Assignment 3
COL764: Info. Retrieval and Web Search

Manoj Kumar
2018CS50411

Implementation Details:

Term Overlap:

For the term overlap or jaccard method, I extracted all the terms of each document and created the respective sets for all documents. Later for each unordered pair of documents, I calculate jaccard similarity using the formula:

$$Sim_{jaccard}(d_1, d_2) = \frac{|TermSet_{d_1} \cap TermSet_{d_2}|}{|TermSet_{d_1} \cup TermSet_{d_2}|}$$

TF-IDF similarity:

For the cosine similarity, we created the following dictionaries:
1. allwords : keys of type string (words) and values integer values that denote the total no. Of documents in which that term is appeared.
2. tfDocuments : keys of type string (document name) and values are the dictionaries with key word and values are their frequencies of that word in that document.
3. total : keys of type string (document name) and values are values are total no. Of words in that document.
4. idf : keys of type string (words) and values are of type float representing the idf of that word.
5. tfIdfDocs : keys of type string (document name) and values of type dictionary with keys = word and values = tf*idf value for that word.

I implented **getCosine** function which finds the cosine similarity of two documents with the above calculated values using these formulae:

$$\forall_{d_i \in D,\, j \in V} \qquad tf_{i,j} = 1 + \log_2(f_{i,j})$$

$$\forall_{d_i \in D,\, j \in V} \qquad idf_j = \log_2\left(1 + \frac{|D|}{df_j}\right)$$

$$Sim_{cosine}(\vec{d}, \vec{d'}) = \frac{\vec{d} \cdot \vec{d'}}{||\vec{d}|| \times ||\vec{d'}||}$$

***minor correction in tf_i,j formula: it is tf_i,j = log_2(1+f_i,j)**

Computing PageRank:

Approaches Taken:

Each document is modeled as nodes of the pagerank graph and the similarity between two nodes are denoted by weighted edges.

Since, we have an undirected weighted graph, the following approach is used to convert it to a directed graph which the pagerank calculating library can accept as input:

1. the pagerank library takes an undirected graph and changes it to directed graph by adding two directed edges for each undirected edge.
2. Weight of the edges is the similarity score obtained from jaccard or cosine algorithms.
3. Now the graph is a weighted and directed graph.

I experimentied with two approaches i) networkx library ii) sknetwork library to calculate the pagerank. Sknetwork library is faster. Hence, I implemented this in the submission.

PageRank computes ranking of the nodes (documents) in the graph G based on the structure of the incoming links.

It is based on using a transition matrix and current state probability vector.

Let probability vector is X = (x1, x2, ..., xn)

In the transition probability matrix P, the ith row tells us where we go next from state i.

The detailed description of this algorithm can be found here: https://www.geeksforgeeks.org/page-rank-algorithm-implementation/

Top 20 documents:

Jaccard:

```
sci.med/59407 0.00017774273943633312
sci.electronics/54247 0.00017722234353789102
soc.religion.christian/21611 0.00017402609924739676
comp.windows.x/68087 0.00017335027691436583
comp.sys.ibm.pc.hardware/60991 0.00017244639650837555
comp.sys.ibm.pc.hardware/60807 0.00017225367927762406
talk.religion.misc/84349 0.00017198237622431247
comp.graphics/38863 0.00017153534955589349
sci.med/59532 0.00017140981671106317
sci.crypt/16139 0.00017138055820663207
comp.sys.ibm.pc.hardware/60964 0.00016951455713586135
rec.autos/103425 0.00016907980328328997
rec.sport.hockey/54264 0.00016816393220842087
comp.sys.mac.hardware/52288 0.00016804809819330577
soc.religion.christian/21586 0.00016788050109865175
alt.atheism/53319 0.00016757274949487112
comp.sys.mac.hardware/52047 0.00016751906225321626
rec.sport.baseball/104999 0.00016738211034825394
soc.religion.christian/21431 0.00016728687155402556
rec.autos/103809 0.00016722969633342295
```

Cosine:

```
talk.politics.misc/178908 0.0003228642167865114
talk.politics.misc/179058 0.0003222271350253987
soc.religion.christian/21496 0.00030715908029268823
talk.politics.misc/179073 0.0003006383356220721
talk.politics.mideast/77198 0.0002900286116183443
talk.politics.mideast/77195 0.00028855086986646374
talk.religion.misc/84223 0.00028628616194368846
comp.sys.mac.hardware/52004 0.0002853538918601026
soc.religion.christian/21597 0.00028489422623630313
talk.politics.misc/178786 0.0002824949509971628
sci.crypt/15812 0.0002819242344801426
soc.religion.christian/21748 0.0002802851471378475
soc.religion.christian/21458 0.0002790805122689751
talk.religion.misc/84079 0.00027796576699090875
comp.graphics/39638 0.0002773884192142614
comp.graphics/39078 0.0002773070857639062
talk.politics.misc/179034 0.00027692391507436995
alt.atheism/53538 0.00027668372108657257
alt.atheism/53639 0.00027664855073263403
alt.atheism/54233 0.0002751261070939691
```

Analysis of results:

Cosine Similarity:

The top 20 documents obtained from cosine similarity (based on TF-IDF) are divided in 2 clusters. One is related to politics and the other is regarding religion.
Top 5 politics documents are from talk.politics.misc and it can be seen that all of them are press releases from the White House.
The second clustor (religion) has the documents from soc.reliogion.christianity and these are related to the email exchanges on homosexuality. Most of the documents are replies to each other and hence are linked to each other in pagerank.
It concludes the following results:

1. Trusted sources of information (eg. White house) are ranked highly.
2. Email exchanges on a perticular topic which are cited by many others in the group are also ranked highly.

Jaccard Similarity:

Not much observable insights could have been found in jaccard similarity. No clusters of documents are found. Even documents from same sub-collection didn’t have the same topics.

Hence, documents retrieved from the cosine similarity have more accurate pagerank as compared to the jaccard similarity.