooglegram.root.20160219.091941.123619: Step 1 of 2) Job launched 711.1s ago, status RUNNING: Running step (mrgooglegram.root.20160219.091941.123619: Step 1 of 2) Oops, ssh subprocess exited with return code 255, restarting... Opening ssh tunnel to Hadoop job tracker Connect to job tracker at: http://localhost:40083/jobtracker.jsp Job launched 743.0s ago, status RUNNING: Running step (mrgooglegram.root.20160219.091941.123619: Step 1 of 2) Unable to load progress from job tracker Job launched 773.8s ago, status RUNNING: Running step (mrgooglegram.root.20160219.091941.123619: Step 1 of 2) Job launched 804.5s ago, status RUNNING: Running step (mrgooglegram.root.20160219.091941.123619: Step 1 of 2) Job launched 835.2s ago, status RUNNING: Running step (mrgooglegram.root.20160219.091941.123619: Step 1 of 2) Job launched 866.1s ago, status RUNNING: Running step (mrgooglegram.root.20160219.091941.123619: Step 1 of 2) Oops, ssh subprocess exited with return code 255, restarting... Opening ssh tunnel to Hadoop job tracker Connect to job tracker at: http://localhost:40083/jobtracker.jsp Job launched 897.9s ago, status RUNNING: Running step (mrgooglegram.root.20160219.091941.123619: Step 1 of 2) Unable to load progress from job tracker Job launched 928.8s ago, status RUNNING: Running step (mrgooglegram.root.20160219.091941.123619: Step 1 of 2) Job launched 959.7s ago, status RUNNING: Running step (mrgooglegram.root.20160219.091941.123619: Step 1 of 2) Job launched 990.5s ago, status RUNNING: Running step (mrgooglegram.root.20160219.091941.123619: Step 1 of 2) Job launched 1021.4s ago, status RUNNING: Running step (mrgooglegram.root.20160219.091941.123619: Step 1 of 2) Oops, ssh subprocess exited with return code 255, restarting... Opening ssh tunnel to Hadoop job tracker Connect to job tracker at: http://localhost:40083/jobtracker.jsp Job launched 1053.2s ago, status RUNNING: Running step (mrgooglegram.root.20160219.091941.123619: Step 1 of 2) Unable to load progress from job tracker Job launched 1084.0s ago, status RUNNING: Running step (mrgooglegram.root.20160219.091941.123619: Step 1 of 2) Job launched 1114.7s ago, status RUNNING: Running step (mrgooglegram.root.20160219.091941.123619: Step 1 of 2) Job launched 1145.5s ago, status RUNNING: Running step (mrgooglegram.root.20160219.091941.123619: Step 1 of 2) Opening ssh tunnel to Hadoop job tracker Connect to job tracker at: http://localhost:40083/jobtracker.jsp Job launched 1208.1s ago, status RUNNING: Running step (mrgooglegram.root.20160219.091941.123619: Step 1 of 2) Unable to load progress from job tracker Job launched 1238.9s ago, status RUNNING: Running step (mrgooglegram.root.20160219.091941.123619: Step 1 of 2) Job launched 1269.7s ago, status RUNNING: Running step (mrgooglegram.root.20160219.091941.123619: Step 1 of 2) Job launched 1300.5s ago, status RUNNING: Running step (mrgooglegram.root.20160219.091941.123619: Step 1 of 2) Job launched 1331.4s ago, status RUNNING: Running step (mrgooglegram.root.20160219.091941.123619: Step 1 of 2) Oops, ssh subprocess exited with return code 255, restarting... Opening ssh tunnel to Hadoop job tracker Connect to job tracker at: http://localhost:40083/jobtracker.jsp Job launched 1363.1s ago, status RUNNING: Running step (mrgooglegram.root.20160219.091941.123619: Step 2 of 2) Unable to load progress from job tracker Job launched 1393.9s ago, status RUNNING: Running step (mrgooglegram.root.20160219.091941.123619: Step 2 of 2) Job completed. Running time was 1158.0s (not counting time spent waiting for the EC2 instances) Fetching counters from S3... Waiting 5.0s for S3 eventual consistency Counters from step 1: File Input Format Counters : Bytes Read: 2156069116 File Output Format Counters : Bytes Written: 4158739 FileSystemCounters: FILE_BYTES_READ: 382729988 FILE_BYTES_WRITTEN: 290061315 HDFS_BYTES_READ: 23769 HDFS_BYTES_WRITTEN: 4158739

S3_BYTES_READ: 2156069116 Job Counters : Launched map tasks: 192 Launched reduce tasks: 13 Rack-local map tasks: 192 SLOTS_MILLIS_MAPS: 17573549 SLOTS_MILLIS_REDUCES: 5617208 Total time spent by all maps waiting after reserving slots (ms): 0 Total time spent by all reduces waiting after reserving slots (ms): 0 Map-Reduce Framework: CPU time spent (ms): 7288230 Combine input records: 306120824 Combine output records: 19532239 Map input bytes: 2156069116 Map input records: 58682266 Map output bytes: 3136729760 Map output materialized bytes: 86151971 Map output records: 293411330 Physical memory (bytes) snapshot: 141439377408 Reduce input groups: 269339 Reduce input records: 6822745 Reduce output records: 269339 Reduce shuffle bytes: 86151971 SPLIT_RAW_BYTES: 23769 Spilled Records: 26354984 Total committed heap usage (bytes): 143313600512 Virtual memory (bytes) snapshot: 387609788416 Counters from step 2: File Input Format Counters : Bytes Read: 4625309 File Output Format Counters : Bytes Written: 824 FileSystemCounters: FILE_BYTES_READ: 3072429 FILE_BYTES_WRITTEN: 7507367 HDFS_BYTES_READ: 4631389 S3_BYTES_WRITTEN: 824 Job Counters : Data-local map tasks: 40 Launched map tasks: 40 Launched reduce tasks: 1 SLOTS_MILLIS_MAPS: 295359 SLOTS_MILLIS_REDUCES: 20916 Total time spent by all maps waiting after reserving slots (ms): 0 Total time spent by all reduces waiting after reserving slots (ms): 0 Map-Reduce Framework: CPU time spent (ms): 49010 Combine input records: 0 Combine output records: 0 Map input bytes: 4158739 Map input records: 269339 Map output bytes: 4428078 Map output materialized bytes: 3333327 Map output records: 269339 Physical memory (bytes) snapshot: 19804708864 Reduce input groups: 269339 Reduce input records: 269339 Reduce output records: 51 Reduce shuffle bytes: 3333327 SPLIT_RAW_BYTES: 6080 Spilled Records: 538678 Total committed heap usage (bytes): 23605542912 Virtual memory (bytes) snapshot: 80484122624 Streaming final output from s3://w261-pg-hw5/output/a11/ "the" 5490815394 "of" 3698583299 "to" 2227866570 "in" 1421312776 "a" 1361123022 "and" 1149577477 "that" 802921147 "is" 758328796 "be" 688707130 "as" 492170314 "it" 487197064 "was" 470367538 "for" 462657760 "not" 400569858 "with" 376084831 "on" 357840502 "by" 347174023 "he" 320157811 "have" 317451227 "which" 282131051 "his" 268635443 "at" 268174875 "had" 256768897 "i" 255535477 "from" 249313825 "are" 248566965 "this" 238396958 "been" 234568049 "an" 206655720 "they" 185912107 "one" 180195771 "or" 164990333 "you" 158605523 "all" 152232780 "were" 148205733 "we" 147630572 "there" 142178507 "would" 139313915 "has" 138263598 "more" 136579896 "their" 130698530 "time" 126853684 "no" 123908710 "who" 118898831 "can" 116568172 "will" 115103704 "what" 112260044 "than" 112084413 "may" 108486904 "him" 107951952 "out" 104595840 removing tmp directory /tmp/mrgooglegram.root.20160219.091941.123619 Removing all files in s3://mrjob-7ef13761ff663fc8/tmp/mrgooglegram.root.20160219.091941.123619/ Removing all files in s3://mrjob-7ef13761ff663fc8/tmp/logs/j-2P9KKB48E4HHI/ Terminating job flow: j-2P9KKB48E4HHI root@prabhakar:~/hw5/HW5-Questions#

