Information Retrieval Phase 5

I have analysed the 504-document HTML corpus by using document clustering. I have used my code from earlier phases of the project for tokenising, calculating normalised term weights and building a term document matrix.

CLUSTERING

As the first step in clustering, the similarity matrix was constructed. Additionally, every document was marked as active by setting a flag. Once the similarity matrix was ready, we retrieved the pair of documents with highest similarity. After checking if the similarity score is greater than 0.4, the two documents were merged together into a new cluster. The new cluster was assigned a number which is one greater than the number of files (504 in this case). A row corresponding to this new cluster was added to Similarity matrix after computing similarity scores between itself and all other documents using group average link method. After this step, the two documents which were merged are marked as deactivated The flag is set for new cluster, marking it as active and also the number of documents in the cluster is stored in the cluster_info as mentioned before. Similarly the process continues, until there exists two clusters (or documents) which have similarity greater than 0.4.

IMPLEMENTATION

I used the term document matrix which was built in earlier phases for constructing the similarity matrix. Term document matrix is a nested dictionary, with the first level indexed by documents and the inner level by terms. Similarity matrix is also a nested dictionary, indexed by pair of documents. Additionally, since a similarity matrix is upper triangular, only one entry was made for a pair of documents. A new row is added to the similarity after formation of cluster, which contains the similarity score of that cluster with all other existing clusters.

I used two other dictionaries to keep track of active clusters and number of documents in each cluster

The **Similarity matrix** was constructed using the cosine similarity score between pair documents. The formula for computing cosine similarity is as follows,

$$\text{cosinesimilarity}(d_i, d_j) = \frac{\text{dotproduct}(d_i, d_j)}{k d_i k \;\; k d_j k}$$

As noted earlier, 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. Term-Document matrix is referenced for obtaining the term weights corresponding to terms in each document.

After forming a new cluster, the Similarity matrix was updated with the scores between the new cluster and existing ones. Group Average Link method was used for computing the similarity score between two clusters or one cluster and another document. The information about documents in a cluster was stored in a dictionary, with the cluster as key and list of documents as value.

OUTPUT

The output is written to the file cluster.txt. For every merge between clusters or documents, a line was written into the file as, Merging cluster1 and cluster2 into new_cluster.

RESULTS

1. Most Similar documents
Documents 403.html and 405.html are most similar as they were first to be merged into a cluster. This was computed using get_highest_sim method in the program which returns the pair of documents in the similarity matrix with highest score.

2. Most dissimilar documents
Documents 186.html and 155.html are most dissimilar with a similarity score of 0.0 (the first such occurrence was chosen).

3. Which document is the closest to the corpus centroid?
After document clustering (without the similarity score threshold of 0.4, the centroid of documents belonging to the cluster was found out. The document with least distance to this centroid is 405.html