COL 764
Assignment 3

Prior Ranking of Documents

PART 2 :-

- **Approach taken for the PageRank algorithm & Method of choice:-**

★ **First choice :-** The first choice of the module to find the pagerank was NetworkX, in which the functions used to,
1. make graph : G = nx.Graph()
2. add node : G.add_nodes_from(filenames)
3. add edges : G.add_weighted_edges_from(Edge_list), where Edge_list = List of (doc1name, docname2, Similarity)
4. Find pagerank : nx.pagerank(G), the damping factor by default was 0.85

Drawback :- The module was taking a lot of time and memory to complete its work. So, I decided to try some other module for it.

★ **Final choice** :- The final choice of the module was Scikit-Network, because it is compartively very faster than the previous module used.
The functions used in this module to,
1. make graph : graph = edgelist2adjacency(edges, undirected=True), where edges is list of 3-tuple, say (a,b,c), where c is the similarity between file1 and file2 where file1 path = filenames[a] and file2 path = filenames[b]
2. add node : It updates the node info every time an edge is added, so we don't have to add the node manually in it.
3. add edges : Edges are added during the formation of the graph above in the form of adjacency list from the edgelist.
4. Find pagerank : p.fit_transform(graph), where p is an object of the PageRank() class of this library

-> It works very faster than the above module and thus fulfills our purpose of page ranking quite well.

- **Top-20 highest pagerank values (Jaccard similarity) :-**

sci.electronics/54247 : 0.00017829176845167544
rec.autos/103727 : 0.00017183642721740127
sci.med/59454 : 0.00016970115862445861
alt.atheism/54160 : 0.00016946863996549074
sci.electronics/54164 : 0.00016752589366186114
sci.med/59407 : 0.00016751982840710488
soc.religion.christian/21438 : 0.00016744802128547466
talk.politics.guns/55040 : 0.00016702622939096542
sci.electronics/54208 : 0.00016694893904671458

talk.politics.misc/179050 : 0.00016667053581579264
sci.med/59271 : 0.0001664531419539805
comp.graphics/38863 : 0.00016633609875519337
comp.sys.ibm.pc.hardware/60741 : 0.00016588395261424595
soc.religion.christian/21611 : 0.0001658302460056389
sci.electronics/54486 : 0.00016573449545556658
comp.sys.ibm.pc.hardware/60804 : 0.0001656926757028287
comp.os.ms-windows.misc/10201 : 0.00016568055507731376
comp.windows.x/68087 : 0.00016558143357398722
soc.religion.christian/21538 : 0.00016523096421543304
soc.religion.christian/21586 : 0.00016509283188127476

- **Top-20 highest pagerank values (Cosine similarity) :-**

sci.crypt/16123 : 0.00038207883924680264
sci.crypt/15812 : 0.00034710974621388676
talk.politics.misc/178786 : 0.000341866309741099
talk.politics.misc/179058 : 0.00034051880264642344
talk.religion.misc/84380 : 0.0003340341852863259
talk.politics.mideast/77195 : 0.0003277035938294755
talk.politics.guns/55067 : 0.00032641519897553194
talk.politics.mideast/77198 : 0.00032362856487607675
comp.sys.mac.hardware/52004 : 0.0003225650005522987
talk.politics.mideast/77186 : 0.00032214347818804563
talk.politics.mideast/77397 : 0.0003201828223613545
soc.religion.christian/21662 : 0.00031957072094940746
alt.atheism/54233 : 0.0003180697718795947
sci.crypt/15929 : 0.0003178089677775077
soc.religion.christian/21726 : 0.00031768689323286655
alt.atheism/53538 : 0.00031712532181934393
soc.religion.christian/21496 : 0.0003158591877420795
talk.politics.misc/178724 : 0.0003129048811159344
soc.religion.christian/21744 : 0.00031257783259370665
talk.politics.misc/178894 : 0.0003103938222035475

- **Conclusions :-**

JACCARD :-

1. The documents which were ranked higher in Jaccard were sort of random documents because the similarity metric which jaccard uses is weak.
2. The page rank of all the documents in jaccard is almost similar because the metric here treats the documents not much differently.

COSINE :-

1. The difference in the ranking scores of the documents in Cosine is visible as the cosine similarity metric is stronger than jaccard and thus it refrains from giving pseudo similairties.

2. Much of the top-20 documents in this are from religion and politics. And these documents were large as well , so they may have contained a lot of terms which have also contributed to their similarity with the other documents.

THE final conclusion is that COSINE works way better than JACCARD in finding the PAGERANKS.

PART 1:-

1. find_termset :- Function to find the termset of the document
2. SimM1 :- Function to find the similariy between the pairs of the document using Jaccard method
3. tfIdf_preProcess :- Function to find the termfrequency map for all the documents and the inverse document frequency map
4. Magnitude. :- Finding the magnotude of the tf-idf map of a document
5. DotProduct. :- Finding the dot product of the tf-idf maps of two documents
6. SimM2 :- Function to find the similariy between the pairs of the document using Cosine method
7. Sim :- Driving function which decides the similarity function to call and the write the similarity output in a file

- **Submitted By :-**
  -> Nikita Bhamu
  -> 2018CS50413