The Top 10 most frequent words:

"the" 5490815394 "of" 3698583299 "to" 2227866570 "in" 1421312776 "a" 1361123022 "and" 1149577477 "that" 802921147 "is" 758328796 "be" 688707130 "as" 492170314

Q.3. Most/Least densely appearing words (count/pages_count) sorted in decreasing order of relative frequency (Hint: save to PART-000* and take the head -n 1000)

In [ ]:

```
root@prabhakar:~/hw5/HW5-Questions# python mrgooglegram.py -r emr --conf
-path w261-mrhw5.conf s3://w261-pg-hw5/input/ --output-dir=s3://w261-pg-
hw5/output/a12/ --otype 3
Got unexpected keyword arguments: ssh_tunnel
using existing scratch bucket mrjob-7ef13761ff663fc8
using s3://mrjob-7ef13761ff663fc8/tmp/ as our scratch dir on S3
creating tmp directory /tmp/mrgooglegram.root.20160219.092348.936845
writing master bootstrap script to /tmp/mrgooglegram.root.20160219.09234
8.936845/b.py

PLEASE NOTE: Starting in mrjob v0.5.0, protocols will be strict by defau
lt. It's recommended you run your job with --strict-protocols or set up
mrjob.conf as described at https://pythonhosted.org/mrjob/whats-new.html
#ready-for-strict-protocols

Copying non-input files into s3://mrjob-7ef13761ff663fc8/tmp/mrgooglegra
m.root.20160219.092348.936845/files/
Waiting 5.0s for S3 eventual consistency
Creating Elastic MapReduce job flow
Job flow created with ID: j-2640HP1D34JM6
Created new job flow j-2640HP1D34JM6
Job launched 30.8s ago, status STARTING: Provisioning Amazon EC2 capacit
y
Job launched 61.6s ago, status STARTING: Provisioning Amazon EC2 capacit
y
Job launched 92.6s ago, status STARTING: Provisioning Amazon EC2 capacit
y
Job launched 123.3s ago, status STARTING: Provisioning Amazon EC2 capaci
ty
Job launched 154.1s ago, status STARTING: Provisioning Amazon EC2 capaci
ty
Job launched 184.8s ago, status STARTING: Configuring cluster software
Job launched 215.6s ago, status STARTING: Configuring cluster software
Job launched 246.5s ago, status BOOTSTRAPPING: Running bootstrap actions
Job launched 277.3s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Opening ssh tunnel to Hadoop job tracker
Connect to job tracker at: http://localhost:40737/jobtracker.jsp
Job launched 314.1s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Unable to load progress from job tracker
Job launched 344.8s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Job launched 375.6s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Job launched 406.4s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Oops, ssh subprocess exited with return code 255, restarting...
Opening ssh tunnel to Hadoop job tracker
Connect to job tracker at: http://localhost:40737/jobtracker.jsp
Job launched 438.2s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Unable to load progress from job tracker
```

```
Job launched 468.9s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Job launched 499.7s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Job launched 530.5s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Job launched 561.4s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Oops, ssh subprocess exited with return code 255, restarting...
Opening ssh tunnel to Hadoop job tracker
Connect to job tracker at: http://localhost:40737/jobtracker.jsp
Job launched 593.2s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Unable to load progress from job tracker
Job launched 623.9s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Job launched 654.7s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Job launched 685.4s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Job launched 716.2s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Oops, ssh subprocess exited with return code 255, restarting...
Opening ssh tunnel to Hadoop job tracker
Connect to job tracker at: http://localhost:40737/jobtracker.jsp
Job launched 748.0s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Unable to load progress from job tracker
Job launched 778.7s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Job launched 809.5s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Job launched 840.3s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Job launched 871.1s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Oops, ssh subprocess exited with return code 255, restarting...
Opening ssh tunnel to Hadoop job tracker
Connect to job tracker at: http://localhost:40737/jobtracker.jsp
Job launched 902.8s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Unable to load progress from job tracker
Job launched 933.6s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Job launched 964.3s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Job launched 995.1s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Job launched 1025.9s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Oops, ssh subprocess exited with return code 255, restarting...
Opening ssh tunnel to Hadoop job tracker
Connect to job tracker at: http://localhost:40737/jobtracker.jsp
```

```
Job launched 1057.7s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Unable to load progress from job tracker
Job launched 1088.5s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Job launched 1119.2s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Job launched 1150.0s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Job launched 1181.2s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Oops, ssh subprocess exited with return code 255, restarting...
Opening ssh tunnel to Hadoop job tracker
Connect to job tracker at: http://localhost:40737/jobtracker.jsp
Job launched 1213.0s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Unable to load progress from job tracker
Job launched 1243.8s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Job launched 1274.5s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Job launched 1305.3s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Job launched 1336.1s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Oops, ssh subprocess exited with return code 255, restarting...
Opening ssh tunnel to Hadoop job tracker
Connect to job tracker at: http://localhost:40737/jobtracker.jsp
Job launched 1367.9s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Unable to load progress from job tracker
Job launched 1398.6s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Job launched 1429.4s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Job launched 1460.2s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Job launched 1491.0s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Oops, ssh subprocess exited with return code 255, restarting...
Opening ssh tunnel to Hadoop job tracker
Connect to job tracker at: http://localhost:40737/jobtracker.jsp
Job launched 1522.9s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Unable to load progress from job tracker
Job launched 1553.8s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Job launched 1584.7s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Job launched 1615.4s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Job launched 1646.3s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
```

```
Oops, ssh subprocess exited with return code 255, restarting...
Opening ssh tunnel to Hadoop job tracker
Connect to job tracker at: http://localhost:40737/jobtracker.jsp
Job launched 1678.1s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Unable to load progress from job tracker
Job launched 1708.8s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Job launched 1739.6s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Job launched 1770.4s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Job launched 1801.1s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Oops, ssh subprocess exited with return code 255, restarting...
Opening ssh tunnel to Hadoop job tracker
Connect to job tracker at: http://localhost:40737/jobtracker.jsp
Job launched 1832.9s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Unable to load progress from job tracker
Job launched 1863.7s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Job launched 1894.4s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Job launched 1925.2s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Job launched 1955.9s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Oops, ssh subprocess exited with return code 255, restarting...
