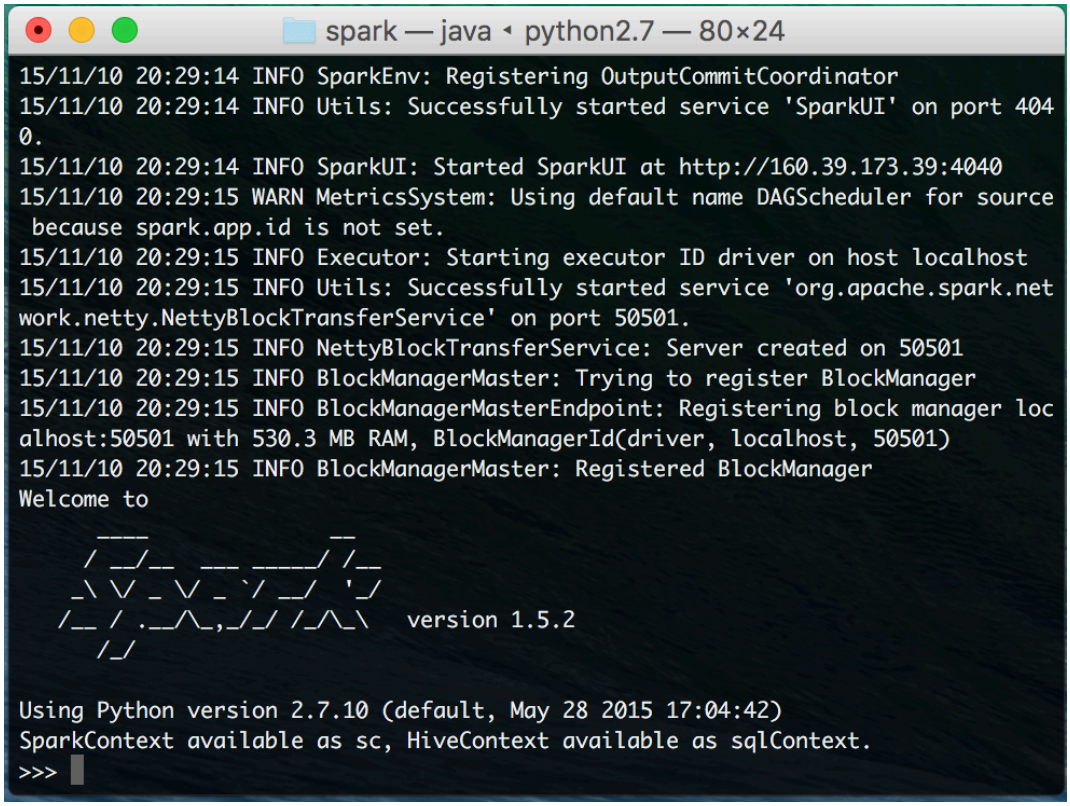
EECS E6893 Homework 3

Zhuxi Cai zc2270

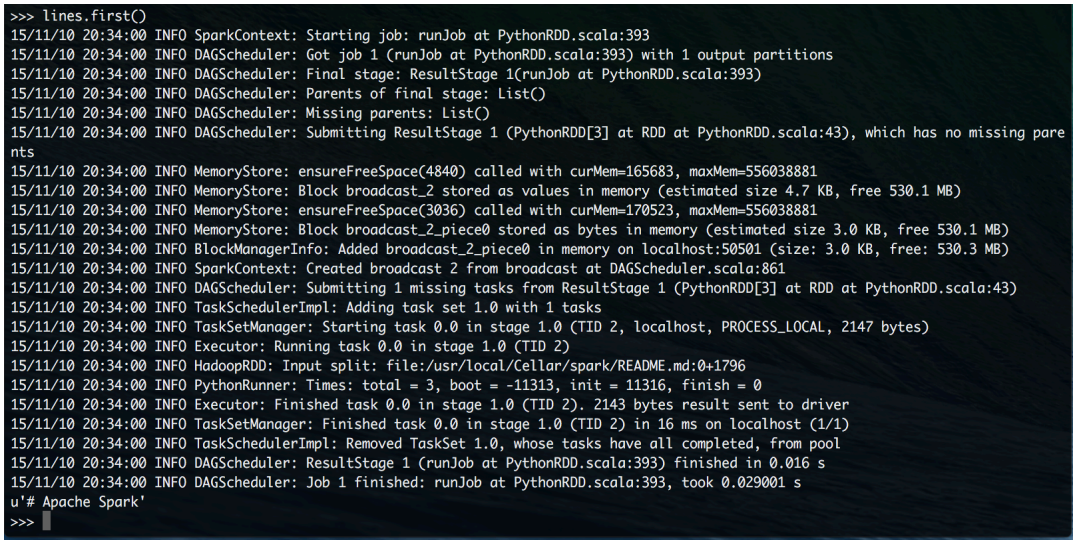
1. Spark:

1.1: Install and test Spark

We download Spark from the website and call PySpark Shell:



We test Spark by counting the lines in PySpark:

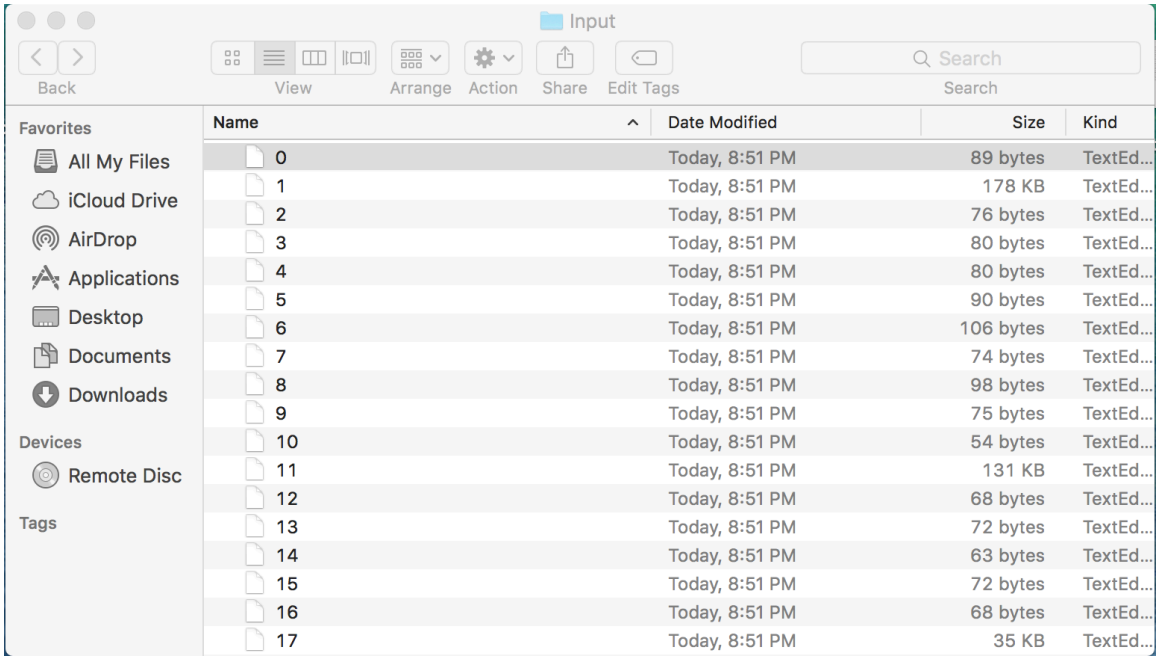


```
>>> lines.count()
15/11/10 20:33:47 INFO FileInputFormat: Total input paths to process : 1
15/11/10 20:33:47 INFO SparkContext: Starting job: count at <stdin>:1
15/11/10 20:33:47 INFO DAGScheduler: Got job 0 (count at <stdin>:1) with 2 output partitions
15/11/10 20:33:47 INFO DAGScheduler: Final stage: ResultStage 0(count at <stdin>:1)
15/11/10 20:33:47 INFO DAGScheduler: Parents of final stage: List()
15/11/10 20:33:47 INFO DAGScheduler: Missing parents: List()
15/11/10 20:33:47 INFO DAGScheduler: Submitting ResultStage 0 (PythonRDD[2] at count at <stdin>:1), which has no missing parents
15/11/10 20:33:47 INFO MemoryStore: ensureFreeSpace(5664) called with curMem=156513, maxMem=556038881
15/11/10 20:33:47 INFO MemoryStore: Block broadcast_1 stored as values in memory (estimated size 5.5 KB, free 530.1 MB)
15/11/10 20:33:47 INFO MemoryStore: ensureFreeSpace(3506) called with curMem=162177, maxMem=556038881
15/11/10 20:33:47 INFO MemoryStore: Block broadcast_1_piece0 stored as bytes in memory (estimated size 3.4 KB, free 530.1 MB)
15/11/10 20:33:47 INFO BlockManagerInfo: Added broadcast_1_piece0 in memory on localhost:50501 (size: 3.4 KB, free: 530.3 MB)
15/11/10 20:33:47 INFO SparkContext: Created broadcast 1 from broadcast at DAGScheduler.scala:861
15/11/10 20:33:47 INFO DAGScheduler: Submitting 2 missing tasks from ResultStage 0 (PythonRDD[2] at count at <stdin>:1)
15/11/10 20:33:47 INFO TaskSchedulerImpl: Adding task set 0.0 with 2 tasks
15/11/10 20:33:47 INFO TaskSetManager: Starting task 0.0 in stage 0.0 (TID 0, localhost, PROCESS_LOCAL, 2147 bytes)
15/11/10 20:33:47 INFO TaskSetManager: Starting task 1.0 in stage 0.0 (TID 1, localhost, PROCESS_LOCAL, 2147 bytes)
15/11/10 20:33:47 INFO Executor: Running task 0.0 in stage 0.0 (TID 0)
15/11/10 20:33:47 INFO Executor: Running task 1.0 in stage 0.0 (TID 1)
15/11/10 20:33:47 INFO HadoopRDD: Input split: file:/usr/local/Cellar/spark/README.md:0+1796
15/11/10 20:33:47 INFO HadoopRDD: Input split: file:/usr/local/Cellar/spark/README.md:1796+1797
15/11/10 20:33:47 INFO deprecation: mapred.tip.id is deprecated. Instead, use mapreduce.task.id
15/11/10 20:33:47 INFO deprecation: mapred.task.id is deprecated. Instead, use mapreduce.task.attempt.id
15/11/10 20:33:47 INFO deprecation: mapred.task.is.map is deprecated. Instead, use mapreduce.task.ismap
15/11/10 20:33:47 INFO deprecation: mapred.task.partition is deprecated. Instead, use mapreduce.task.partition
15/11/10 20:33:47 INFO deprecation: mapred.job.id is deprecated. Instead, use mapreduce.job.id
15/11/10 20:33:48 INFO PythonRunner: Times: total = 918, boot = 902, init = 15, finish = 1
15/11/10 20:33:48 INFO PythonRunner: Times: total = 919, boot = 907, init = 11, finish = 1
15/11/10 20:33:48 INFO Executor: Finished task 1.0 in stage 0.0 (TID 1). 2124 bytes result sent to driver
15/11/10 20:33:48 INFO Executor: Finished task 0.0 in stage 0.0 (TID 0). 2124 bytes result sent to driver
15/11/10 20:33:48 INFO TaskSetManager: Finished task 0.0 in stage 0.0 (TID 0) in 1066 ms on localhost (1/2)
15/11/10 20:33:48 INFO TaskSetManager: Finished task 1.0 in stage 0.0 (TID 1) in 1047 ms on localhost (2/2)
15/11/10 20:33:48 INFO TaskSchedulerImpl: Removed TaskSet 0.0, whose tasks have all completed, from pool
15/11/10 20:33:48 INFO DAGScheduler: ResultStage 0 (count at <stdin>:1) finished in 1.088 s
15/11/10 20:33:48 INFO DAGScheduler: Job 0 finished: count at <stdin>:1, took 1.188206 s
98
```

