Graph Analysis

Here, I would calculate the similarity of the documents I used in clustering_with_real_data.

And then, set a threshold value on the similarity to convert what I get above into a graph.

Finally, I would use Spark GraphX to calculate process of my graph: PageRank, Connected Components, Triangle Counting.

Similarity

```
from pyspark.ml.feature import Normalizer

dot_udf = udf(lambda x,y: float(x.dot(y)))
normalizer = Normalizer(inputCol="features", outputCol="norm")
data = normalizer.transform(featurizedData)
```

```
data = data.sample(False, 0.15, 0)
data.count()
```

```
In [18]: data = data.sample(False, 0.15, 0)
         data.count()

Out[18]: 1286
```

```
import numpy as np
import pandas as pd

feature = np.array(data.select('norm').collect())
feature = np.squeeze(feature)
similarity = np.dot(feature, feature.T)
similarity_df = pd.DataFrame(similarity[:,:])
similarity_df.head(20)
```

Out[19]:

	0	1	2	3	4	5	6	7	8	9	...	1276	1277	1278	1279	1280	1281	1282	1283	1284	1285
0	1.0	0.000000	0.000000	0.000000	0.0000	0.0	0.0	0.000000	0.000000	0.000000	...	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	0.0	1.000000	0.000000	0.000000	0.0000	0.0	0.0	0.585373	0.445609	0.000000	...	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.000000	1.000000	0.000000	0.0000	0.0	0.0	0.000000	0.000000	0.000000	...	0.0	0.0	0.762498	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.000000	0.000000	1.000000	0.7389	0.0	0.0	0.000000	0.000000	0.000000	...	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.000000	0.000000	0.738900	1.0000	0.0	0.0	0.000000	0.000000	0.000000	...	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.000000	0.000000	0.000000	0.0000	1.0	0.0	0.000000	0.000000	0.000000	...	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	0.0	0.000000	0.000000	0.000000	0.0000	0.0	1.0	0.000000	0.000000	0.000000	...	0.0	0.0	0.000000	0.0	1.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.585373	0.000000	0.000000	0.0000	0.0	0.0	1.000000	0.000000	0.000000	...	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.445609	0.000000	0.000000	0.0000	0.0	0.0	0.000000	1.000000	0.744859	...	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	0.0	0.000000	0.000000	0.000000	0.0000	0.0	0.0	0.000000	0.744859	1.000000	...	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	0.0	0.585373	0.000000	0.000000	0.0000	0.0	0.0	1.000000	0.000000	0.000000	...	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	0.0	0.469369	0.000000	0.479345	0.0000	0.0	0.0	0.000000	0.468922	0.000000	...	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.000000	0.000000	0.000000	0.0000	0.0	1.0	0.000000	0.000000	0.000000	...	0.0	0.0	0.000000	0.0	1.0	0.0	0.0	0.0	0.0	0.0
13	0.0	0.000000	0.000000	0.000000	0.0000	0.0	0.0	0.000000	0.000000	0.000000	...	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0
14	0.0	0.000000	0.000000	0.000000	0.0000	0.0	0.0	0.000000	0.000000	0.000000	...	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0
15	0.0	0.000000	0.000000	0.000000	0.0000	0.0	0.0	0.000000	0.000000	0.000000	...	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0
16	0.0	0.000000	0.767070	0.432295	0.0000	0.0	0.0	0.000000	0.000000	0.000000	...	0.0	0.0	0.584889	0.0	0.0	0.0	0.0	0.0	0.0	0.0
17	0.0	0.667858	0.000000	0.000000	0.0000	0.0	0.0	0.000000	0.667221	0.000000	...	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0
18	0.0	0.000000	0.600964	0.000000	0.0000	0.0	0.0	0.000000	0.000000	0.000000	...	0.0	0.0	0.788152	0.0	0.0	0.0	0.0	0.0	0.0	0.0
19	0.0	0.000000	0.000000	0.000000	0.0000	0.0	0.0	0.000000	0.000000	0.000000	...	0.0	0.0	0.343603	0.0	0.0	0.0	0.0	0.0	0.0	0.0