Opening ssh tunnel to Hadoop job tracker
Connect to job tracker at: http://localhost:40737/jobtracker.jsp
Job launched 1987.7s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Unable to load progress from job tracker
Job launched 2018.6s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Job launched 2049.3s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Job launched 2080.2s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Job launched 2111.0s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Oops, ssh subprocess exited with return code 255, restarting...
Opening ssh tunnel to Hadoop job tracker
Connect to job tracker at: http://localhost:40737/jobtracker.jsp
Job launched 2142.8s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Unable to load progress from job tracker
Job launched 2173.5s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Job launched 2204.3s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Job launched 2235.1s ago, status RUNNING: Running step (mrgooglegram.roo
```

```
t.20160219.092348.936845: Step 1 of 1)
Job launched 2265.9s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Oops, ssh subprocess exited with return code 255, restarting...
Opening ssh tunnel to Hadoop job tracker
Connect to job tracker at: http://localhost:40737/jobtracker.jsp
Job launched 2297.7s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Unable to load progress from job tracker
Job launched 2328.5s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Job launched 2359.2s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Job launched 2390.0s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Job launched 2420.7s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Oops, ssh subprocess exited with return code 255, restarting...
Opening ssh tunnel to Hadoop job tracker
Connect to job tracker at: http://localhost:40737/jobtracker.jsp
Job launched 2607.5s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Unable to load progress from job tracker
Job launched 2638.3s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Job launched 2669.0s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Job launched 2699.8s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Job launched 2730.6s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Oops, ssh subprocess exited with return code 255, restarting...
Opening ssh tunnel to Hadoop job tracker
Connect to job tracker at: http://localhost:40737/jobtracker.jsp
Job launched 2762.4s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Unable to load progress from job tracker
Job launched 2793.2s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Job launched 2824.0s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Job launched 2854.9s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Job launched 2885.6s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Oops, ssh subprocess exited with return code 255, restarting...
Opening ssh tunnel to Hadoop job tracker
Connect to job tracker at: http://localhost:40737/jobtracker.jsp
Job launched 2917.4s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Unable to load progress from job tracker
Job launched 2948.2s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
```

```
Job launched 2979.0s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Job launched 3009.7s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Job launched 3040.6s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Oops, ssh subprocess exited with return code 255, restarting...
Opening ssh tunnel to Hadoop job tracker
Connect to job tracker at: http://localhost:40737/jobtracker.jsp
Job launched 3072.4s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Unable to load progress from job tracker
Job launched 3103.1s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Job launched 3133.9s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Job launched 3164.7s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Job launched 3195.4s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Oops, ssh subprocess exited with return code 255, restarting...
Opening ssh tunnel to Hadoop job tracker
Connect to job tracker at: http://localhost:40737/jobtracker.jsp
Job launched 3227.3s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Unable to load progress from job tracker
Job launched 3258.1s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Job launched 3289.0s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Job launched 3319.7s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Job launched 3350.5s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Oops, ssh subprocess exited with return code 255, restarting...
Opening ssh tunnel to Hadoop job tracker
Connect to job tracker at: http://localhost:40737/jobtracker.jsp
Job launched 3382.4s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Unable to load progress from job tracker
Job launched 3413.2s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Job launched 3444.0s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Job launched 3474.9s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Job launched 3505.6s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Oops, ssh subprocess exited with return code 255, restarting...
Opening ssh tunnel to Hadoop job tracker
Connect to job tracker at: http://localhost:40737/jobtracker.jsp
Job launched 3537.5s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
```

```
Unable to load progress from job tracker
Job launched 3568.2s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Job launched 3599.0s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Job launched 3629.8s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Job launched 3660.6s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Oops, ssh subprocess exited with return code 255, restarting...
Opening ssh tunnel to Hadoop job tracker
Connect to job tracker at: http://localhost:40737/jobtracker.jsp
Job launched 3692.4s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Unable to load progress from job tracker
Job launched 3723.2s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Job launched 3753.9s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Job launched 3784.7s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Job launched 3815.4s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Oops, ssh subprocess exited with return code 255, restarting...
Opening ssh tunnel to Hadoop job tracker
Connect to job tracker at: http://localhost:40737/jobtracker.jsp
Job launched 3847.3s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Unable to load progress from job tracker
Job launched 3878.0s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Job launched 3908.7s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
Job launched 3939.5s ago, status RUNNING: Running step (mrgooglegram.roo
t.20160219.092348.936845: Step 1 of 1)
```

root@prabhakar:~/hw5# head -50 mrgooglegram-3.txt "a" 1.118421052631579 "all" 1.118421052631579 "s" 1.118421052631579 "is" 1.118421052631579 "p" 1.118421052631579 "a" 1.0429042904290429 "bibliography" 1.0429042904290429 "of" 1.0429042904290429 "united" 1.0429042904290429 "states" 1.0429042904290429 "a" 1.0 "book" 1.0 "of" 1.0 "nursery" 1.0 "logic" 1.0 "a" 1.0571428571428572 "book" 1.0571428571428572 "of" 1.0571428571428572 "roxburghe" 1.0571428571428572 "ballads" 1.0571428571428572 "a" 1.0 "brief" 1.0 "treatise" 1.0 "of" 1.0 "the" 1.0 "a" 1.04 "brief" 1.04 "and" 1.04 "true" 1.04 "narrative" 1.04 "a" 1.0 "canadian" 1.0 "by" 1.0 "birth" 1.0 "and" 1.0 "a" 1.0099009900990099 "case" 1.0099009900990099 "study" 1.0099009900990099 "of" 1.0099009900990099 "response" 1.0099009900990099 "a" 1.0 "christian" 1.0 "name" 1.0 "has" 1.0 "likewise" 1.0 "a" 1.0099009900990099 "comparison" 1.0099009900990099 "between" 1.0099009900990099 "the" 1.0099009900990099 "french" 1.0099009900990099 root@prabhakar:~/hw5#

Q.4. Distribution of 5-gram sizes (using counts info.) sorted in decreasing order of relative frequency.

