# PROJECT PHASE 5

## Name: Chelsea Morais

## CMSC476 Information Retrieval

### Introduction

This project phase code implements agglomerative clustering using the single-link method to the collection of HTML documents based on their textual content. It employs text preprocessing techniques including lowercasing, removing punctuation and numbers, and removing stop words. Following that, the documents' Term Frequency-Inverse Document Frequency (TF-IDF) representation is calculated, and a pairwise similarity matrix is produced using cosine similarity. The most similar clusters are iteratively combined until a certain threshold is attained. In addition, it lists the total number of clusters, the most different and similar document pairs, and the document that is the closest to the collection's centroid.

Input : python3 Clusters.py input

Output : Print statements

**PS: The merged clusters in the output screenshots are numbered from 0 hence 0 corresponds to Document 001.html and 1 corresponds to Document 002.html and so on.**

**Preprocessing + Tf-idf scores + Similarity matrix**

1. The first preprocessing step is converting the html document to text using html2text(). This removes the html tags. Next the text is lowercased and line breaks, punctuations , numbers, underscores as well as all the stopwords are removed.
2. After the preprocessing step is completed the Tf Idf values are obtained. The collection of html documents is transformed into a matrix of TF-IDF (Term Frequency-Inverse Document Frequency) properties using this vectorizer.
The TF-IDF statistic measures the prominence of a phrase inside a document or group of documents. It is frequently employed in text mining and information retrieval activities.
The document's variable is then passed to the TfidfVectorizer object's fit_transform() method.
3.This process converts the provided documents into a matrix representation to be converted into a similarity matrix. The similarity matrix is calculated using the calculated tf-idf scores.

**Algorithm : Agglomerative Clustering with single link method**

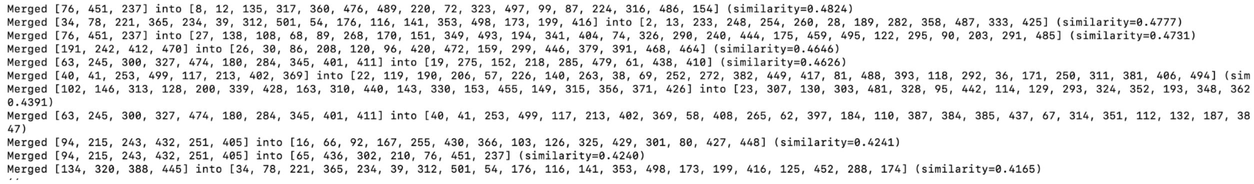
In Agglomerative clustering documents are clustered based on their similarity. In each iteration, the algorithm finds the pair of clusters (max_i and max_j) with the highest similarity (calculated as the minimum similarity between any two points, one from each cluster). The algorithm mergers the clusters and updates the similarity matrix and then sets the similarity scores of the merged clusters to 1.0. The process continues until the highest similarity falls below the specified threshold.

Step1:  Each html document is initially placed in its own cluster.

Step2: We iterate over each cluster and compare each pair of clusters to find the pair with the highest similarity. Then Calculate the similarity between two clusters as the minimum similarity between any two points, one from each cluster.

Step3: The clusters are merged if the threshold of 0.4 is not crossed and their similarity scores of the merged clusters are updated.

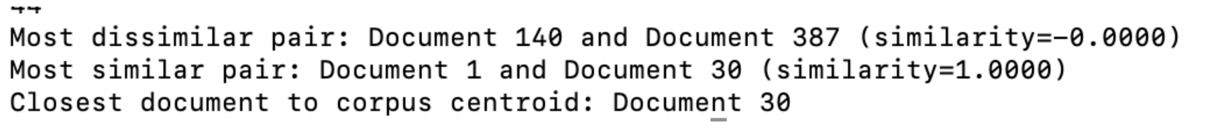
Step4: Step 2 and 3 is repeated until the threshold of 0.4 is not crossed.



```
Merged [76, 451, 237] into [8, 12, 135, 317, 360, 476, 489, 220, 72, 323, 497, 99, 87, 224, 316, 486, 154] (similarity=0.4824)
Merged [34, 78, 221, 365, 234, 39, 312, 501, 54, 176, 116, 141, 353, 498, 173, 199, 416] into [2, 13, 233, 248, 254, 260, 28, 189, 282, 358, 487, 333, 425] (similarity=0.4777)
Merged [76, 451, 237] into [27, 138, 108, 68, 89, 268, 170, 151, 349, 493, 194, 341, 404, 74, 326, 290, 240, 444, 175, 459, 495, 122, 295, 90, 203, 291, 485] (similarity=0.4731)
Merged [191, 242, 412, 470] into [26, 30, 86, 208, 120, 96, 420, 472, 159, 299, 446, 379, 391, 468, 464] (similarity=0.4646)
Merged [63, 245, 300, 327, 474, 180, 284, 345, 401, 411] into [19, 275, 152, 218, 285, 479, 61, 438, 410] (similarity=0.4626)
Merged [40, 41, 253, 499, 117, 213, 402, 369] into [22, 119, 190, 206, 57, 226, 140, 263, 38, 69, 252, 272, 382, 449, 417, 81, 488, 393, 118, 292, 36, 171, 250, 311, 381, 406, 494] (sim
Merged [102, 146, 313, 128, 200, 339, 428, 163, 310, 440, 143, 330, 153, 455, 149, 315, 356, 371, 426] into [23, 307, 130, 303, 481, 328, 95, 442, 114, 129, 293, 324, 352, 193, 348, 362
0.4391)
Merged [63, 245, 300, 327, 474, 180, 284, 345, 401, 411] into [40, 41, 253, 499, 117, 213, 402, 369, 58, 408, 265, 62, 397, 184, 110, 387, 384, 385, 437, 67, 314, 351, 112, 132, 187, 38
47)
Merged [94, 215, 243, 432, 251, 405] into [16, 66, 92, 167, 255, 430, 366, 103, 126, 325, 429, 301, 80, 427, 448] (similarity=0.4241)
Merged [94, 215, 243, 432, 251, 405] into [65, 436, 302, 210, 76, 451, 237] (similarity=0.4240)
Merged [134, 320, 388, 445] into [34, 78, 221, 365, 234, 39, 312, 501, 54, 176, 116, 141, 353, 498, 173, 199, 416, 125, 452, 288, 174] (similarity=0.4165)
```

The image above shows the ending few merges where the last cluster merge takes place at 0.4165 similarity

**Results**



```
Most dissimilar pair: Document 140 and Document 387 (similarity=-0.0000)
Most similar pair: Document 1 and Document 30 (similarity=1.0000)
Closest document to corpus centroid: Document 30
```

Which pair of HTML documents is the most similar?

Documents 1 and Documents 30

Which pair of documents is the most dissimilar?

Document 140 and Document 387

Which document is the closest to the corpus centroid?

Document 30

Number of obtained clusters :  44