20 rows × 1286 columns

Spark GraphX

```
from pyspark.sql.functions import *
from pyspark.mllib.linalg.distributed import IndexedRow, IndexedRowMatrix
dot_udf = udf(lambda x,y: float(x.dot(y)))
final = data.alias("i").join(data.alias("j"), col("i.ID") < col("j.ID")).select(
    col("i.ID").alias("i"), col("j.ID").alias("j"), dot_udf("i.norm",
    "j.norm").alias("dot")).sort("i", "j")
final.show(20)
```

In [21]:
```
dot_udf = udf(lambda x,y: float(x.dot(y)))
final = data.alias("i").join(data.alias("j"), col("i.ID") < col("j.ID")).select(
    col("i.ID").alias("i"), col("j.ID").alias("j"), dot_udf("i.norm", "j.norm").alias("dot")).sort("i", "j")
final.show(20)
```

```
+---+-----+-------------------+
|  i|    j|                dot|
+---+-----+-------------------+
|  1|10034|0.18812289948451108|
|  1|10035|0.16434868260177252|
|  1|10041| 0.5237723764224875|
|  1|10045|0.24906630219818326|
|  1|10047|0.20197095305715634|
|  1|10048|0.20172805964855012|
|  1| 1006|                0.0|
|  1|10061|0.21712901010863264|
|  1| 1007|  0.546505476465932|
|  1|10072| 0.5903035910728687|
|  1|10076|0.25387297384196034|
|  1|10081| 0.5596986921429918|
|  1|10089|0.24942921722448055|
|  1|10097| 0.2119689818407752|
|  1|10099|0.22585512467472038|
|  1|10102|0.25425271829534546|
|  1|10114| 0.3831223125067753|
|  1|10117|0.25387297384196034|
|  1|10120|0.19878364469652213|
|  1|10125|0.49695131518962377|
+---+-----+-------------------+
only showing top 20 rows
```

```
concat_func = udf(lambda x: 3 if x >= 0.7 else 1 )
final_filter = final.filter(final.dot>0.5)
final_filter.show()
```

In [22]:
```
concat_func = udf(lambda x: 3 if x >= 0.7 else 1 )
final_filter = final.filter(final.dot>0.5)
final_filter.show()
```

```
+---+-----+------------------+
|  i|    j|               dot|
+---+-----+------------------+
|  1|10041|0.5237723764224875|
|  1| 1007| 0.546505476465932|
|  1|10072|0.5903035910728687|
|  1|10081|0.5596986921429918|
|  1|10128|0.5110273480790805|
|  1| 1014|0.5669338874012189|
|  1|  103|0.5003188294736406|
|  1|10320|0.5003188294736406|
|  1|10396|0.5015953499219707|
|  1|10508|0.6634057359450216|
|  1|10511|0.6634057359450216|
|  1|10519| 0.520535562406713|
|  1|10557|0.8101809070465138|
|  1|10586|0.5158979454846762|
|  1|11500|0.7037763785262148|
|  1| 1163|0.5125719021879845|
|  1|12025|0.5234086191349514|
|  1|12029|0.5828820779297796|
|  1|12030|0.5965573593221276|
|  1|12103| 0.614667660007497|
+---+-----+------------------+
only showing top 20 rows
```

```
concat_func = udf(lambda x: int(3) if x >= float(0.7) else int(1))
final_filter_float=final_filter.select(final_filter.i,final_filter.j,final_filter.dot.c
ast("float"))
concat_df = final_filter_float.withColumn("relationship",
concat_func(final_filter_float.dot))
concat_df = final_filter_float.withColumn("relationship",
concat_func(final_filter_float.dot))
concat_df.show()
```

In [23]:
```
concat_func = udf(lambda x: int(3) if x >= float(0.7) else int(1))
final_filter_float=final_filter.select(final_filter.i,final_filter.j,final_filter.dot.cast("float"))
concat_df = final_filter_float.withColumn("relationship", concat_func(final_filter_float.dot))
concat_df = final_filter_float.withColumn("relationship", concat_func(final_filter_float.dot))
concat_df.show()
```

```
+---+-----+----------+------------+
|  i|    j|       dot|relationship|
+---+-----+----------+------------+
|  1|10041|0.52377236|           1|
|  1| 1007|0.54650545|           1|
|  1|10072| 0.5903036|           1|
|  1|10081| 0.5596987|           1|
|  1|10128|0.51102734|           1|
|  1| 1014| 0.5669339|           1|
|  1|  103| 0.5003188|           1|
|  1|10320| 0.5003188|           1|
|  1|10396| 0.5015954|           1|
|  1|10508| 0.6634057|           1|
|  1|10511| 0.6634057|           1|
|  1|10519| 0.5205356|           1|
|  1|10557| 0.8101809|           3|
|  1|10586| 0.5158979|           1|
|  1|11500|0.70377636|           3|
|  1| 1163|0.51257193|           1|
|  1|12025| 0.5234086|           1|
|  1|12029| 0.5828821|           1|
|  1|12030| 0.5965574|           1|
|  1|12103|0.61466765|           1|
+---+-----+----------+------------+
only showing top 20 rows
```

```
featurizedData_rename= featurizedData.withColumnRenamed('_unit_id','id')
concat_df_rename=concat_df.withColumnRenamed("i","src").withColumnRenamed("j","dst")
e = concat_df_rename
v = featurizedData_rename.select("id","text","features")
```

```
e.toPandas().to_csv('e.csv',sep=',',index = False, encoding = 'utf-8')
v.toPandas().to_csv('v.csv',sep=',',index = False, encoding = 'utf-8')
```

```
home/jb4076/spark/bin/pyspark --packages graphframes:graphframes:0.6.0-spark2.3-s_2.11
```

```
>>> from graphframes import *
>>> e = sqlContext.read.format("csv").option("header", "true").load("./hw3/e.csv")
>>> v = sqlContext.read.format("csv").option("header", "true").load("./hw3/v.csv")
>>> g = GraphFrame(v, e)
```

```
>>> g.vertices.show()
```

```
>>> g = GraphFrame(v, e)
>>> g.vertices.show()
+---+--------------------+--------------------+
| id|                text|            features|
+---+--------------------+--------------------+
|  1|              _n Keuë|(20,[8,15],[1.908...|
|  6|                 Acèh|(20,[2,19],[1.062...|
|  8|            Nabi Adam|(20,[9,15],[1.678...|
|  9|            Adat Acèh|(20,[2,6,19],[1.0...|
| 10|         Afghanistan|(20,[11],[1.91401...|
| 11|              Afrika|(20,[10],[1.71406...|
| 12|        Afrika Barat|(20,[10,13],[1.71...|
| 13|Afrika Blah Seulatan|(20,[9,10,14],[1....|
| 14|      Afrika Teungoh|(20,[2,10],[1.062...|
| 15|         Afrika Timu|(20,[10,17],[1.71...|
| 16|        Afrika Barôh|(20,[2,7,10],[1.0...|
| 17|               Agama|(20,[6],[1.650915...|
| 18|Ahlussunah Wal-ja...|(20,[0,2,4,17],[1...|
| 19|             Akhirat|(20,[15],[1.54420...|
| 21|               Allah|(20,[14],[1.68515...|
| 22|            Almazán|(20,[5,15],[1.764...|
| 23|             Amirika|(20,[12],[1.37694...|
| 24|    Amirika Seulatan|(20,[12,14],[1.37...|
| 25|     Amirika Teungöh|(20,[1,2,12],[1.3...|
| 26|       Amirika Utara|(20,[2,12],[1.062...|
+---+--------------------+--------------------+
only showing top 20 rows
```

```
>>> g.edges.show()
```

```
>>> g.edges.show()
+---+-----+---------+------------+
|src|  dst|      dot|relationship|
+---+-----+---------+------------+
|  1|10041|0.52377236|           1|
|  1| 1007|0.54650545|           1|
|  1|10072| 0.5903036|           1|
|  1|10081| 0.5596987|           1|
|  1|10128|0.51102734|           1|
|  1| 1014| 0.5669339|           1|
|  1|  103| 0.5003188|           1|
|  1|10320| 0.5003188|           1|
|  1|10396| 0.5015954|           1|
|  1|10508| 0.6634057|           1|
|  1|10511| 0.6634057|           1|
|  1|10519| 0.5205356|           1|
|  1|10557| 0.8101809|           3|
|  1|10586| 0.5158979|           1|
|  1|11500|0.70377636|           3|
|  1| 1163|0.51257193|           1|
|  1|12025| 0.5234086|           1|
|  1|12029| 0.5828821|           1|
|  1|12030| 0.5965574|           1|
|  1|12103|0.61466765|           1|
+---+-----+---------+------------+
only showing top 20 rows
```

```
>>> vertexInDegrees = g.inDegrees
>>> vertexInDegrees.show()
```

```
>>> vertexInDegrees = g.inDegrees
>>> vertexInDegrees.show()
+-----+--------+
|   id|inDegree|
+-----+--------+
| 1436|      49|
| 1512|      55|
|14887|      37|
|12394|      44|
| 1572|     114|
| 5925|     171|
|11332|      62|
| 6248|      80|
|  125|       9|
|11236|      33|
|10114|       9|
|14525|      31|
|12811|      57|
| 2696|      86|
| 2700|      88|
|14455|      50|
| 6081|      72|
| 9622|      67|
|15176|      57|
|10296|       8|
+-----+--------+
only showing top 20 rows
```

PageRank

```
>>> results = g.pageRank(resetProbability=0.15, maxIter=10)
>>> results.vertices.select("id", "pagerank").show()
```

```
>>> numFollows = g.edges.filter("relationship = 1").count()
>>> results = g.pageRank(resetProbability=0.15, maxIter=10)
2018-10-21 04:03:16 WARN  CacheManager:66 - Asked to cache already cached data.
>>> results.vertices.select("id", "pagerank").show()
+--------------------+------------------+
|                  id|          pagerank|
+--------------------+------------------+
|                2162|0.6800907666239872|
|                3414|0.6800907666239872|
|                6033|0.6800907666239872|
|                 987|0.6800907666239872|
|               10752|0.6800907666239872|
|               10609|0.6800907666239872|
|               14159|0.6800907666239872|
|               15043|0.6800907666239872|
|               12533|0.6800907666239872|
|                4833|0.6800907666239872|
|                2676|  2.31121391859917|
|               10732|0.6800907666239872|
|Post 15 jaroj da ...|0.6800907666239872|
|               15177|0.6800907666239872|
|                9756|17.746375818761152|
|                 481|0.6800907666239872|
|               12704|0.9365779771168136|
|                1772|0.6800907666239872|
|                1772|0.6800907666239872|
|        Antaŭnelonge|0.6800907666239872|
+--------------------+------------------+
only showing top 20 rows
```

```
>>> results.edges.select("src", "dst", "weight").show()
```