[2016-02-19 02:40.32] /drives/c/berkeley/w261/assignment5/HW5-Questions [pgundugola.USLP6TV8P32] ➤ head -50 part-00000 23 247869.0 21 220842.0 17 204922.0 16 188162.0 19 156663.0 16 150516.0 17 117992.0 19 106557.0 20 105861.0 31 97419.0 19 76522.0 29 71788.0 28 68277.0 29 67945.0 23 123760.0 19 57164.0 28 54371.0 20 52951.0 23 51975.0 22 51633.0 18 50634.0 19 48545.0 20 48486.0 21 47092.0 34 45546.0 26 42429.0 21 42395.0 23 42282.0 22 161679.0 20 36339.0 17 36246.0 27 34805.0 23 32455.0 26 31508.0 23 31339.0 21 31296.0 26 30035.0 20 29626.0 23 58119.0 19 56373.0 21 27813.0 19 27470.0 20 27293.0 41 26914.0 26 26868.0 22 26328.0 18 26322.0 25 25866.0 19 25716.0 23 25479.0

In [ ]:
```
%load_ext autoreload
%autoreload 2
from mrgooglegram import MRGoogleGram

def run_google_mrjob(otype):
    output_data = []

    mr_job = MRGoogleGram(args=['filtered-5Grams\googlebooks-eng-all-5gr
am-20090715-0-filtered.txt', '--otype', str(otype)])

    with mr_job.make_runner() as runner:
        runner.run()

        for line in runner.stream_output():
            output_data.append(mr_job.parse_output_line(line))

    return output_data

run_google_mrjob('1')
```

HW 5.4. (over 2Gig of Data)

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) Build stripes for the most frequent 10,000 words using cooccurence informationa based on the words ranked from 1001,-10,000 as a basis/vocabulary (drop stopword-like terms), and output to a file in your bucket on s3 (bigram analysis, though the words are non-contiguous).

(2) Using two (symmetric) comparison methods of your choice (e.g., correlations, distances, similarities), pairwise compare all stripes (vectors), and output to a file in your bucket on s3.

==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 5-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 5-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 (2)== 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.

Build data54.txt

In [8]:
```
%%writefile preparedata54.py
#!/usr/bin/python
# preparedata54.py
# Author: Prabhakar Gundugola
# Description: Prepare the data

out = open('output/vocab.txt', 'w')
with open('output/topwords.txt', 'r') as fp:
    count = 0
    out_count = 0
    for line in fp.readlines():
        line = line.strip()
        count += 1
        if count >= 9000 and count < 10000:
            out_count += 1
            out.write(line + '\n')
print count
print out_count
out.close()
```

```
Overwriting preparedata54.py
```

In [9]:
```
!preparedata54.py
```

```
269339
1000
```

Using two (symmetric) comparison methods of your choice (e.g., correlations, distances, similarities), pairwise compare all stripes (vectors), and output to a file in your bucket on s3

In [54]:

```
%%writefile mrstripes.py
#!/usr/bin/python
# mrstripes.py
# Author: Prabhakar Gundugola
# Description: MRJob for Stripes

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

class MRStripes(MRJob):

    def configure_options(self):
        super(MRStripes, self).configure_options()
        self.add_passthrough_option('--filename', type='str')

    def load_options(self, args):
        super(MRStripes, self).load_options(args)
        self.filename = self.options.filename

    def mapper_stripe_init(self):
        self.vocabwords = set()
        self.doccount = 0
        with open(self.filename, 'r') as fp:
            for line in fp.readlines():
                tokens = line.strip().split('\t')
                for word in tokens[0].replace('"', '').split():
                    self.vocabwords.add(word)


    def mapper_stripes(self, _, line):
        tokens = line.strip().lower().split('\t')
        wordlist = sorted(list(set(tokens[0].replace('"', '').split())))
        count = int(tokens[1])
        self.doccount += count
        for i in range(len(wordlist)-1):
            stripe = {}
            if wordlist[i] in self.vocabwords:
                for j in range(i+1,len(wordlist)):
                    if wordlist[j] in self.vocabwords:
                        stripe[wordlist[j]] = count
                if len(stripe) > 0:
                    yield wordlist[i], stripe
    def mapper_stripes_final(self):
        yield '0000TOTAL', {'0000TOTAL':self.doccount}


    def combiner_stripes(self, key, values):
        stripe = {}
        for val in values:
            for word in val:
                if word in stripe:
                    stripe[word] += val[word]
                else:
```

```
                    stripe[word] = val[word]
        yield key, stripe

    def reducer_stripes_init(self):
        self.doccount = 0

    def reducer_stripes(self, key, values):
        stripe = {}
        for val in values:
            for word in val:
                if word in stripe:
                    stripe[word] += val[word]
                else:
                    stripe[word] = val[word]

        if key == '0000TOTAL':
            self.doccount = stripe[key]
        else:
            #for word in stripe:
                stripe[word] = stripe[word]*1.0/self.doccount
            yield key, stripe

    def steps(self):
        return [
            MRStep(mapper_init=self.mapper_stripe_init,
                   mapper=self.mapper_stripes,
                   mapper_final=self.mapper_stripes_final,
                   combiner=self.combiner_stripes,
                   reducer_init=self.reducer_stripes_init,
                   reducer=self.reducer_stripes,
                   jobconf={'mapred.reduce.tasks':1})
        ]

if __name__ == "__main__":
    MRStripes.run()

'''
return [
                MRStep(mapper_init=self.mapper_longest_5gram_init,
                       mapper=self.mapper_longest_5gram,
                       mapper_final=self.mapper_longest_5gram_final,
                       reducer_init=self.reducer_longest_5gram_init,
                       reducer=self.reducer_longest_5gram,
                       reducer_final=self.reducer_longest_5gram_final)
                ]
'''
```

```
Overwriting mrstripes.py
```

In [55]:
```
from mrstripes import MRStripes

def run_stripes_mrjob(vocab_file):
    output_data = []

    mr_job = MRStripes(args=['filtered-5Grams\googlebooks-eng-all-5gram-
20090715-0-filtered.txt',
                                '--filename', vocab_file, '--file', voca
b_file])

    with mr_job.make_runner() as runner:
        runner.run()

        for line in runner.stream_output():
            output_data.append(mr_job.parse_output_line(line))

    return output_data

run_stripes_mrjob('vocab.txt')
```

```
WARNING:mrjob.runner:
WARNING:mrjob.runner:PLEASE NOTE: Starting in mrjob v0.5.0, protocols w
ill be strict by default. It's recommended you run your job with --stri
ct-protocols or set up mrjob.conf as described at https://pythonhosted.
org/mrjob/whats-new.html#ready-for-strict-protocols
WARNING:mrjob.runner:
```

