Kurt Drexel

5/6/2024

Information Retrieval

At the start of this homework assignment I decided to use the baseline of Homework 2, although the most recent homework would have worked with the assignment I decided to use hw2 off of the recommendation of a classmate as a good baseline. I did not alter or remove any previous parts from the homework assignment before I simply added a part at the very bottom of the code to allow me to compete the necessary requirements. The program at this point will take in the entire 503 documents tokenizing them and removing stopwords. It will then calculate the tf\*idf weights for each term and output them into the output files. The Output file folder will contain 503 text documents each pertaining to their respected html document. The program will then take the output files made and calculate a matrix of the weights. Using the matrix of the weights a similarity matrix will be developed using cosine. Using the similarity matrix I develop a linkage matrix using the linkage function given by Scipy. Finally using the Linkage matrix I develop the clusters where the documents will be clustered together. When creating this program I originally attempted to use sklearn to try and successfully get all clusters to appear. While there was promise with the creation of the program in order to be able to better visualize the clusters created I decided to use SciPy. With the use of Scipy I was able to have a dendrogram created to assist in my visualization of the clusters and ensure the clusters had gone to completion. With the inclusion of Scipy I was able to use their clustering techniques as well as their functions linkage, fcluster, and cosine. My first step to create when creating this algorithm was to create a function that would assist the program in reading my output files in the form of “term | Weight”. The first thing my function did was create a dictionary to house the weights for each document. Then iterating through the documents, I begin to iterate through each line in the specified document splitting at the “|” to get the term and weight separated, and storing each in the dictionary under the specified document. My next line of code I then iterate over each document, iterating over each token in each document, retrieving the tf\*idf values for each token in each document. I then use a nested for loop in order to create a 2d matrix for each token with 0 being present if the token is not present in a document, if it is the weight for the token in that document is uploaded to the 2D matrix, calling the 2d matrix created my tfidf matrix. I then create my similarity matrix using the cosine imported with Scipy. To create my similarity matrix I iterate over the rows (documents) in my tfidf matrix, on each iteration I compare the current document with every other document within the my output files. On each comparison I do 1-cosince(doc1,doc2) calculating the cosine similarity between the two documents. With 1 meaning the documents are identical and -1 meaning the documents are not similar at all. Subtracting this value from 1 gives the measure of the cosine distance. Once doc1 has been compared to all other documents in corpus. I then create a linkage matrix using the linkage function also provided by Scipy to compute the linkage matrix for hierarchical clustering based on my similarity matrix calculated. The linkage function uses the parameter of my similarity matrix and the method of ‘average’. The average method specifies the linkage criteria for computing the linkage matrix computing the average of the pairwise distances between all points in two clusters. I then calculated my clusters using the fcluster function given by Scipy. This function fcluster took 3 parameters my previously calculated linkage matrix, the threshold of 0.4 and the criteria of ‘distance’. The distance criteria tells fcluster that the threshold is interpreted as the distance value in the linage matrix. My next snippet of code is used to create a print my dendrogram. Assigning titles to both the x and y axis. The size of the graph. And using the dendrogram function with the parameters of the linkage matrix. Then printing out my dendrogram. My couple lines of code are used to create a dictionary to house all my created clusters. I then iterate through all my clusters and for each cluster I check if it is not already within my dictionary, if it is not I add it to the dictionary. If it is already within the cluster dictionary the document id is appended to the list of documents already associated with the cluster id indicating a cluster has been made of another cluster. I then print my resulting list of clusters to screen with a simple for loop. This for loop iterates through my cluster dictionary printing out each document within the cluster along with its cluster id. A screenshot of a computer

Description automatically generatedWhen assigning names to my clusters I decided to simply assign them number names that will start with the first cluster being 1 and increase as more clusters are created with the largest number name being the last cluster. Initially each document will be given its own cluster id. For instance if there are 4 documents document 1 will have a cluster id of 1, document 2 a cluster id of 2, document 3 a cluster id of 3, and document 4 will have a cluster id of 4. If document 1 and 2 and merged together it will have a cluster id of 5, and if cluster 3 and 4 are merged together it will have a cluster id of 6. Finally if the two final clusters are merged together it will have a cluster id of 7. Finding the most similar pair of documents was relatively easy as it was the first cluster created by my algorithm being document 102.txt and document 130.txt. The most dissimilar pair of documents was slightly more difficult to find, with my final results for the most dissimilar pair being document 433.txt and 435.txt. Then finally to find the document most closely to the center of the corpus, I found the dendrogram provided by Scipy to be very helpful for this as with this dendrogram is can be easily seen that document 001.txt is the closest to the corpus centroid.

Outputs

Most Similar Pair of Documents:

('102.txt', '130.txt')

Most Dissimilar Pair of Documents:

('433.txt', '435.txt')

Document Closest to the Corpus Centroid:

001.txt

First 100 steps after the creation of the original clusters for the original 503 clusters.

