Assignment 3

-Shruti Kumari
-2018CS50420

Similarity Computations

All the filenames in the given collection directory were traversed and were stored in an array.

Now, vocabulary is generated using these filenames and is written to a file. Vocabulary can either be generated during runtime or can be loaded from the disk using functions `getVocabData` and `loadStored` respectively.

Now, the terms in all of these files were generated using the `getTerms` function, which splits the text using a set of delimiters and then stems the words return after splitting. All these terms were stored in a dictionary `tfDocs` using the function `getTfFiles` . If the input is given to compute cosine similarities, then tf-idf vectors for all files were also computed beforehand using the function `getTfIdfDocs`. Initially the tfidf vector was computed for each doc for each word in the vocabulary. This led to a lot of time being used up to get the similarity scores for the whole collection. The next approach was to compute the tfidf vector for each doc for only the terms present in that document. While computing the similarity of such 2 docs, a union of the terms present in both the docs was generated and final cosine similarity was computed between tfidf vectors of these documents for the terms that were present in the union. This led to a huge decrease in time taken to compute the whole collection.

The functions to compute term frequencies and idf were:

$$tf_{i,j} = \log_2(1+f_{i,j})$$

$$idf_j = \log_2(1+|D|/f_j)$$

Now to compute similarity scores, spatial.distance.cosine was used to compute cosine distances between two tfidf vectors in case of cosine similarity and in case of jaccard similarity, python set methods such as intersection and union were used.

Computing PageRank

Edge list was created using the similarity score file generated in the first part. Each line ‘doc1 doc2 score’ was modelled as an edge between nodes doc1 and doc2 having a weighted edge between them with the weight score. Also, each document name was mapped to a unique integer so that it can be given as a node to the sknetwork library. In this way, edge_list was constructed. Taking help of sknetwork library, we transform this undirected weighted graph to a directed weighted graph.

I used the sknetwork library to compute pageRank scores. I created a graph using the `edgelist2adjacency()` with the parameter undirected as True. Then I used this graph to compute scores using `pagerank.fit_transform(graph)`. These scores were then sorted and top 20 results were retrieved. The method `edgelist2adjacency` constructs a symmetric adjacency matrix which is basically a directed graph. Now this directed weighted graph can then easily be used to compute pagerank scores.

The list of documents and their PageRank scores with top-20 highest PageRank values for each of the similarity functions are-

Jaccard:

[('sci.electronics/54247', 0.0001776243358991465),
('sci.med/59271', 0.0001738743332941905),
('sci.med/59454', 0.00017070609459822846),
('talk.religion.misc/84349', 0.00017001942181942725),
('alt.atheism/54160', 0.00016916910265607124),
('talk.politics.guns/54554', 0.00016865770890702223),
('sci.med/59407', 0.000168566590579249),
('rec.sport.baseball/104999', 0.00016814541996052645),
('sci.electronics/54263', 0.00016784082169108339),
('comp.sys.ibm.pc.hardware/60804', 0.00016733498340582817),
('comp.sys.ibm.pc.hardware/60807', 0.00016730456319635474),
('comp.graphics/39040', 0.00016698888800119544),
('sci.electronics/54208', 0.0001664321060007481),
('comp.sys.mac.hardware/52250', 0.00016662181867292712),
('soc.religion.christian/21719', 0.00016618896926011423),
('rec.autos/103727', 0.00016604934605055587),

('comp.os.ms-windows.misc/10781', 0.00016603199388684698),
('sci.electronics/54164', 0.00016545728830959537),
('rec.motorcycles/104755', 0.00016540661723089795),
('rec.sport.hockey/54264', 0.00016534429682019824)]

Cosine:

[('talk.politics.misc/179058', 0.00041064793547760476),
('sci.crypt/16123', 0.0003948361474914495),
('talk.politics.misc/178908', 0.00037422929515837263),
('sci.crypt/15812', 0.00035391676860596464),
('talk.politics.misc/178786', 0.00034284069385450653),
('alt.atheism/53637', 0.0003330343961937091),
('talk.politics.mideast/77195', 0.0003305938291924295),
('talk.politics.misc/179029', 0.00032960766673475236),
('talk.religion.misc/84380', 0.0003289447010105843),
('talk.politics.mideast/77198', 0.00032846250230695115),
('talk.politics.mideast/77397', 0.0003282221335104556),
('talk.religion.misc/84079', 0.0003267491548402098),
('talk.politics.misc/178724', 0.00032419482358373813),
('soc.religion.christian/21662', 0.00032382655361517555),
('comp.sys.mac.hardware/52004', 0.00032282825245196056),
('talk.politics.mideast/77186', 0.00032093797173886085),
('sci.crypt/15929', 0.0003199705664955383),
('talk.politics.misc/178776', 0.0003191761714288871),
('talk.politics.guns/55067', 0.00031833910272709517),
('talk.politics.guns/55087', 0.00031720830488568823)]

It can be observed that most of the documents ranked by cosine similarity are from either politics or religion.

Politics: These documents contain press releases by the White House. These are generally very much cited and referenced to because they are legitimate documents and they hold a lot of value since they are views of a US president.

Religion: These documents are email exchanges which talk about homosexuality. These are linked together as most of them are replies to each other and hence are cited and referred to more in comparison to other documents.

Apart from these two groups, a random observation of the file 52004 in the folder comp.sys.mac.hardware tells us that it is an FAQ file. Since it clarifies doubts on multiple topics, it is related to and cited by many other documents and hence is present in the top 20 files.

Trying to get such insights from the file ranking by Jaccard similarity is difficult as we cant find such specific groups there. This shows that Cosine similarity is a better measure than jaccard similarity as it provides meaningful results.