Out[55]:

```
[('alternate', {'viewing': 82}),
 ("alzheimer's", {'dementia': 181}),
 ('amidst', {'restless': 43, 'tumult': 80}),
 ('ammonium', {'hydroxide': 78}),
 ('anemia', {'pernicious': 58}),
 ('annum', {'thereon': 69}),
 ('approximated', {'subcutaneous': 120}),
 ('architectural', {'decoration': 46}),
 ('articular', {'cartilage': 51}),
 ('authoritative', {'interpreter': 50}),
 ('balcony', {'overlooking': 139}),
 ('bottles', {'necks': 65}),
 ('brightest', {'diamond': 71}),
 ('canons', {'commonest': 43}),
 ('careless', {'hasty': 58}),
 ('cartilage', {'localized': 51}),
 ('ce', {'qui': 48}),
 ('commence', {'palestinian': 62, 'qui': 63}),
 ('commonplace', {'feathers': 84}),
 ('complexion', {'darker': 86}),
 ('contradictory', {'predicate': 171}),
 ('conveying', {'pipes': 155}),
 ('dame', {'habitation': 66}),
 ('darkest', {'superstition': 110}),
 ('deliberations', {'trent': 47}),
 ('discomfort', {'localized': 43}),
 ('dividend', {'shareholders': 43}),
 ('dumb', {'pretending': 69}),
 ('endowment', {'unstable': 84}),
 ('est', {'qui': 83}),
 ('establishments', {'sanitary': 49}),
 ('flexor', {'sheath': 203}),
 ('flocks', {'herds': 969}),
 ('flourish', {'jungle': 137}),
 ('fossil', {'shells': 103}),
 ('hasty', {'impatience': 41}),
 ('herds', {'restless': 43}),
 ('humiliation', {'intolerable': 127, 'rejoiced': 132}),
 ('hydroxide', {'soda': 42}),
 ('indiana', {'linguistics': 128}),
 ('inspector', {'sanitary': 46}),
 ('irresistible', {'speedily': 44}),
 ('jones', {'summon': 83}),
 ('laden', {'spoils': 67}),
 ('magnificence', {'sketches': 66}),
 ('matthew', {'sayings': 76}),
 ('meridian', {'zenith': 141}),
 ('operative', {'wholesale': 43}),
 ('peritoneal', {'sac': 77}),
 ('pink', {'replacing': 47, 'wax': 411}),
 ('pitched', {'tents': 123}),
 ('polished', {'shells': 59}),
 ('relaxed', {'vigilance': 145}),
```

```
('resembling', {'shells': 107}),
('sac', {'uterine': 48}),
('telescope', {'viewing': 93})]
```

In [98]:
```
count = 0
with open('mrstripesout.txt', 'r') as fp:
    print 'Printing Top 50 lines'
    print "---------------------"
    print ""
    for line in fp.readlines():
        print line.strip()
        count += 1
        if count > 50:
            break
```

```
Printing Top 50 lines
---------------------

"abnormalities" {"mitral": 700, "plexus": 111, "schizophrenia": 198,
"esophagus": 294, "dysfunction": 45, "alzheimer's": 67, "screening":
54, "babies": 125, "congestive": 134, "lymphocytes": 58, "cleft": 21
6, "uterine": 85, "resembling": 62, "cage": 129, "placenta": 68, "cor
relate": 140, "skeletal": 466, "diabetic": 275, "lobes": 124, "unstab
le": 63, "cartilage": 92, "ovary": 126, "pathological": 76, "indicato
rs": 63}
"abyss" {"misfortunes": 113, "plunge": 1148, "damned": 53, "mind's":
236, "miseries": 44, "odor": 80, "turbulent": 115, "hangs": 69, "vomi
ting": 89, "shouts": 561, "calamity": 97, "dumb": 362, "amidst": 354,
"darkest": 68, "anarchy": 158}
"accent"        {"individuality": 94, "est": 143, "impatience": 154,
"hungarian": 44, "broadest": 185, "penetrating": 58, "spake": 180, "b
arred": 90, "displeasure": 57, "expressive": 130, "fallacy": 52}
"accepts"       {"denies": 41, "allah": 71, "repentance": 263, "offer
ings": 62, "brother's": 45, "jane": 86, "recognizes": 238}
"accompaniment" {"hymn": 231, "dishes": 66, "shouts": 353, "softly":
89, "escort": 112, "subsidiary": 59, "hymns": 333, "coronation": 42,
"oaths": 41}
"accrue"        {"thereto": 254, "shareholders": 57, "thereon": 103}
"accumulate"    {"holdings": 45, "floors": 77, "transplantation": 82,
"lymphocytes": 42, "cleft": 79, "superfluous": 43}
"accusation"    {"falsehood": 255, "centered": 48, "wholesale": 42,
"preparatory": 103, "suspicions": 63, "incur": 55, "defender": 42, "p
leaded": 50, "shrink": 74, "impending": 184}
"acetic"        {"dissociation": 467, "buffer": 76, "odor": 52, "ammo
nium": 669, "insoluble": 3271, "ether": 211}
"acknowledgments"       {"individually": 63, "funding": 56, "contribu
tors": 429, "indebtedness": 67, "funded": 754, "profoundly": 51, "avo
wed": 45, "reprint": 164, "oaths": 71}
"actuated"      {"deliberations": 59, "faction": 40, "benevolent": 4
7, "defenders": 77, "enmity": 88, "lever": 227, "ferdinand": 48, "pur
est": 1263, "irresistible": 89}
"acutely"       {"shortcomings": 127, "distressed": 66, "miseries": 6
4, "futility": 43, "congestive": 136, "habitual": 45, "infusion": 57,
"analysed": 46, "sequences": 44, "dreaded": 44, "uptake": 55}
"adduced"       {"preponderance": 132, "parallels": 137, "calamity":
41, "plaintiffs": 80, "appropriateness": 49, "attested": 142}
"administering" {"employing": 188, "enforcing": 734, "loosely": 52,
"punishments": 53, "implementing": 500, "appellate": 71, "workplace":
90, "designing": 45, "ordinances": 233, "rite": 135, "oaths": 402}
"admitting"     {"err": 85, "economists": 47, "frankly": 269, "shy":
129, "maxim": 49, "establishments": 150, "priesthood": 101, "extravag
ant": 42, "compressed": 103, "anonymous": 42, "soundness": 207, "prac
tise": 52, "shrink": 233, "humiliation": 93}
"adolescence"   {"vicissitudes": 61, "hygiene": 129, "psychoanalyti
c": 200, "transitional": 191}
"adopting"      {"humiliation": 84, "frankly": 128, "implementing": 1
21, "enforcing": 65, "speedily": 59, "charitable": 125, "stead": 150,
"terminology": 49, "consonant": 55, "proportionate": 44, "shrink": 15
```