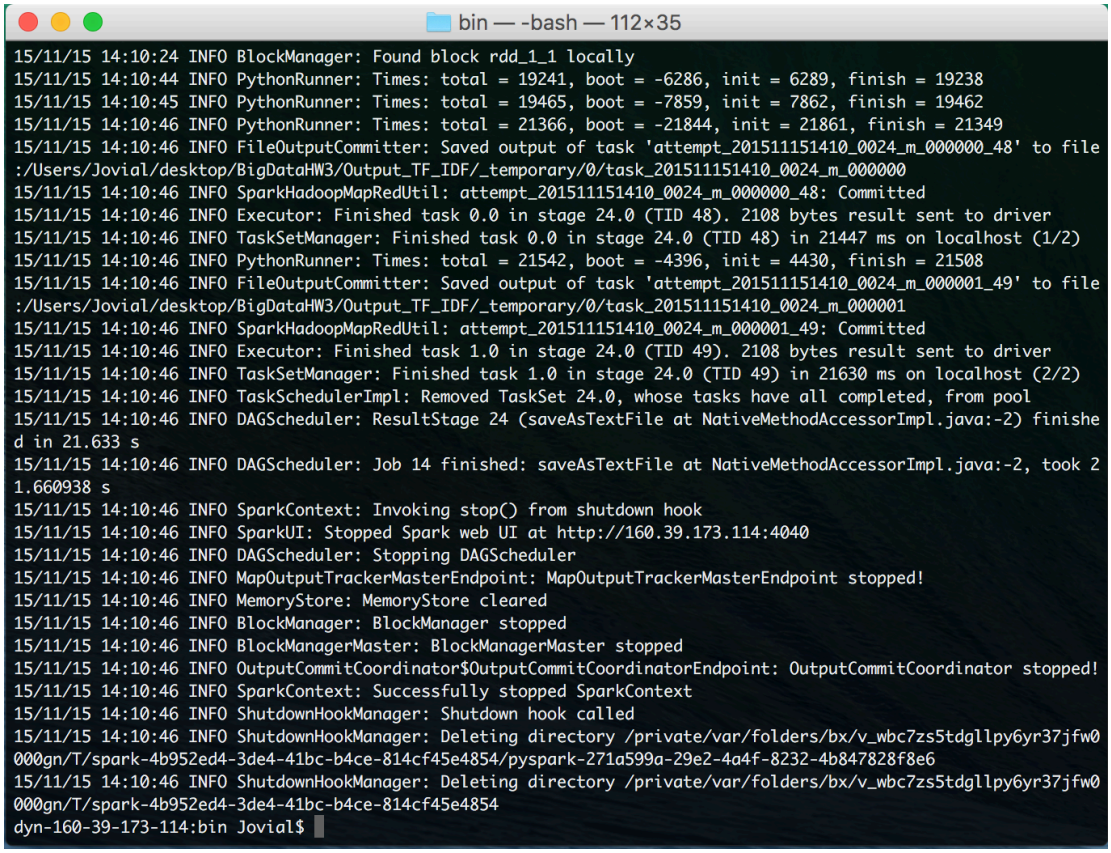
1.2: Use the Wikipedia data you downloaded in Homework 2. Calculate TF-IDF. Do clustering. Discuss the outcome, comparing with what you did from Mahout.

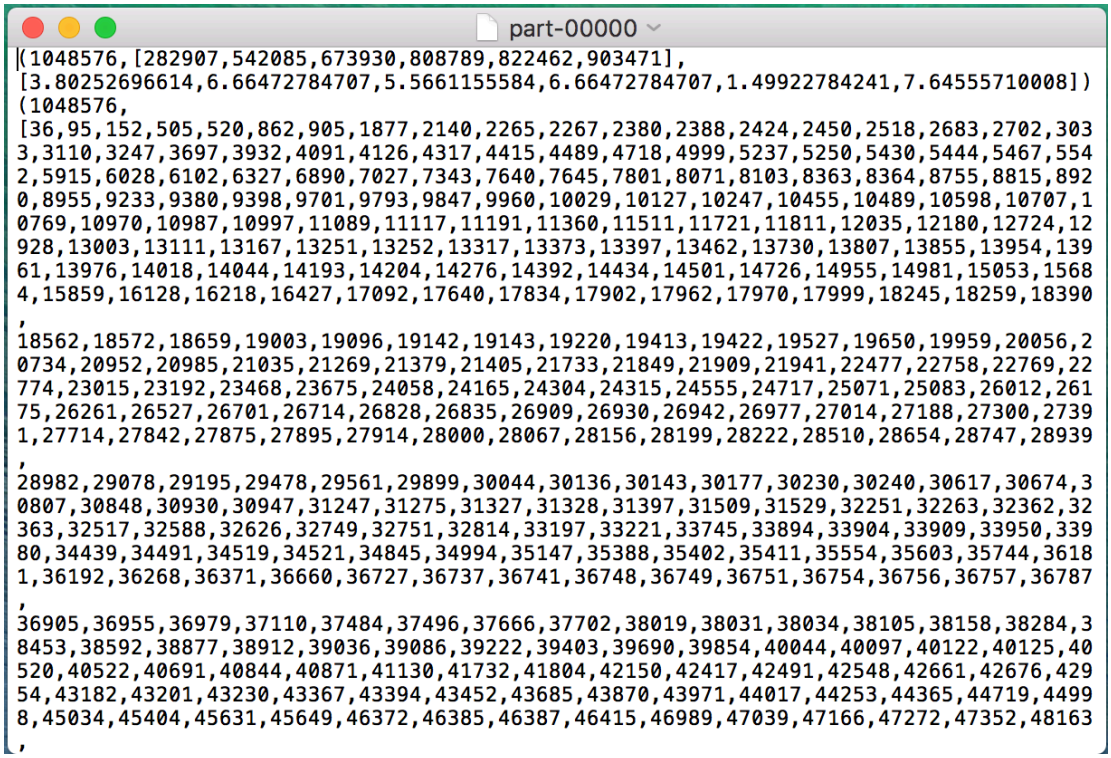
Here we still use the Wikipedia data I downloaded in Homework 2: enwiki-20150901-pages-articles1.xml-p000000010p000010000.

