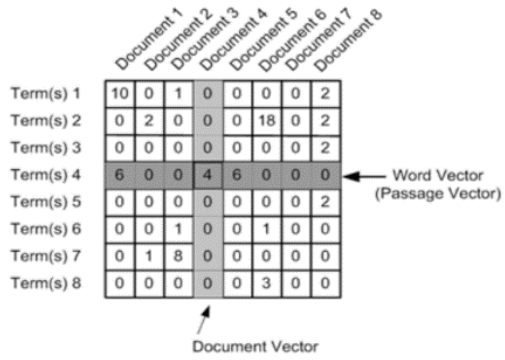
Information Retrieval
Project Phase 5

Introduction:

In this assignment we go over 504 html files and introduce clustering. A similarity matrix of the similarity of every document with each other is formed. Using this matrix Hierarchical agglomerative clustering it done , the clustering is done based on the value of cosine similarity. The idea is to cluster the most similar documents into closer clusters and most dissimilar documents into clusters far apart from each other. As a hierarchical clustering is used and a dendrogram is used, if no threshold is applied all the documents would eventually fall into a single large cluster.

Methodology:

The 504 html files are read one at a time, the regular preprocessing is done as was done in previous assignments, tokenization, removal of stopwords and special characters. Once this is done the term document matrix is formed as in assignment 3, every term has an entry for a document if it has a non-zero tf -idf weight in the below format.



Once the TDM is obtained , similarity matrix is to be formed as similarity is the basis for clustering.

$$\mathbf{A} \cdot \mathbf{B} = \|\mathbf{A}\| \, \|\mathbf{B}\| \cos\theta \qquad \text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}},$$

Cosine similarity is used for this, the formula to find cosine similarity is as below. The cosine of 2 vectors can be found by the Euclidian dot product.

The resulting similarity ranges from −1 meaning exactly opposite, to 1 meaning exactly the same, with 0 indicating orthogonality or decorrelation, while in-between values indicate intermediate similarity or dissimilarity. In information retrieval this value will range from 0 to 1 as tf-idf values cannot be negative. Angular distance between two vectors is calculated by angular similarity = 1 - angular distance , which is

what we use in this project . Once similarity for every cell is calculated, the similarly matric is formed . A document is perfectly similar to itself, therefore the elements on the diagonal are all 1 , similarity id symmetric sim(a,b) = sim (b,a) . Below is a section of the cosine similarity matrix.

Cosine_similarity [[1. 0.02092255 0. ... 0.00805277 0.00528622 0.00596853]

[0.02092255 1. 0.00244534 ... 0.02802044 0.01500667 0.00557273]

[0. 0.00244534 1. ... 0.01373655 0.01522495 0.00538624]

...

[0.00805277 0.02802044 0.01373655 ... 1. 0.86171198 0.88899571]

[0.00528622 0.01500667 0.01522495 ... 0.86171198 1. 0.91685896]

[0.00596853 0.00557273 0.00538624 ... 0.88899571 0.91685896 1.]]

Subtracting it from 1 provides cosine distance which I will use for plotting on a Euclidean (2-dimensional) plane.

The values obtained after 1-cosine similarity is as below.

[[5.55111512e-16 9.79077448e-01 1.00000000e+00 ... 9.91947226e-01

9.94713783e-01 9.94031471e-01]

[9.79077448e-01 -6.66133815e-16 9.97554665e-01 ... 9.71979560e-01

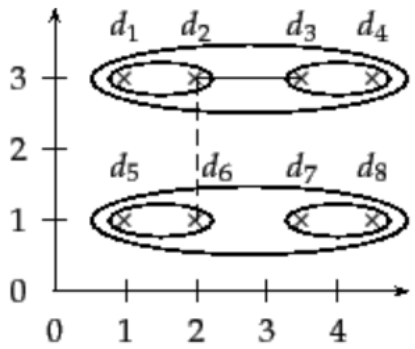
9.84993332e-01 9.94427273e-01]

[1.00000000e+00 9.97554665e-01 3.33066907e-16 ... 9.86263454e-01

9.84775055e-01 9.94613763e-01]

...

Clustering: **Hierarchical agglomerative clustering** is a method of clustering which uses bottom up approach, it starts by making singleton clusters with single documents, each cluster contains one data point C1 = {x1} .



$$D(X, Y) = \min_{x \in X, y \in Y} d(x, y),$$

Find pair of clusters with closest min D (ci ,cj), merge ci and cj into new cluster c(i+j) , remove ci and cj from C and add C(i+j) . Repeat this process till only one cluster is left or till where the threshold is set.