Merging Steps:

Step 1: Merge Document 102.0 and Document 130.0

Step 2: Merge Document 441.0 and Document 444.0

Step 3: Merge Document 403.0 and Document 405.0

Step 4: Merge Document 422.0 and Document 424.0

Step 5: Merge Document 421.0 and Document 423.0

Step 6: Merge Document 404.0 and Document 406.0

Step 7: Merge Document 416.0 and Document 418.0

Step 8: Merge Document 407.0 and Document 410.0

Step 9: Merge Document 439.0 and Document 442.0

Step 10: Merge Document 417.0 and Document 420.0

Step 11: Merge Document 399.0 and Document 400.0

Step 12: Merge Document 409.0 and Document 411.0

Step 13: Merge Document 65.0 and Document 81.0

Step 14: Merge Document 340.0 and Document 359.0

Step 15: Merge Document 408.0 and Document 427.0

Step 16: Merge Document 149.0 and Document 180.0

Step 17: Merge Document 398.0 and Document 431.0

Step 18: Merge Document 434.0 and Document 435.0

Step 19: Merge Document 419.0 and Document 426.0

Step 20: Merge Document 412.0 and Document 414.0

Step 21: Merge Document 436.0 and Document 437.0

Step 22: Merge Document 492.0 and Document 498.0

Step 23: Merge Document 111.0 and Document 148.0

Step 24: Merge Document 381.0 and Document 392.0

Step 25: Merge Document 163.0 and Document 175.0

Step 26: Merge Document 125.0 and Document 143.0

Step 27: Merge Document 461.0 and Document 462.0

Step 28: Merge Document 379.0 and Document 380.0

Step 29: Merge Document 326.0 and Document 328.0

Step 30: Merge Document 465.0 and Document 466.0

Step 31: Merge Document 371.0 and Document 373.0

Step 32: Merge Document 67.0 and Document 198.0

Step 33: Merge Document 12.0 and Document 225.0

Step 34: Merge Document 450.0 and Document 452.0

Step 35: Merge Document 320.0 and Document 322.0

Step 36: Merge Document 173.0 and Document 193.0

Step 37: Merge Document 318.0 and Document 319.0

Step 38: Merge Document 44.0 and Document 222.0

Step 39: Merge Document 258.0 and Document 269.0

Step 40: Merge Document 98.0 and Document 112.0

Step 41: Merge Document 3.0 and Document 214.0

Step 42: Merge Document 311.0 and Document 315.0

Step 43: Merge Document 304.0 and Document 308.0

Step 44: Merge Document 250.0 and Document 256.0

Step 45: Merge Document 463.0 and Document 464.0

Step 46: Merge Document 456.0 and Document 457.0

Step 47: Merge Document 200.0 and Document 210.0

Step 48: Merge Document 454.0 and Document 474.0

Step 49: Merge Document 30.0 and Document 38.0

Step 50: Merge Document 34.0 and Document 50.0

Step 51: Merge Document 476.0 and Document 478.0

Step 52: Merge Document 238.0 and Document 335.0

Step 53: Merge Document 56.0 and Document 72.0

Step 54: Merge Document 470.0 and Document 473.0

Step 55: Merge Document 317.0 and Document 321.0

Step 56: Merge Document 127.0 and Document 165.0

Step 57: Merge Document 281.0 and Document 284.0

Step 58: Merge Document 282.0 and Document 286.0

Step 59: Merge Document 55.0 and Document 71.0

Step 60: Merge Document 243.0 and Document 246.0

Step 61: Merge Document 179.0 and Document 189.0

Step 62: Merge Document 446.0 and Document 447.0

Step 63: Merge Document 188.0 and Document 211.0

Step 64: Merge Document 395.0 and Document 396.0

Step 65: Merge Document 303.0 and Document 307.0

Step 66: Merge Document 279.0 and Document 287.0

Step 67: Merge Document 475.0 and Document 479.0

Step 68: Merge Document 291.0 and Document 313.0

Step 69: Merge Document 271.0 and Document 273.0

Step 70: Merge Document 298.0 and Document 299.0

Step 71: Merge Document 234.0 and Document 331.0

Step 72: Merge Document 116.0 and Document 123.0

Step 73: Merge Document 11.0 and Document 101.0

Step 74: Merge Document 338.0 and Document 365.0

Step 75: Merge Document 233.0 and Document 334.0

Step 76: Merge Document 346.0 and Document 368.0

Step 77: Merge Document 219.0 and Document 229.0

Step 78: Merge Document 192.0 and Document 212.0

Step 79: Merge Document 360.0 and Document 375.0

Step 80: Merge Document 208.0 and Document 218.0

Step 81: Merge Document 289.0 and Document 292.0

Step 82: Merge Document 270.0 and Document 272.0

Step 83: Merge Document 17.0 and Document 126.0

Step 84: Merge Document 160.0 and Document 162.0

Step 85: Merge Document 164.0 and Document 166.0

Step 86: Merge Document 355.0 and Document 370.0

Step 87: Merge Document 28.0 and Document 36.0

Step 88: Merge Document 358.0 and Document 377.0

Step 89: Merge Document 382.0 and Document 394.0

Step 90: Merge Document 168.0 and Document 178.0

Step 91: Merge Document 336.0 and Document 350.0

Step 92: Merge Document 49.0 and Document 57.0

Step 93: Merge Document 341.0 and Document 352.0

Step 94: Merge Document 20.0 and Document 93.0

Step 95: Merge Document 347.0 and Document 367.0

Step 96: Merge Document 385.0 and Document 386.0

Step 97: Merge Document 348.0 and Document 376.0

Step 98: Merge Document 251.0 and Document 259.0

Step 99: Merge Document 87.0 and Document 94.0

Step 100: Merge Document 134.0 and Document 201.0