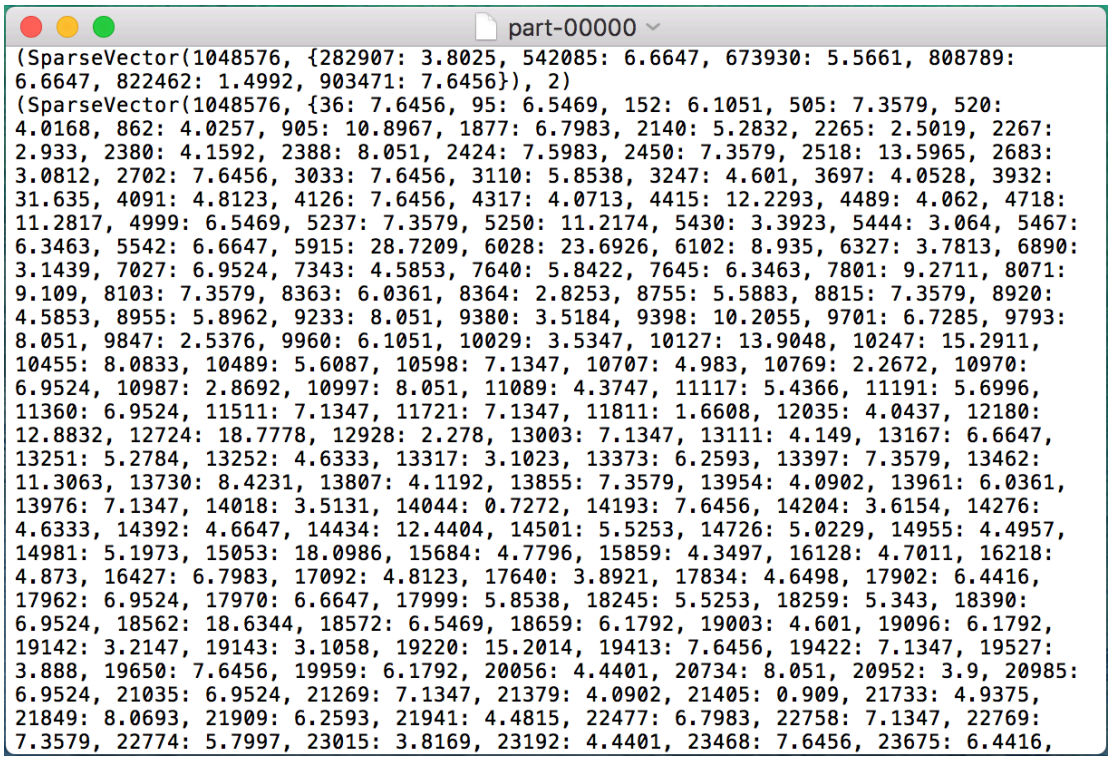
First I extract text file from Wikipedia .xml dataset and store them in file 'Input':



Then I calculate TF-IDF and do clustering:





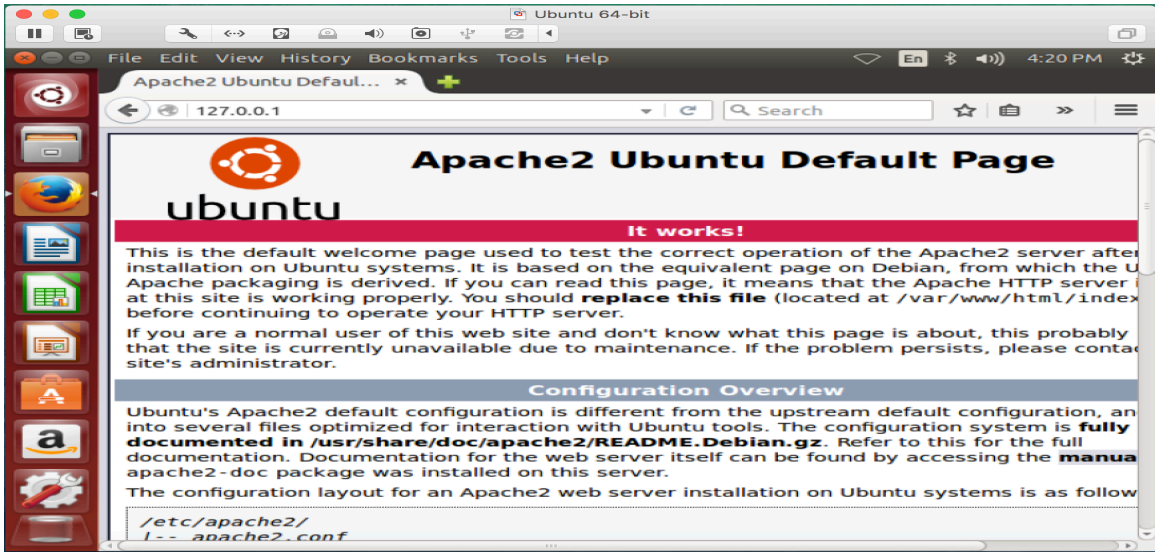


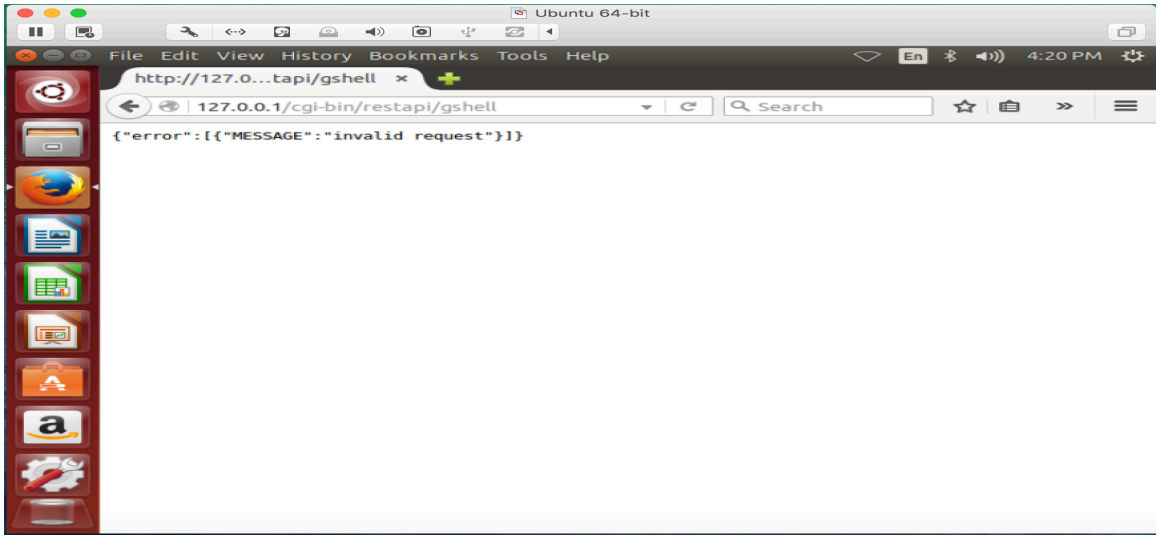
By comparing with what I did from Mahout, the result and accuracy are quite similar here. However, the efficiency of Spark MLlib here is much lower than Mahout. According to my Google research, one data scientist named Dmitriy Lyubimov has given us the answer that Mahout is not necessarily moving to be primarily a collection things like MLlib does. So Mahout can enable one to prototype his own distributed data instead of the whole data, which can explain the low efficiency of MLlib here.

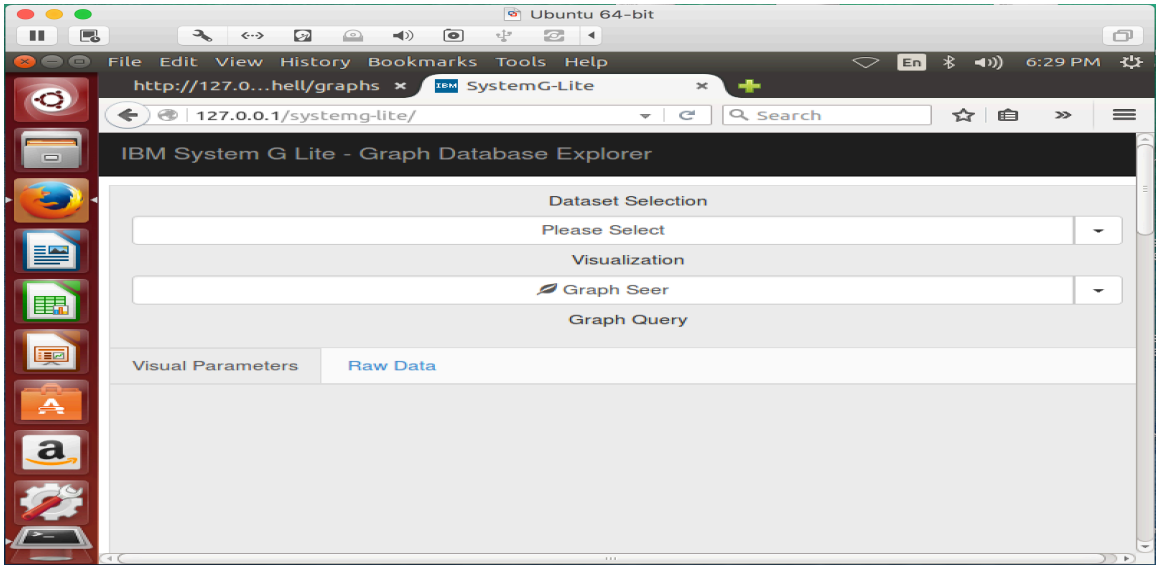
2. Graph Database:

2.1: Download IBM System G Graph Tools

After exhausting efforts, finally I finished the installation and configuration of IBM System G Graph Tools in Ubuntu LTS 14.04:


Ubuntu 64-bit
File Edit View History Bookmarks Tools Help
4:20 PM
Apache2 Ubuntu Defaul...
127.0.0.1
Search
ubuntu
Apache2 Ubuntu Default Page
It works!
This is the default welcome page used to test the correct operation of the Apache2 server after
installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the U
Apache packaging is derived. If you can read this page, it means that the Apache HTTP server
at this site is working properly. You should replace this file (located at /var/www/html/index
before continuing to operate your HTTP server.
If you are a normal user of this web site and don't know what this page is about, this probably
that the site is currently unavailable due to maintenance. If the problem persists, please conta
site's administrator.
Configuration Overview
Ubuntu's Apache2 default configuration is different from the upstream default configuration, an
into several files optimized for interaction with Ubuntu tools. The configuration system is fully
documented in /usr/share/doc/apache2/README.Debian.gz. Refer to this for the full
documentation. Documentation for the web server itself can be found by accessing the manua
apache2-doc package was installed on this server.
The configuration layout for an Apache2 web server installation on Ubuntu systems is as follow
/etc/apache2/
|-- apache2.conf


Ubuntu 64-bit
File Edit View History Bookmarks Tools Help
4:20 PM
http://127.0...tapi/gshell
127.0.0.1/cgi-bin/restapi/gshell
Search
{"error":[{"MESSAGE":"invalid request"}]}


Ubuntu 64-bit
File Edit View History Bookmarks Tools Help
6:29 PM
http://127.0...hell/graphs
SystemG-Lite
127.0.0.1/systemg-lite/
Search
IBM System G Lite - Graph Database Explorer
Dataset Selection
Please Select
Visualization
Graph Seer
Graph Query
Visual Parameters
Raw Data

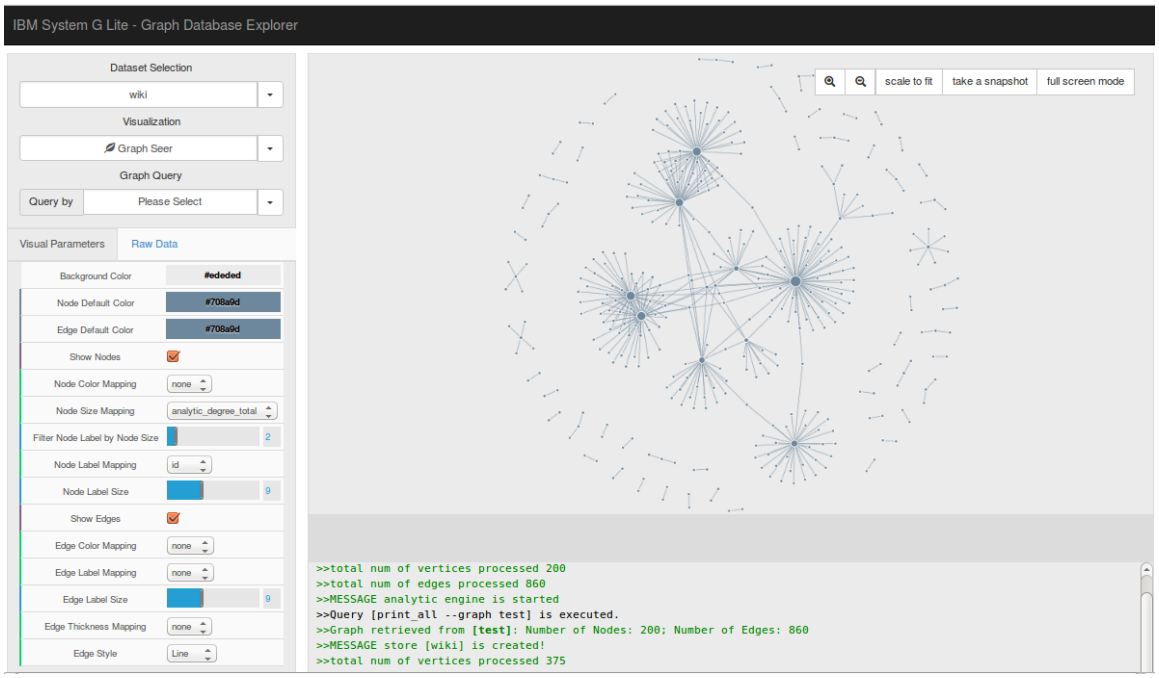
2.2: Download URL links in the Wikipedia data. Create a knowledge graph. Inject it into a graph database. Try some queries to find relevant terms. This can serve as keyword expansion. Show some visualizations of your queries.