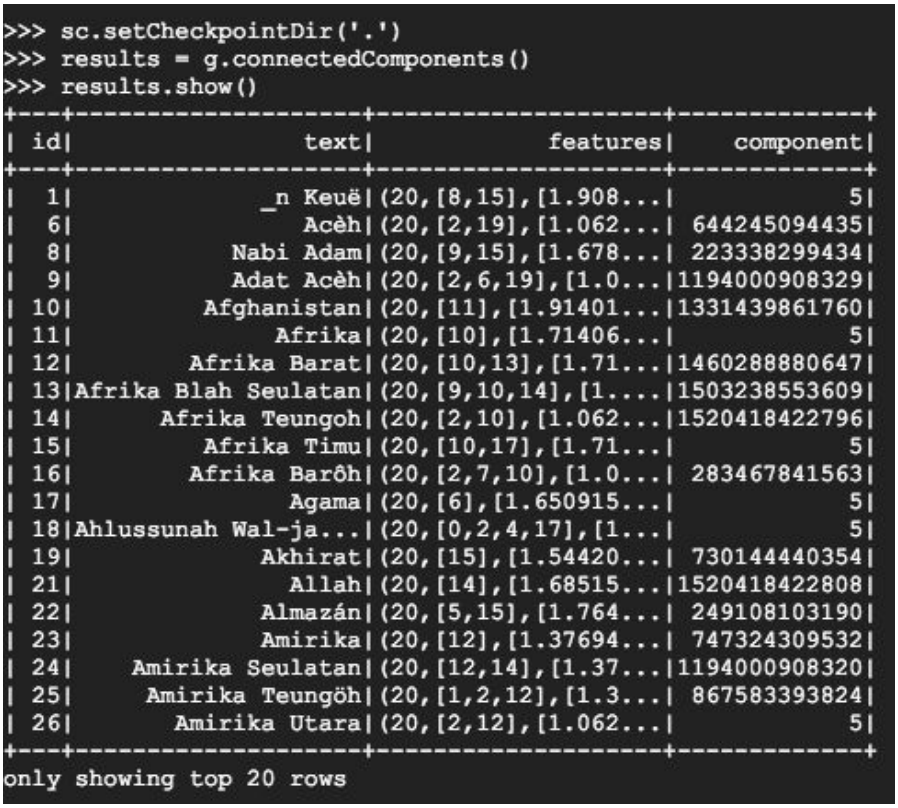
Triangle Counting:

```
>>> results = g.triangleCount()
>>> results.select("id", "count").show()
```

```
>>> results = g.triangleCount()
>>> results.select("id", "count").show()
+-----+-----+
|   id|count|
+-----+-----+
|10096|    0|
|10351|    0|
|10436|    0|
| 1090|    0|
|11078|    0|
|11332| 3878|
|11563|    0|
|12394| 2706|
|12529|    0|
|12847|    0|
|13192|    0|
|13282|    0|
|13442|    0|
|13610|    0|
|13772|    0|
|13865|    0|
|14157|    0|
|14204|    0|
| 1436| 7047|
|14369|    0|
+-----+-----+
only showing top 20 rows
```

Connected Components:

```
>>> sc.setCheckpointDir('.')
>>> results = g.connectedComponents()
>>> results.show()
```



```
>>> sc.setCheckpointDir('.')
>>> results = g.connectedComponents()
>>> results.show()
+---+--------------------+--------------------+------------+
| id|                text|            features|   component|
+---+--------------------+--------------------+------------+
|  1|              _n Keuë|(20,[8,15],[1.908...|           5|
|  6|                Acèh|(20,[2,19],[1.062...| 644245094435|
|  8|           Nabi Adam|(20,[9,15],[1.678...| 223338299434|
|  9|           Adat Acèh|(20,[2,6,19],[1.0...|1194000908329|
| 10|         Afghanistan|(20,[11],[1.91401...|1331439861760|
| 11|              Afrika|(20,[10],[1.71406...|           5|
| 12|        Afrika Barat|(20,[10,13],[1.71...|1460288880647|
| 13|Afrika Blah Seulatan|(20,[9,10,14],[1....|1503238553609|
| 14|      Afrika Teungoh|(20,[2,10],[1.062...|1520418422796|
| 15|         Afrika Timu|(20,[10,17],[1.71...|           5|
| 16|        Afrika Barôh|(20,[2,7,10],[1.0...| 283467841563|
| 17|              Agama|(20,[6],[1.650915...|           5|
| 18|Ahlussunah Wal-ja...|(20,[0,2,4,17],[1...|           5|
| 19|             Akhirat|(20,[15],[1.54420...| 730144440354|
| 21|               Allah|(20,[14],[1.68515...|1520418422808|
| 22|             Almazán|(20,[5,15],[1.764...| 249108103190|
| 23|             Amirika|(20,[12],[1.37694...| 747324309532|
| 24|    Amirika Seulatan|(20,[12,14],[1.37...|1194000908320|
| 25|     Amirika Teungöh|(20,[1,2,12],[1.3...| 867583393824|
| 26|       Amirika Utara|(20,[2,12],[1.062...|           5|
+---+--------------------+--------------------+------------+
only showing top 20 rows
```

If we print more data, we could see that there may be some relationships between inDegree and PageRank, because the id with larger PageRank usually has larger inDegree, which indicates that when a website's content has higher correlation with other websites' content, it has larger effects in the Wiki website network.