Name : Prathusha JS Naidu

Information Retrieval
Phase 5

1. Introduction

In this assignment, I have performed analysis of 504 HTML documents using document clustering. The code from the previous stages has been used for performing preprocessing steps such as tokenization and also calculation of term weights. Following command has to be executed to run the program:
$python3 clustering.py input output

where,
clustering.py is the name of the program
input is the input directory with all html files.
output folder is for writing the result of clustering. The result is written into file cluster.txt

2. Data Structures Used

Nested dictionaries have been used extensively to store information like Term Document Matrix and Similarity matrix. Single level dictionaries have been used to store information about clusters.

With the formation of each new cluster, a new row is added to the similarity matrix and the similarity score with all other entries is updated.

3. Similarity Matrix

A similarity matrix is constructed using the term document matrix and is defined as a matrix in which entry *i,j* is the similarity of documents i and j, computed using the cosine score or some other metric. A document is perfectly similar to itself, so the entries on the main diagonal are all 1. Similarity is also symmetric, i.e. *sim(i,j) = sim(j,i)*, so the similarity matrix is upper triangular in form.

Formula for calculating cosine similarity:

$$similarity(doc_i, doc_j) = \frac{dotproduct(doc_i, doc_j)}{\|\mathbf{doc_i}\| * \|\mathbf{doc_j}\|}$$

4. Group Average Link Method

This is the method used for executing agglomerative clustering. After forming a new cluster, the information is updated in the cluster_info dictionary and a new row is added in the Similarity matrix. The Group Average Link method is used to calculate similarities between documents in the clusters and all other clusters/documents.

$$distance(cl_p, cl_r) = \frac{\sum_{doc_i \in cl_p} \sum_{doc_j \in cl_r} similarity(doc_i, doc_j)}{n_p + n_r}$$

5. Document Clustering Steps:

while(num of active clusters > 1)

Step1: Create simiarity matrix and a new flag for every document/cluster indicating its status.

Step2: Find documents/clusters with highest similarity and greater than 0.4.

Step3: Merge and update cluster_list accordingly. Also set the flags of merged documents to -1.

Step4: Update similarity matrix using Group Average Link method.

Step5: Repeat from Step2
end of while

6. Evaluation

a) Which pair of HTML documents is the most similar?

Documents 467.html and 426.html are the most similar and are the first documents to be merged into a cluster. Computed using calculate_highest_sim.

b) Which pair of documents is the most dissimilar?

Documents 6.html and 3.html are the most dissimilar document with a score of 0.0 . Computed using calculate_lowest_sim.

c)Which document is the closest to the corpus centroid?

The centroid is calculated after the completion of document clustering. The document with least distance to this centroid is 089.html

Following is a snapshot of the output file cluster.txt. Number after the arrow is the new cluster number

```
Clustering documents :
467  426 ------>504
451  74 ------>505
494  159 ------>506
372  146 ------>507
104  33 ------>508
296  268 ------>509
16  11 ------>510
206  27 ------>511
208  169 ------>512
269  97 ------>513
447  257 ------>514
307  244 ------>515
227  128 ------>516
410  325 ------>517
342  174 ------>518
356  348 ------>519
316  148 ------>520
450  387 ------>521
443  238 ------>522
```