Here I used 100 Wikipedia URL links and converted them into node.csv and edge.csv as following:

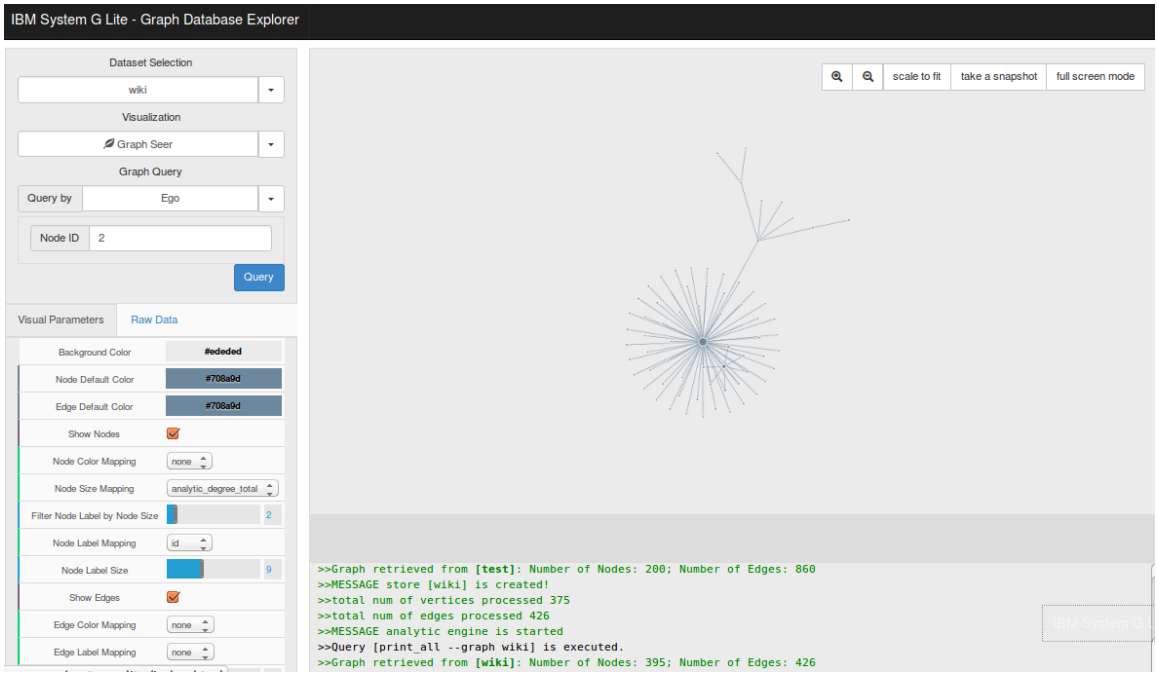
id	title
1	!
2	!!
3	!!!
4	!!!!
5	!!!Fuck_You!!!
6	!!!Fuck_You!!!_And_Then_Some
7	!!!Fuck_You!!!_And_then_Some
8	!!!Fuck_You!!!_and_Then_Some
9	!!!_(album)
10	!!!_(band)
11	!!Destroy-Oh-Boy!!
12	!!Fuck_you!!
13	!!M
14	!!Que_Corra_La_Voz!!
15	!!_(chess)
16	!!m
17	!'O-!khung_language
18	!=
19	!?
20	!?!
22	!?_(chess)
23	!A_Luchar!
24	!Action_Pact!
25	!Adios_Amigos!
26	!Alabadle!
27	!Alarma!
28	!Alarma!_(album)
29	!Alarma!_(disambiguation)
30	!Alarma!_(magazine)
31	!Alarma!_Records
32	!Alarma!_magazine
33	!Alfaro_Vive
34	!All-Time_Quarterback!
35	!All-Time_Quarterback!_(EP)
36	!All-Time_Quarterback!_(album)
37	!Alla_tu!

source	target	value
1	1664968	1
2	3	1
2	747213	1
2	1664968	1
2	1691047	1
2	4095634	1
2	5535664	1
3	9	1
3	77935	1
3	79583	1
3	84707	1
3	564578	1
3	594898	1
3	681805	1
3	681886	1
3	835470	1
3	880698	1
3	1109091	1
3	1125108	1
3	1279972	1
3	1463445	1
3	1497566	1
3	1783284	1
3	1997564	1
3	2006526	1
3	2070954	1
3	2250217	1
3	2268713	1
3	2276203	1
3	2374802	1
3	2571397	1
3	2640902	1
3	2647217	1
3	2732378	1
3	2821237	1
3	3088028	1

Then I uploaded these files into IBM System G Graph Tools and get the following knowledge graph:



Finally I tried two different queries to get keyword expansion, first all the links with node number 2:



Second the analytics_degree equals to 5: