

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:

```
spark — java ▾ python2.7 — 80x24
15/11/10 20:29:14 INFO SparkEnv: Registering OutputCommitCoordinator
15/11/10 20:29:14 INFO Utils: Successfully started service 'SparkUI' on port 4040.
15/11/10 20:29:14 INFO SparkUI: Started SparkUI at http://160.39.173.39:4040
15/11/10 20:29:15 WARN MetricsSystem: Using default name DAGScheduler for source
because spark.app.id is not set.
15/11/10 20:29:15 INFO Executor: Starting executor ID driver on host localhost
15/11/10 20:29:15 INFO Utils: Successfully started service 'org.apache.spark.net
work.netty.NettyBlockTransferService' on port 50501.
15/11/10 20:29:15 INFO NettyBlockTransferService: Server created on 50501
15/11/10 20:29:15 INFO BlockManagerMaster: Trying to register BlockManager
15/11/10 20:29:15 INFO BlockManagerMasterEndpoint: Registering block manager loc
alhost:50501 with 530.3 MB RAM, BlockManagerId(driver, localhost, 50501)
15/11/10 20:29:15 INFO BlockManagerMaster: Registered BlockManager
Welcome to

   ___
  / _\__  ___ ____/ /_
  \ V - V - `/_/  ' /
  /_ / .__^_,/_/ /_\  version 1.5.2
  /_/

Using Python version 2.7.10 (default, May 28 2015 17:04:42)
SparkContext available as sc, HiveContext available as sqlContext.
>>>
```

We test Spark by counting the lines in PySpark:

```
>>> lines.first()
15/11/10 20:34:00 INFO SparkContext: Starting job: runJob at PythonRDD.scala:393
15/11/10 20:34:00 INFO DAGScheduler: Got job 1 (runJob at PythonRDD.scala:393) with 1 output partitions
15/11/10 20:34:00 INFO DAGScheduler: Final stage: ResultStage 1(runJob at PythonRDD.scala:393)
15/11/10 20:34:00 INFO DAGScheduler: Parents of final stage: List()
15/11/10 20:34:00 INFO DAGScheduler: Missing parents: List()
15/11/10 20:34:00 INFO DAGScheduler: Submitting ResultStage 1 (PythonRDD[3] at RDD at PythonRDD.scala:43), which has no missing parents
15/11/10 20:34:00 INFO MemoryStore: ensureFreeSpace(4840) called with curMem=165683, maxMem=556038881
15/11/10 20:34:00 INFO MemoryStore: Block broadcast_2 stored as values in memory (estimated size 4.7 KB, free 530.1 MB)
15/11/10 20:34:00 INFO MemoryStore: ensureFreeSpace(3036) called with curMem=170523, maxMem=556038881
15/11/10 20:34:00 INFO MemoryStore: Block broadcast_2_piece0 stored as bytes in memory (estimated size 3.0 KB, free 530.1 MB)
15/11/10 20:34:00 INFO BlockManagerInfo: Added broadcast_2_piece0 in memory on localhost:50501 (size: 3.0 KB, free: 530.3 MB)
15/11/10 20:34:00 INFO SparkContext: Created broadcast 2 from broadcast at DAGScheduler.scala:861
15/11/10 20:34:00 INFO DAGScheduler: Submitting 1 missing tasks from ResultStage 1 (PythonRDD[3] at RDD at PythonRDD.scala:43)
15/11/10 20:34:00 INFO TaskSchedulerImpl: Adding task set 1.0 with 1 tasks
15/11/10 20:34:00 INFO TaskSetManager: Starting task 0.0 in stage 1.0 (TID 2, localhost, PROCESS_LOCAL, 2147 bytes)
15/11/10 20:34:00 INFO Executor: Running task 0.0 in stage 1.0 (TID 2)
15/11/10 20:34:00 INFO HadoopRDD: Input split: file:/usr/local/Cellar/spark/README.md:0+1796
15/11/10 20:34:00 INFO PythonRunner: Times: total = 3, boot = -11313, init = 11316, finish = 0
15/11/10 20:34:00 INFO Executor: Finished task 0.0 in stage 1.0 (TID 2). 2143 bytes result sent to driver
15/11/10 20:34:00 INFO TaskSetManager: Finished task 0.0 in stage 1.0 (TID 2) in 16 ms on localhost (1/1)
15/11/10 20:34:00 INFO TaskSchedulerImpl: Removed TaskSet 1.0, whose tasks have all completed, from pool
15/11/10 20:34:00 INFO DAGScheduler: ResultStage 1 (runJob at PythonRDD.scala:393) finished in 0.016 s
15/11/10 20:34:00 INFO DAGScheduler: Job 1 finished: runJob at PythonRDD.scala:393, took 0.029001 s
u'# Apache Spark'
```

```

>>> 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':

	Name	Date Modified	Size	Kind
	0	Today, 8:51 PM	89 bytes	TextEdit...
	1	Today, 8:51 PM	178 KB	TextEdit...
	2	Today, 8:51 PM	76 bytes	TextEdit...
	3	Today, 8:51 PM	80 bytes	TextEdit...
	4	Today, 8:51 PM	80 bytes	TextEdit...
	5	Today, 8:51 PM	90 bytes	TextEdit...
	6	Today, 8:51 PM	106 bytes	TextEdit...
	7	Today, 8:51 PM	74 bytes	TextEdit...
	8	Today, 8:51 PM	98 bytes	TextEdit...
	9	Today, 8:51 PM	75 bytes	TextEdit...
	10	Today, 8:51 PM	54 bytes	TextEdit...
	11	Today, 8:51 PM	131 KB	TextEdit...
	12	Today, 8:51 PM	68 bytes	TextEdit...
	13	Today, 8:51 PM	72 bytes	TextEdit...
	14	Today, 8:51 PM	63 bytes	TextEdit...
	15	Today, 8:51 PM	72 bytes	TextEdit...
	16	Today, 8:51 PM	68 bytes	TextEdit...
	17	Today, 8:51 PM	35 KB	TextEdit...

Then I calculate TF-IDF and do clustering:

```
bin — bash — 112x35
15/11/15 14:10:24 INFO BlockManager: Found block rdd_1_1 locally
15/11/15 14:10:44 INFO PythonRunner: Times: total = 19241, boot = -6286, init = 6289, finish = 19238
15/11/15 14:10:45 INFO PythonRunner: Times: total = 19465, boot = -7859, init = 7862, finish = 19462
15/11/15 14:10:46 INFO PythonRunner: Times: total = 21366, boot = -21844, init = 21861, finish = 21349
15/11/15 14:10:46 INFO FileOutputCommitter: Saved output of task 'attempt_201511151410_0024_m_000000_48' to file :/Users/Jovial/Desktop/BigDataHW3/Output_TF_IDF/_temporary/0/task_201511151410_0024_m_000000
15/11/15 14:10:46 INFO SparkHadoopMapRedUtil: attempt_201511151410_0024_m_000000_48: Committed
15/11/15 14:10:46 INFO Executor: Finished task 0.0 in stage 24.0 (TID 48). 2108 bytes result sent to driver
15/11/15 14:10:46 INFO TaskSetManager: Finished task 0.0 in stage 24.0 (TID 48) in 21447 ms on localhost (1/2)
15/11/15 14:10:46 INFO PythonRunner: Times: total = 21542, boot = -4396, init = 4430, finish = 21508
15/11/15 14:10:46 INFO FileOutputCommitter: Saved output of task 'attempt_201511151410_0024_m_000001_49' to file :/Users/Jovial/Desktop/BigDataHW3/Output_TF_IDF/_temporary/0/task_201511151410_0024_m_000001
15/11/15 14:10:46 INFO SparkHadoopMapRedUtil: attempt_201511151410_0024_m_000001_49: Committed
15/11/15 14:10:46 INFO Executor: Finished task 1.0 in stage 24.0 (TID 49). 2108 bytes result sent to driver
15/11/15 14:10:46 INFO TaskSetManager: Finished task 1.0 in stage 24.0 (TID 49) in 21630 ms on localhost (2/2)
15/11/15 14:10:46 INFO TaskSchedulerImpl: Removed TaskSet 24.0, whose tasks have all completed, from pool
15/11/15 14:10:46 INFO DAGScheduler: ResultStage 24 (saveAsTextFile at NativeMethodAccessorImpl.java:-2) finished in 21.633 s
15/11/15 14:10:46 INFO DAGScheduler: Job 14 finished: saveAsTextFile at NativeMethodAccessorImpl.java:-2, took 2.1.660938 s
15/11/15 14:10:46 INFO SparkContext: Invoking stop() from shutdown hook
15/11/15 14:10:46 INFO SparkUI: Stopped Spark web UI at http://160.39.173.114:4040
15/11/15 14:10:46 INFO DAGScheduler: Stopping DAGScheduler
15/11/15 14:10:46 INFO MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stopped!