This procedure has a run time of $O(n^2 d + n^3)$ create, traverse through the similarity matrix. A single linkage maximum similarity clustering is used where the clustering is done using cosine similarity of two most similar documents (closest pair of elements) as shown below. n single-linkage clustering, the distance between two clusters is determined by a single element pair, namely those two elements (one in each cluster) that are closest to each other. The shortest of these links that remains at any step causes the fusion of the two clusters whose elements are involved. The method is also known as nearest neighbor clustering. The result of the clustering can be visualized as a dendrogram.

The linkage function is described by where *X* and *Y* are any two sets of elements considered as clusters, and *d*(*x*,*y*) denotes the distance between the two elements *x* and *y*. To obtain better results similarity using ward's method is used in this method agglomerative hierarchical clustering is done by choosing a pair of clusters based on the optimal value of similarity. This method minimizes the total variance within a cluster. This is to make sure most similar documents are tightly clustered and the dissimilar documents are away from each other.

The initial cluster distances in Ward's minimum variance method are therefore defined to be the squared Euclidean distance between points:

$$d_{ij} = d(\{X_i\}, \{X_j\}) = \|X_i - X_j\|^2.$$

The matrix obtained from cosine similarity as shown above is then passed into the above mentioned clustering algorithm. In this clustering method, meaning that at each stage, the pair of clusters with minimum between-cluster dist are merged. I used the precomputed cosine distance matrix (*dist*) to calclate a linkage_matrix, which I then plot on to a dendrogram.

The Linkage Matrix is as below

[[1.01000000e+02 1.29000000e+02 0.00000000e+00 2.00000000e+00]

[4.16000000e+02 4.19000000e+02 0.00000000e+00 2.00000000e+00]

[4.34000000e+02 4.35000000e+02 1.31112131e-02 2.00000000e+00]

...

[1.00000000e+03 1.00100000e+03 1.77989812e+01 3.57000000e+02]

[9.96000000e+02 1.00200000e+03 2.63751915e+01 4.61000000e+02]

[9.84000000e+02 1.00300000e+03 3.98891649e+01 5.03000000e+02]]

To perform this the library used is scipy.cluster.hierarchy.linkage. scipy.cluster.hierarchy.linkage(y, method='single', metric=similarity, optimal_ordering=False)

The input to this is from the library and function sklearn.metrics.pairwise.**cosine_similarity**(*X*, *Y=None*, *dense_output=True*) where X and Y are the two documents to be compared , this returns a matrix of size X xY.

The clustering algorithm is performed using the library *class* sklearn.cluster.Ward(*n_clusters=2*, *memory=Memory(cachedir=None)*, *connectivity=None*, *copy= None*, *n_components=None*, *compute_full_tree='auto'*, *pooling_func=<function mean at 0x2b9c7c5e7320>*)

Here we do not mention the number of clusters as we do not want to limit the clustering in these terms . We instead apply a clustering threshold which would clustering will cease when no two clusters (or documents) have similarity greater than 0.4. We have been asked in the assignment to have a threshold of 0.4 for clustering, which means if the distance between the 2 clusters which could be merged is lesser than 0.4 then the clustering should be stopped right there. We explore both the options of stopping at 0.4 and also continuing after 0.4, just as to get the documents which are most dissimilar.

This is done using the function and library scipy.cluster.hierarchy.fcluster**(***Z***,** *t***,** *criterion='inconsistent'***,** *depth=2***,** *R=None***,** *monocrit=None*). Assigning t=0.4 does the job of ceases the clustering when the similarity between two clusters is not more than 0.4. It returns a cluster of length n.

Once the clustering is done, to view the results better a dendrogram is plot. This is done with the help of dendrogram(linkage_matrix, orientation="right", labels=titles); The linkage matrix would be the matrix returned from the clustering algorithm.

The dendrogram is plotted as below by importing dendrogram from scipy.cluster.hierarchy

```
ax = dendrogram(linkage_matrix, above_threshold_color='#ffffff'
,orientation="right",labels=list(docAndWords.keys()),color_threshold=0.4)

plt.tick_params(\
    axis= 'x',                  # changes apply to the x-axis
    which='both',               # both major and minor ticks are affected
    bottom='off',               # ticks along the bottom edge are off
    top='off',                  # ticks along the top edge are off
    labelbottom='off')
```

Below is the sample of dendrogram without applying the threshold.