6, "transparent": 40}
"adversely" {"disclosure": 109, "dietary": 48, "thereon": 178, "f
avorably": 94}
"advertisement" {"brand": 105, "wording": 121, "controversies": 508,
"packet": 341, "whoever": 49}
"adviser" {"confidential": 3180, "consultant": 772, "habitual":
56, "shrewd": 76, "franklin": 167, "judicious": 205, "sandy": 46}
"affiliated" {"lutheran": 578, "psychoanalytic": 75, "benevolent":
104, "assemblies": 214, "loosely": 786, "synagogue": 301, "clubs": 14
6, "broadcasting": 192, "subsidiary": 202, "officially": 48, "orien
t": 50}
"afterward" {"sticking": 50, "unconstitutional": 44, "bottles": 7
0, "confided": 44, "frankly": 197, "tents": 64, "relaxed": 170, "main
tains": 54, "inherit": 72, "summon": 41, "bade": 123, "prosecuted": 1
22, "cleaned": 189, "dishes": 57, "damp": 55, "merged": 213, "vogue":
54}
"aggravated" {"misfortunes": 199, "discontent": 225, "amidst": 59,
"inflict": 78, "miseries": 147, "bitterly": 53, "jar": 72, "terrors":
96, "deficiencies": 50, "suspicions": 98, "displeasure": 237, "aversi
on": 82, "darker": 68, "wording": 102}
"agrarian" {"discontent": 50, "madras": 119, "indebtedness": 58,
"insists": 207, "maxim": 51, "myths": 74, "restructuring": 481, "trib
une": 62, "sweeping": 51, "programmes": 57}
"alaska" {"orient": 147, "purchases": 50, "recreation": 79, "e
astward": 52, "ceded": 107, "americas": 144, "steamer": 72, "fog": 4
6, "arkansas": 567, "nova": 73, "fossil": 65, "gravel": 200, "touris
m": 144, "southeastern": 1513, "sack": 58}
"allah" {"believeth": 261, "whoever": 185, "believers": 72, "accept
s": 71, "wills": 72, "repentance": 191, "jar": 47, "amidst": 57, "sho
uts": 222, "bade": 71, "merciful": 386, "bounty": 265, "conqueror": 8
1}
"allude" {"uppermost": 58, "localities": 40, "disproportionat
e": 73, "articulate": 104}
"alpha" {"penetrating": 327, "radioactive": 360, "energetic": 139, "s
ecreted": 596, "alphabet": 167, "necrosis": 6668, "sequences": 47, "b
locked": 51, "dysfunction": 103}
"alphabet" {"revealing": 88, "mystical": 41, "backwards": 200,
"hasty": 51, "rudiments": 182, "imagery": 51, "mastered": 89, "fier
y": 77, "multiplication": 463, "alpha": 167, "spelling": 280, "invent
or": 500}
"altering" {"ship's": 101, "pyramid": 53, "purport": 45, "renoun
ce": 64, "demographic": 78, "elegance": 83, "profoundly": 63, "comple
xion": 59, "morphology": 94, "spelling": 117}
"alternate" {"barred": 44, "contending": 62, "predominance": 44,
"thereafter": 65, "episodes": 99, "floors": 46, "flush": 114, "hymn":
141, "impatience": 68, "factions": 62, "uric": 51, "gravel": 41, "exh
austion": 54, "turmoil": 70, "amiable": 53, "downs": 42, "frost": 40,
"viewing": 162, "preponderance": 52, "threads": 401, "strenuous": 92,
"transparent": 193, "backwards": 54, "vicissitudes": 40, "pores": 72,
"dilatation": 351, "sadness": 199, "victories": 637, "sensibility": 5
9}
"alternating" {"remission": 108, "uric": 68, "surround": 66, "episo
des": 166, "amidst": 50, "masculine": 192, "phosphorus": 51}