15/11/15 14:10:46 INFO MemoryStore: MemoryStore cleared
15/11/15 14:10:46 INFO BlockManager: BlockManager stopped
15/11/15 14:10:46 INFO BlockManagerMaster: BlockManagerMaster stopped
15/11/15 14:10:46 INFO OutputCommitCoordinator$OutputCommitCoordinatorEndpoint: OutputCommitCoordinator stopped!
15/11/15 14:10:46 INFO SparkContext: Successfully stopped SparkContext
15/11/15 14:10:46 INFO ShutdownHookManager: Shutdown hook called
15/11/15 14:10:46 INFO ShutdownHookManager: Deleting directory /private/var/folders/bx/v_wbc7zs5tdgllpy6yr37jfw000gn/T/spark-4b952ed4-3de4-41bc-b4ce-814cf45e4854/pyspark-271a599a-29e2-4a4f-8232-4b847828f8e6
15/11/15 14:10:46 INFO ShutdownHookManager: Deleting directory /private/var/folders/bx/v_wbc7zs5tdgllpy6yr37jfw000gn/T/spark-4b952ed4-3de4-41bc-b4ce-814cf45e4854
dyn-160-39-173-114:bin Jovial$
```

```
part-00000
[(1048576,[282907,542085,673930,808789,822462,903471],
[3.80252696614,6.66472784707,5.5661155584,6.66472784707,1.49922784241,7.64555710008])
(1048576,
[36,95,152,505,520,862,905,1877,2140,2265,2267,2380,2388,2424,2450,2518,2683,2702,303
3,3110,3247,3697,3932,4091,4126,4317,4415,4489,4718,4999,5237,5250,5430,5444,5467,554
2,5915,6028,6102,6327,6890,7027,7343,7640,7645,7801,8071,8103,8363,8364,8755,8815,892
0,8955,9233,9380,9398,9701,9793,9847,9960,10029,10127,10247,10455,10489,10598,10707,1
0769,10970,10987,10997,11089,11117,11191,11360,11511,11721,11811,12035,12180,12724,12
928,13003,13111,13167,13251,13252,13317,13373,13397,13462,13730,13807,13855,13954,139
61,13976,14018,14044,14193,14204,14276,14392,14434,14501,14726,14955,14981,15053,1568
4,15859,16128,16218,16427,17092,17640,17834,17902,17962,17970,17999,18245,18259,18390
,
18562,18572,18659,19003,19096,19142,19143,19220,19413,19422,19527,19650,19959,20056,2
0734,20952,20985,21035,21269,21379,21405,21733,21849,21909,21941,22477,22758,22769,22
774,23015,23192,23468,23675,24058,24165,24304,24315,24555,24717,25071,25083,26012,261
75,26261,26527,26701,26714,26828,26835,26909,26930,26942,26977,27014,27188,27300,2739
1,27714,27842,27875,27895,27914,28000,28067,28156,28199,28222,28510,28654,28747,28939
,
28982,29078,29195,29478,29561,29899,30044,30136,30143,30177,30230,30240,30617,30674,3
0807,30848,30930,30947,31247,31275,31327,31328,31397,31509,31529,32251,32263,32362,32
363,32517,32588,32626,32749,32751,32814,33197,33221,33745,33894,33904,33909,33950,339
80,34439,34491,34519,34521,34845,34994,35147,35388,35402,35411,35554,35603,35744,3618
1,36192,36268,36371,36660,36727,36737,36741,36748,36749,36751,36754,36756,36757,36787
,
36905,36955,36979,37110,37484,37496,37666,37702,38019,38031,38034,38105,38158,38284,3
8453,38592,38877,38912,39036,39086,39222,39403,39690,39854,40044,40097,40122,40125,40
520,40522,40691,40844,40871,41130,41732,41804,42150,42417,42491,42548,42661,42676,429
54,43182,43201,43230,43367,43394,43452,43685,43870,43971,44017,44253,44365,44719,4499
8,45034,45404,45631,45649,46372,46385,46387,46415,46989,47039,47166,47272,47352,48163
,]
```

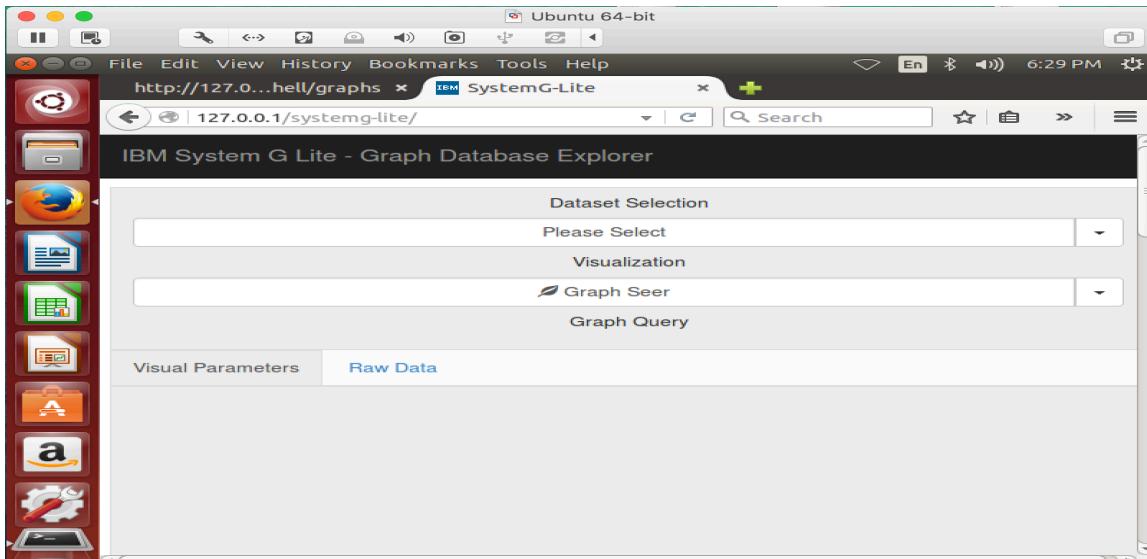
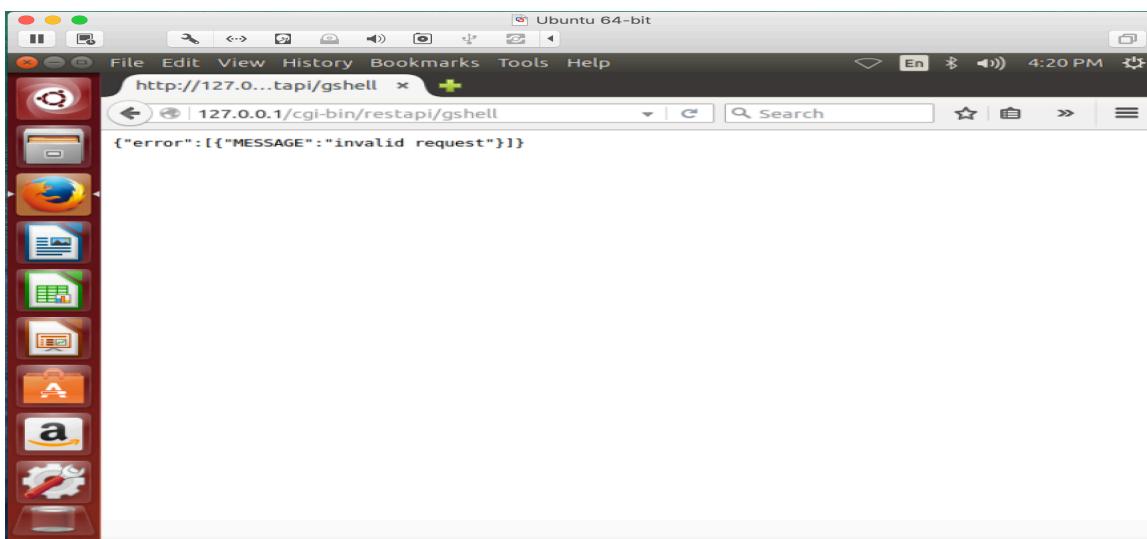
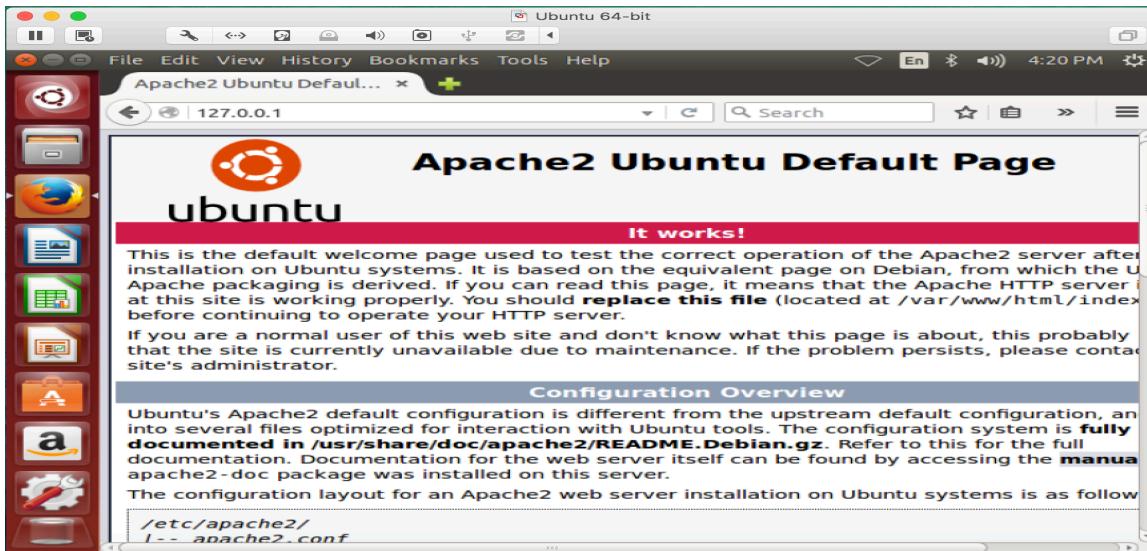
```
(SparseVector(1048576, {282907: 3.8025, 542085: 6.6647, 673930: 5.5661, 808789: 6.6647, 822462: 1.4992, 903471: 7.6456}), 2)
(SparseVector(1048576, {36: 7.6456, 95: 6.5469, 152: 6.1051, 505: 7.3579, 520: 4.0168, 862: 4.0257, 905: 10.8967, 1877: 6.7983, 2140: 5.2832, 2265: 2.5019, 2267: 2.933, 2380: 4.1592, 2388: 8.051, 2424: 7.5983, 2450: 7.3579, 2518: 13.5965, 2683: 3.0812, 2702: 7.6456, 3033: 7.6456, 3110: 5.8538, 3247: 4.601, 3697: 4.0528, 3932: 31.635, 4091: 4.8123, 4126: 7.6456, 4317: 4.0713, 4415: 12.2293, 4489: 4.062, 4718: 11.2817, 4999: 6.5469, 5237: 7.3579, 5250: 11.2174, 5430: 3.3923, 5444: 3.064, 5467: 6.3463, 5542: 6.6647, 5915: 28.7209, 6028: 23.6926, 6102: 8.935, 6327: 3.7813, 6890: 3.1439, 7027: 6.9524, 7343: 4.5853, 7640: 5.8422, 7645: 6.3463, 7801: 9.2711, 8071: 9.109, 8103: 7.3579, 8363: 6.0361, 8364: 2.8253, 8755: 5.5883, 8815: 7.3579, 8920: 4.5853, 8955: 5.8962, 9233: 8.051, 9380: 3.5184, 9398: 10.2055, 9701: 6.7285, 9793: 8.051, 9847: 2.5376, 9960: 6.1051, 10029: 3.5347, 10127: 13.9048, 10247: 15.2911, 10455: 8.0833, 10489: 5.6087, 10598: 7.1347, 10707: 4.983, 10769: 2.2672, 10970: 6.9524, 10987: 2.8692, 10997: 8.051, 11089: 4.3747, 11117: 5.4366, 11191: 5.6996, 11360: 6.9524, 11511: 7.1347, 11721: 7.1347, 11811: 1.6608, 12035: 4.0437, 12180: 12.8832, 12724: 18.7778, 12928: 2.278, 13003: 7.1347, 13111: 4.149, 13167: 6.6647, 13251: 5.2784, 13252: 4.6333, 13317: 3.1023, 13373: 6.2593, 13397: 7.3579, 13462: 11.3063, 13730: 8.4231, 13807: 4.1192, 13855: 7.3579, 13954: 4.0902, 13961: 6.0361, 13976: 7.1347, 14018: 3.5131, 14044: 0.7272, 14193: 7.6456, 14204: 3.6154, 14276: 4.6333, 14392: 4.6647, 14434: 12.4404, 14501: 5.5253, 14726: 5.0229, 14955: 4.4957, 14981: 5.1973, 15053: 18.0986, 15684: 4.7796, 15859: 4.3497, 16128: 4.7011, 16218: 4.873, 16427: 6.7983, 17092: 4.8123, 17640: 3.8921, 17834: 4.6498, 17902: 6.4416, 17962: 6.9524, 17970: 6.6647, 17999: 5.8538, 18245: 5.5253, 18259: 5.343, 18390: 6.9524, 18562: 18.6344, 18572: 6.5469, 18659: 6.1792, 19003: 4.601, 19096: 6.1792, 19142: 3.2147, 19143: 3.1058, 19220: 15.2014, 19413: 7.6456, 19422: 7.1347, 19527: 3.888, 19650: 7.6456, 19959: 6.1792, 20056: 4.4401, 20734: 8.051, 20952: 3.9, 20985: 6.9524, 21035: 6.9524, 21269: 7.1347, 21379: 4.0902, 21405: 0.909, 21733: 4.9375, 21849: 8.0693, 21909: 6.2593, 21941: 4.4815, 22477: 6.7983, 22758: 7.1347, 22769: 7.3579, 22774: 5.7997, 23015: 3.8169, 23192: 4.4401, 23468: 7.6456, 23675: 6.4416,
```

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:

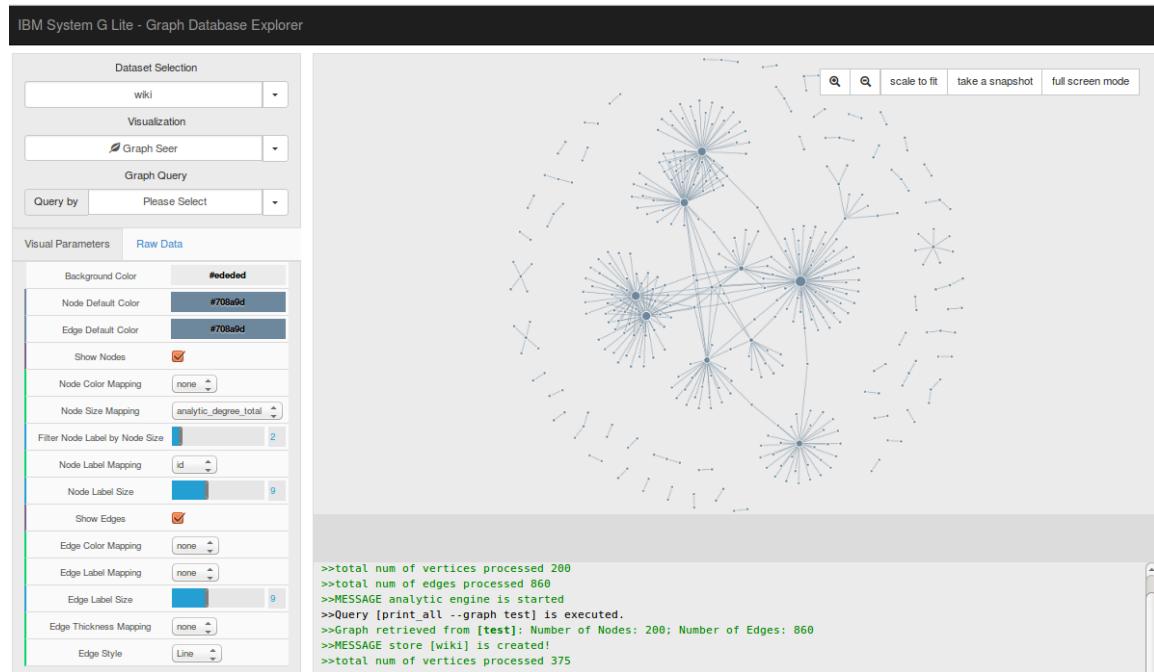


2.2: Download URL links in the Wikipedia data. Create a knowledge graph. Inject it into a graph database. Try some queries to find relevant terms. This can serve as keyword expansion. Show some visualizations of your queries.