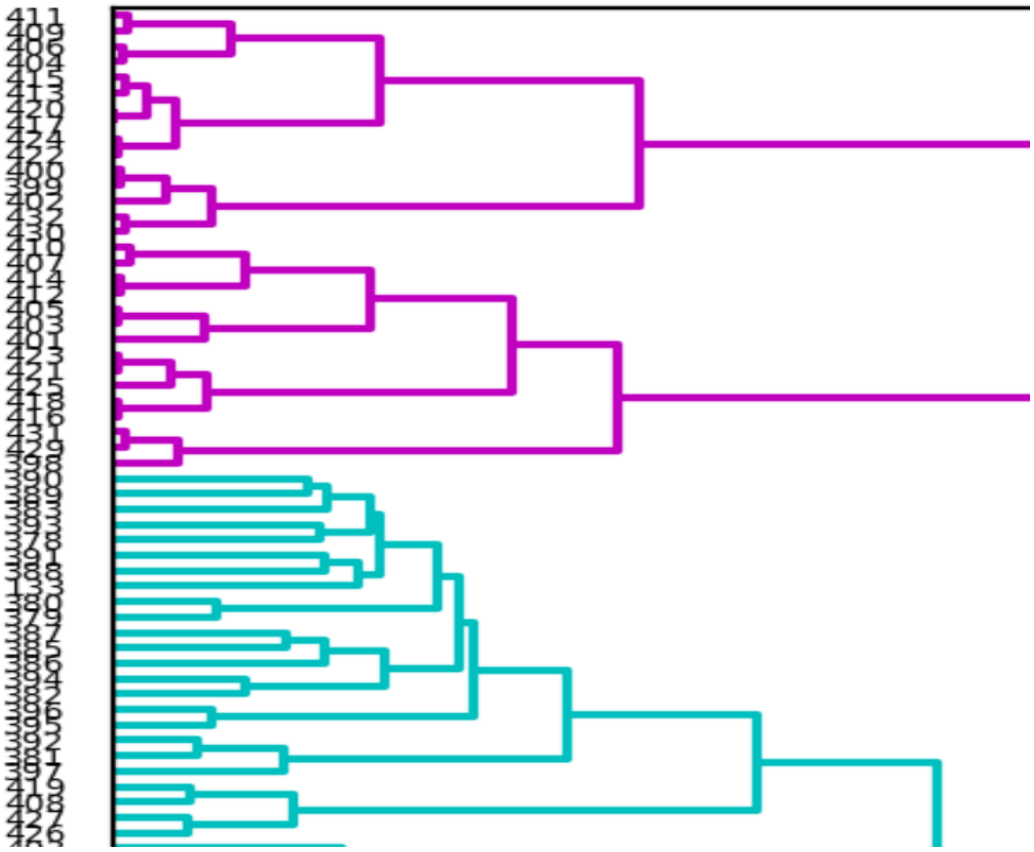


Figure: Dendrogram without Threshold

The complete file is attached along with the report in the zip file. In the above dendrogram all the documents are clustered eventually into a single cluster . The threshold is colored with blue.

When the 0.4 threshold is applied we will not be able to cluster all the documents and hence obtain dendrogram as below.

Figure: Dendrogram with threshold

From the dendrogram and also from the similarity matrix shown above

we see that there are a few documents very similar to each other like the document

- 420, 417
- 405, 412, 403
- 416, 418
- 436, 435, 434 etc

The most dissimilar documents would not be obtained as we stop clustering for anything below 0.4 similarity, however I have removed the cutoff to find the most dissimilar documents , which would be the documents that have the longest vertical line between them , in our case it would be the 2 documents that would form the cluster in the very end , the last tree formed in the dendrogramwithoutthreshold.

- 373 , 411
- 354, 409
- 404, 371 etc

References:

1. https://scikit-learn.org/stable/
2. http://www.site.uottawa.ca/~diana/csi5180/TextClustering.pdf
3. https://www.youtube.com/watch?v=XJ3194AmH40&list=PLBv09BD7ez_7qIbBhyQDr-LAKWUeycZtx&index=2
4. http://brandonrose.org/clustering
5. https://en.wikipedia.org/wiki/Hierarchical_clustering
6. https://en.wikipedia.org/wiki/Ward%27s_method