"alveolar" {"secreted": 132, "wasting": 58, "maxillary": 841, "a
rticulation": 89, "eruption": 74, "residual": 418, "pores": 58, "peri
toneal": 94, "necrosis": 740, "arches": 264, "resembling": 51, "regen
eration": 68, "rabbit": 192, "disruption": 62, "atrophy": 503, "clef
t": 543, "canals": 58, "diffuse": 491}
"alzheimer's" {"rating": 245, "paradigm": 60, "secreted": 62, "prec
ursor": 1024, "ganglion": 99, "schizophrenia": 131, "counseling": 49,
"atrophy": 490, "abnormalities": 67, "dementia": 5606, "pathologica
l": 64, "dysfunction": 44}
"amazement" {"permitting": 65, "spectators": 66, "solemnity": 12
2, "distressed": 46, "scarlet": 120, "dumb": 1795, "magnificence": 4
9}
"americas" {"voyages": 43, "cognizance": 64, "warmer": 69, "alas
ka": 144, "colonization": 156}
"amiable" {"misfortunes": 61, "discontent": 42, "polished": 85,
"captive": 139, "athenians": 44, "graceful": 58, "nephew": 114, "huma
ne": 174, "purest": 121, "clergyman": 162, "alternate": 53, "eleganc
e": 42, "speedily": 47, "imputed": 97, "atonement": 40, "practitione
r": 43, "neighbour": 143, "judicious": 43, "benevolent": 212, "harmon
ious": 51, "franklin": 114, "deserving": 155, "natures": 81, "sensibi
lity": 185, "countess": 76}
"amidst" {"misfortunes": 288, "individuality": 48, "contendin
g": 283, "restless": 43, "pleases": 85, "allah": 57, "turbulent": 46,
"pursuits": 43, "shouts": 3029, "slender": 77, "hymns": 395, "provok
e": 42, "rushing": 59, "orient": 49, "fortified": 50, "frost": 114,
"abyss": 354, "effectually": 44, "roses": 78, "anarchy": 187, "foe":
54, "faded": 73, "mutations": 57, "disclosed": 63, "fiery": 102, "jud
icious": 52, "gravel": 52, "superfluous": 59, "meridian": 43, "deligh
ts": 54, "alternating": 50, "turmoil": 973, "plunge": 182, "energeti
c": 48, "habitual": 59, "downs": 59, "repentance": 51, "kindred": 13
3, "terrors": 696, "expanse": 52, "recollections": 63, "magnificenc
e": 88, "flourish": 254, "imperfections": 48, "vicissitudes": 370, "a
ggravated": 59, "balls": 96, "solemnity": 73, "tumult": 562, "shell
s": 43, "surround": 77, "miseries": 151, "barren": 49, "excesses": 10
8, "stationary": 140}
"ammonium" {"hydroxide": 13626, "dissociation": 137, "buffer": 4
4, "uptake": 110, "uric": 111, "acetic": 669, "soda": 45, "sulphate":
8735, "filtered": 54, "purified": 476, "crystalline": 124, "insolubl
e": 338, "phosphorus": 50}
"amplifier" {"buffer": 335, "terminals": 998, "laser": 103, "ce":
70}
"amply" {"revealing": 69, "gratify": 46, "practitioner": 100, "fortif
ied": 51, "sequel": 51, "testified": 576, "speedily": 71, "repaid": 1
2798, "lacked": 40, "attested": 420}
"amuse" {"gratify": 53, "employing": 44, "spectators": 238, "architec
tural": 52, "flatter": 161, "pleases": 125, "narrower": 70, "viewin
g": 48, "captive": 41, "thoughtful": 50, "careless": 41, "traveller
s": 54, "scandal": 42, "nephew": 44}
"analysed" {"acutely": 46}
"analyzing" {"shortcomings": 59, "monitored": 105, "groundwork":
47, "discerned": 43, "predicting": 844, "explanatory": 81, "viewing":
60, "episodes": 98, "designing": 46, "domains": 156, "paradigm": 41}
"anarchy" {"misfortunes": 43, "abyss": 158, "plunge": 382, "fac

```
tion": 112, "dreaded": 69, "futility": 2073, "whither": 48, "amidst":
187, "matthew": 1036, "miseries": 114, "resembling": 93, "intolerabl
e": 71, "purest": 104}
"anemia"        {"postoperative": 62, "arthritis": 69, "regeneratio
n": 98, "pernicious": 18534, "commonest": 74, "autosomal": 63, "trans
plantation": 372, "mutations": 51, "congestive": 47, "resembling": 13
4}
"anesthesia"    {"monitored": 212, "uterine": 158, "postoperative": 2
84, "plexus": 236, "necrosis": 170, "potentials": 66, "vomiting": 33
4, "prolong": 166, "bedside": 92, "skeletal": 165, "peritoneal": 48,
"ether": 435, "operative": 99, "dysfunction": 124}
"anglican"      {"lutheran": 559, "sympathies": 86, "clergyman": 103
8, "defenders": 118, "kindred": 107, "synod": 392, "bade": 44, "archi
tectural": 551, "rite": 112, "priesthood": 279}
"annoyed"       {"pretending": 150, "intrusion": 218, "forts": 51, "d
istressed": 89, "enforcing": 51, "ridicule": 71, "extravagant": 48,
"travellers": 85, "honors": 48}
"annum" {"sterling": 3211, "dividend": 85, "thereon": 114, "compounde
d": 714, "crowns": 76}
"anonymous"     {"troublesome": 48, "contributors": 67, "confidentia
l": 153, "performances": 89, "victories": 108, "admitting": 42, "capt
ains": 45, "painters": 68, "appellate": 53, "babies": 51, "conquero
r": 111, "authorship": 93, "defender": 46}
```

In [87]:

```
%%writefile test.txt
'docA'  {'X':20,'Y':30,'Z':5}
'docB'  {'X':100,'Y':20}
'docC'  {'M':5,'N':20,'Z':5}
```

```
Overwriting test.txt
```

In [92]:

```
%%writefile mrcomparator.py
#!/usr/bin/python
# mrcomparator.py
# Author: Prabhakar Gundugola
# Description: Code to determine Inverted index and Cosine

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

from ast import literal_eval
import math

class MRComparator(MRJob):
    def configure_options(self):
        super(MRComparator, self).configure_options()
        self.add_passthrough_option('--comtype', type='str', default='in
verted')

    def load_options(self, args):
        super(MRComparator, self).load_options(args)
        self.comtype = self.options.comtype

    '''
    Calculating inverted index
    '''
    def mapper_inverted_init(self):
        self.total = 0

    def mapper_inverted(self, _, line):
        line = line.strip().split('\t')
        doc, words = literal_eval(line[0]), literal_eval(line[1])
        stripe = {}
        for word in words.iteritems():
            self.total += int(word[1])
            stripe[doc] = int(word[1])
            yield word[0], stripe

    def mapper_inverted_final(self):
        stripe = {}
        stripe['0000TOTAL'] = self.total
        yield '0000TOTAL', stripe

    def reducer_inverted_init(self):
        self.total = 0

    def reducer_inverted(self, key, stripes):

        if key == '0000TOTAL':
            for s in stripes:
                self.total += s[key]
        else:
            stripe = {}
            count = 0
```

```
        for s in stripes:
            for word in s:
                count += s[word]
                if word in stripe:
                    stripe[word] += s[word]
                else:
                    stripe[word] = s[word]

        inverted_index = 1.0*count/self.total
        yield key, inverted_index

    '''
    Calculating Cosine
    '''
    def mapper_cosine(self, _, line):
        line = line.strip().split('\t')
        doc, words = literal_eval(line[0]), literal_eval(line[1])
        total_square = 0
        for word in words.iteritems():
            total_square += int(word[1])*int(word[1])

        for word in words.iteritems():
            cos_product = 1.0*int(word[1])/math.sqrt(total_square)
            yield word[0], (doc, cos_product)

    def reducer_cosine(self, word, stripes):
        stripe = {}
        for doc, product in stripes:
            stripe[doc] = product
        yield word, stripe

    def mapper_cosine_product(self, word, stripes):
        docs = sorted(stripes.keys())

        for doc1 in docs:
            for doc2 in docs:
                if doc1 != doc2:
                    yield (doc1, doc2), stripes[doc1]*stripes[doc2]

    def reducer_cosine_product(self, pair, cosine_product):
        yield pair, sum(cosine_product)


    def steps(self):
        if self.comtype == 'inverted':
            return [
                MRStep(mapper_init=self.mapper_inverted_init,
                       mapper=self.mapper_inverted,
                       mapper_final=self.mapper_inverted_final,
                       reducer_init=self.reducer_inverted_init,
                       reducer=self.reducer_inverted,
                       jobconf={'mapred.reduce.task':1})
            ]
```

```
        elif self.comtype == 'cosine':
            return [
                MRStep(mapper=self.mapper_cosine,
                       reducer=self.reducer_cosine),
                MRStep(mapper=self.mapper_cosine_product,
                       reducer=self.reducer_cosine_product,
                       jobconf={'mapred.reduce.tasks':1})
            ]

if __name__ == "__main__":
    MRComparator.run()
```

```
Overwriting mrcomparator.py
```

In [93]:
```
%load_ext autoreload
%autoreload 2
from mrcomparator import MRComparator

def run_comparator_mrjob(comtype):
    output_data = []

    mr_job = MRComparator(args=['test.txt', '--comtype', comtype])

    with mr_job.make_runner() as runner:
        runner.run()

        for line in runner.stream_output():
            output_data.append(mr_job.parse_output_line(line))

    return output_data
print ""
print ""
print 'Cosine similarity of docA, docB, docC are: '
run_comparator_mrjob('cosine')
```

```
WARNING:mrjob.runner:
WARNING:mrjob.runner:PLEASE NOTE: Starting in mrjob v0.5.0, protocols w
ill be strict by default. It's recommended you run your job with --stri
ct-protocols or set up mrjob.conf as described at https://pythonhosted.
org/mrjob/whats-new.html#ready-for-strict-protocols
WARNING:mrjob.runner:

The autoreload extension is already loaded. To reload it, use:
  %reload_ext autoreload


Cosine similarity of docA, docB, docC are:
```

Out[93]:
```
[(['docA', 'docB'], 0.7004041959724748),
 (['docA', 'docC'], 0.0323761954119088),
 (['docB', 'docA'], 0.7004041959724748),
 (['docC', 'docA'], 0.0323761954119088)]
```

In [94]:
```
%load_ext autoreload
%autoreload 2
from mrcomparator import MRComparator

def run_comparator_mrjob(comtype):
    output_data = []

    mr_job = MRComparator(args=['test.txt', '--comtype', comtype])

    with mr_job.make_runner() as runner:
        runner.run()

        for line in runner.stream_output():
            output_data.append(mr_job.parse_output_line(line))

    return output_data
print ""
print ""
print 'Inverted index of words - M, N, X, Y, Z in docA, docB, docC are:
'
run_comparator_mrjob('inverted')
```

```
WARNING:mrjob.runner:
WARNING:mrjob.runner:PLEASE NOTE: Starting in mrjob v0.5.0, protocols w
ill be strict by default. It's recommended you run your job with --stri
ct-protocols or set up mrjob.conf as described at https://pythonhosted.
org/mrjob/whats-new.html#ready-for-strict-protocols
WARNING:mrjob.runner:

The autoreload extension is already loaded. To reload it, use:
  %reload_ext autoreload


Inverted index of words - M, N, X, Y, Z in docA, docB, docC are:
```

Out[94]:
```
[('M', 0.024390243902439025),
 ('N', 0.0975609756097561),
 ('X', 0.5853658536585366),
 ('Y', 0.24390243902439024),
 ('Z', 0.04878048780487805)]
```

In [102]:

```
%%writefile synonym_5_4.py
#!/usr/bin/env python

from mrjob.job import MRJob
from mrjob.step import MRJobStep
import ast
import math

class Similarity(MRJob):
    def mapper(self, _, line):
        # read streaming word occurrence input and evaluate stripes
        stripes = line.strip().split('\t')

        word = stripes[0].strip('"')
        stripe_pairs = ast.literal_eval(stripes[1])
        xx = sum(stripe_pairs.get(k, 0)**2 for k in stripe_pairs.keys())


        # read second instance of word cooccurrences broadcasted to the
mappers
        for line in open('word_stripes.txt', 'r').read().strip().split
('\n'):
            # parse word and stripes
            l_stripes = line.split('\t')

            l_word = l_stripes[0].strip('"')
            l_stripe_pairs = ast.literal_eval(l_stripes[1])

            # avoid symmetric calculations
            # (a, b) = (b, a)
            if word > l_word:
                continue

            # form combined keys to check distance each word
            combined_words = set(l_stripe_pairs.keys()).union(set(stripe
_pairs.keys()))

            # jaccard similarity
            jaccard_similarity = 0.0
            intersection = len(set.intersection(*[set(l_stripe_pairs.key
s()), set(stripe_pairs.keys())]))
            union = len(combined_words)
            jaccard_similarity = intersection*1.0/union

            # cosine similarity
            cosine_similarity = 0.0
            # norm X
            xx = sum(stripe_pairs.get(k, 0)**2 for k in combined_words)
            # norm Y
            yy = sum(l_stripe_pairs.get(k, 0)**2 for k in combined_word
s)
            # norm XY
```

```
            xy = sum(stripe_pairs.get(k, 0)*l_stripe_pairs.get(k, 0) for
k in combined_words)
            cosine_similarity = xy/math.sqrt(xx*yy)

            # emit word and similarity metrics
            yield ((word, l_word), (jaccard_similarity,cosine_similarit
y))


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

```
Writing synonym_5_4.py
```

In [104]:
```
!synonym_5_4.py mrstripesout.txt -q -r local --file word_stripes.txt --n
o-strict-protocol > word_similarity.txt
```

HW 5.5

In this part of the assignment you will evaluate the success of you synonym detector. Take the top 1,000 closest/most similar/correlative pairs of words as determined by your measure in (2), and use the synonyms function in the accompanying python code:

nltk_synonyms.py

Note: This will require installing the python nltk package:

http://www.nltk.org/install.html (http://www.nltk.org/install.html)

and downloading its data with nltk.download().

For each (word1,word2) pair, check to see if word1 is in the list, synonyms(word2), and vice-versa. If one of the two is a synonym of the other, then consider this pair a 'hit', and then report the precision, recall, and F1 measure of your detector across your 1,000 best guesses. Report the macro averages of these measures.

In [101]:
```
#!/usr/bin/python2.7
''' pass a string to this funciton ( eg 'car') and it will give you a li
st of
words which is related to cat, called lemma of CAT. '''
import nltk
from nltk.corpus import wordnet as wn
import sys
#print all the synset element of an element

def synonyms(string):
    syndict = {}
    for i,j in enumerate(wn.synsets(string)):
        syns = j.lemma_names()
        for syn in syns:
            syndict.setdefault(syn,1)
    return syndict.keys()

hit_count = 0
total_count = 0
for line in open('word_similarity.txt', 'r').read().strip().split('\n'):
    word_pair = line.strip().split('\t')[0]
    words = word_pair.strip('[]').split(',')
    synonyms_list = synonyms(words[0].strip('"'))
    total_count += 1
    if words[1].strip(' "') in synonyms_list:
        hit_count +=1

precision = hit_count*1.0/total_count
print precision

# Recall - you have to go through all similarity pairs (not only the top
1000), and count the total pairs that match synonyms result.
# Recall = hit_count*1.0/total pairs match synonyms result from step abo
ve.
# F1 = 2*(precision*recall)/(precision+recall)
# macro-average precision = (P1+P2)/2
# macro-average recall = (R1+R2)/2
```

```
0.0412262156448
```