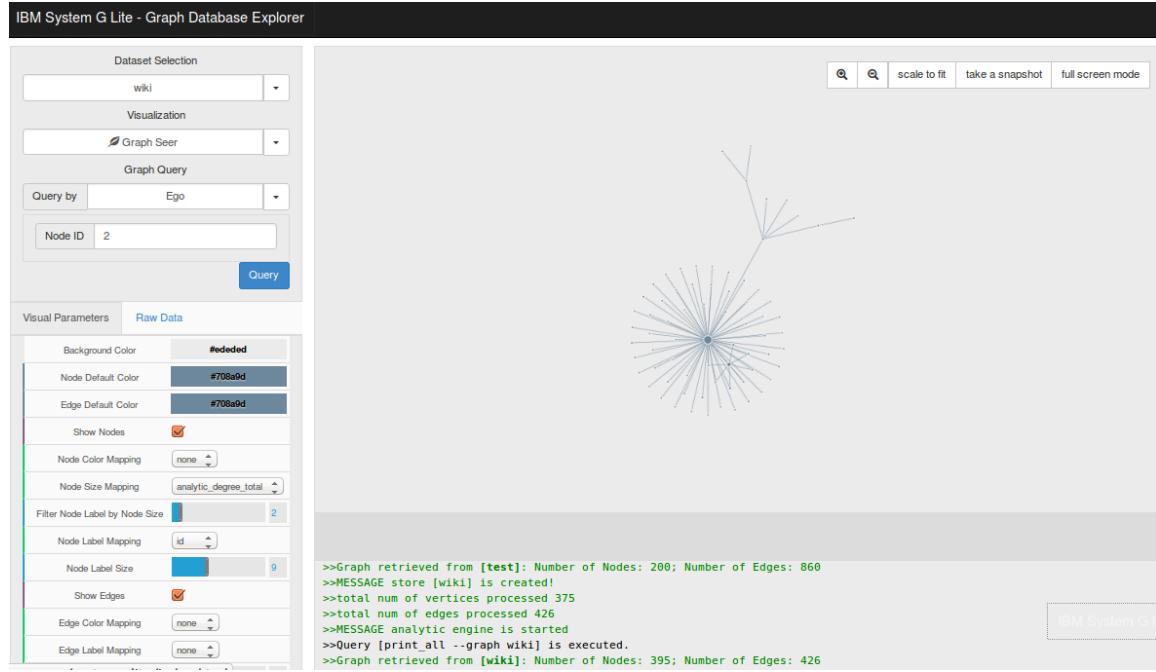
Here I used 100 Wikipedia URL links and converted them into node.csv and edge.csv as following:

A	B	A	B	C
id	title	source	target	value
1	!!	1	1664968	1
2	!!	2	3	1
3	!!!	2	747213	1
4	!!!!	2	1664968	1
5	!!!Fuck_You!!!	2	1691047	1
6	!!!Fuck_You!!!_And_Then_Some	2	4095634	1
7	!!!Fuck_You!!!_And_then_Some	2	5535664	1
8	!!!Fuck_You!!!_and_Then_Some	3	9	1
9	!!_(album)	3	77935	1
10	!!_(band)	3	79583	1
11	!!Destroy-Oh-Boy!!	3	84707	1
12	!!Fuck_you!!	3	564578	1
13	!!M	3	594898	1
14	!!Que_Corra_La_Voz!!	3	681805	1
15	!!_(chess)	3	681886	1
16	!!m	3	835470	1
17	I'O-Ikhung_language	3	880698	1
18	I=	3	1109091	1
19	I?	3	1125108	1
20	!?!?	3	1279972	1
22	I?_(chess)	3	1463445	1
23	IA_Luchar!	3	1497566	1
24	IAction_Pact!	3	1783284	1
25	IAdios_Amigos!	3	1997564	1
26	IAlabade!	3	2006526	1
27	IAlarma!	3	2070954	1
28	IAlarma!_(album)	3	2250217	1
29	IAlarma!_(disambiguation)	3	2268713	1
30	IAlarma!_(magazine)	3	2276203	1
31	IAlarma!_Records	3	2374802	1
32	IAlarma!_magazine	3	2571397	1
33	IAlfaro_Vive	3	2640902	1
34	IAll-Time_Qarreback!	3	2647217	1
35	IAll-Time_Qarreback!_(EP)	3	2732378	1
36	IAll-Time_Qarreback!_(album)	3	2821237	1
37	IAlla_tu!	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:

