Krish Kabra, 404593821
Ranjana Vittal, 605629219
Zhiyuan Chen, 605363281

Project 2: Clustering

EE 219 - Large-Scale Data Mining: Models and Algorithms
Winter 2021

2021-02-02

Part 1 - Clustering of Text Data

Basic Workflow

Question 1: Building the TF-IDF matrix

The dimension of the TF-IDF matrix is $(7822, 23522)$

Question 2: K-means clustering

The contingency table of the clustering result on the dataset

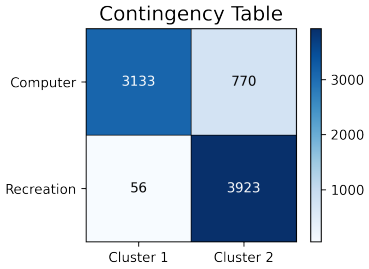

Figure 1: Contingency Matrix

Question 3: Clustering metrics

The five measures for the K-means clustering results:

Homogeneity	Completeness	V-measure	Adjusted Rand Index	Adjusted Mutual Info.
0.565	0.580	0.573	0.625	0.572

Table 1: K-means measures

Dimensionality Reduction and Data Transformation

Question 4: Percent variance of principal components

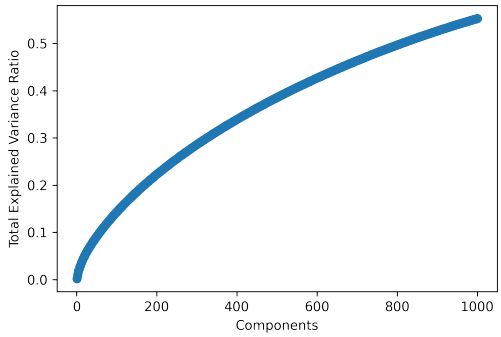

Figure 2: Total Explained Variance Ratio

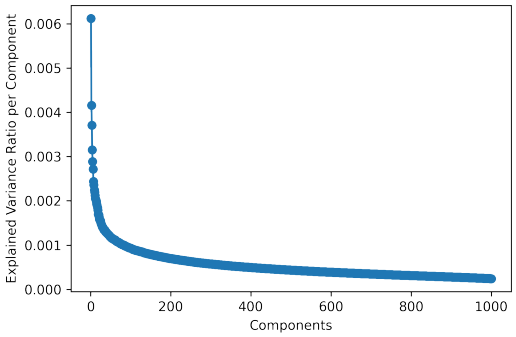

Figure 3: Explained Variance Ratio per Component

Question 5: Truncated SVD vs. NMF

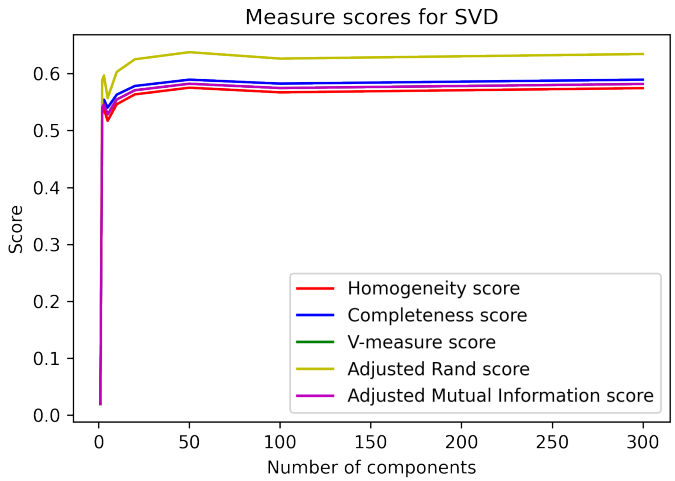

Figure 4: SVD Metrics Graph

```
SVD
[0.019034591198472048, 0.5318527560716836, 0.5377651779879102, 0.517036422502093, 0.5460921220112308, 0.563426312706913, 0.57516931569769, 0.5670330934635874, 0.5744113629390132]
[0.0193604608365066627, 0.5491344627077224, 0.5542779518154466, 0.5401602853893641, 0.5629480411958336, 0.5782222768022195, 0.5893200079451415, 0.5823852599054565, 0.5892858802537997]
[0.01919614314371571, 0.5403554683557176, 0.545896720153317 4, 0.5283454620142469, 0.5543198795057 73, 0.5707284156217397, 0.5821586833212198, 0.5746066513447411, 0.5817535576669169]
[0.02600277723478503, 0.588725244819119, 0.5965400007596544, 0.5568446089604012, 0.6028289245085969, 0.6250997063911322, 0.6376018619358004, 0.6263041993292062, 0.6343636646220453]
[0.01910558756050995, 0.5403127144748595, 0.5458545177246293, 0.5283013366507997, 0.5543505718649027, 0.5706886059370165, 0.5821199646013246, 0.5745671858723812, 0.5817147772526922]
```

Figure 5: SVD Metrics Values

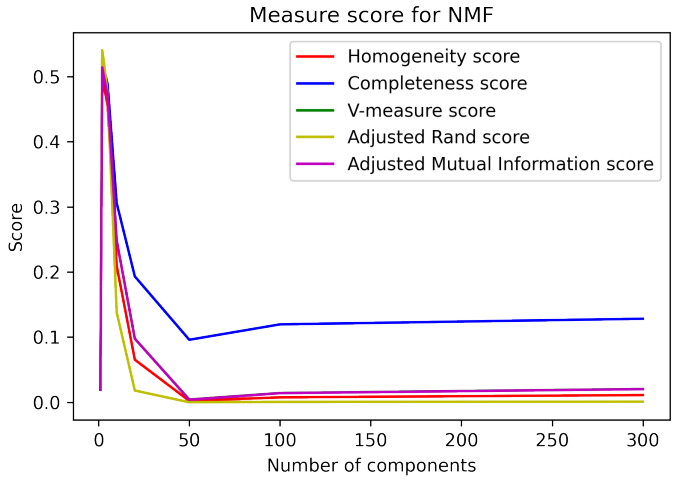

Figure 6: NMF Metrics Graph

NMF
[0.019034591198472048, 0.5022478335120428, 0.48609683035120665, 0.45500660137619775, 0.20915909036595745, 0.06529404662847932, 0.0021303952018709594, 0.007502018996706376, 0.011022010519163693]
[0.019360460836506627, 0.5263843051392446, 0.5118300127229015, 0.4881474012058305, 0.30563865746695934, 0.1931811334155371, 0.09591019434272455, 0.11959893945551682, 0.12823289117096448]
[0.01919614314371571, 0.5140328926482289, 0.49863163535474486, 0.47099474610769254, 0.248358132278426, 0.09759991602155817, 0.004168204593370178, 0.014118438235357446, 0.020299239139663821]
[0.02600277723478503, 0.5406769711846864, 0.521443534760384, 0.4724383437991488, 0.1378993331182748, 0.017986724053573 5, -6.487642756916324e-05, 0.0004994593822397486, 0.000879750035061344]
[0.01910558756050995, 0.51398735739796, 0.4985845503268862, 0.47094461230034934, 0.248276407317059 72, 0.09747631727528892, 0.0039841010495616416, 0.013947096434058 34, 0.020133084769439 05]

Figure 7: NMF Metrics Values

The first row of each dimensionality reduction algorithm corresponds to homogeneity score. The second row corresponds to completeness score. The third row corresponds to V-measure score. The fourth row corresponds to adjusted Rand Index. The fifth row corresponds to adjusted mutual information score. As it can be seen from the two graphs, some values of r give the best scores for all the measures across the board. In the case of SVD, r=50 gave the best result for all measure scores. In the case of NMF, r=2 gave the best result for all measure scores as well. Thus, r=50 was chosen for SVD and r=2 for NMF.

Question 6: Non-monotonic behavior of percent variance

There is a non-monotonic behavior in the measures as r increases. As the number of components increases, the dimensions in which k-means needs to perform clustering increases. K-means suffers from the curse of dimensionality because the Euclidean distance is not a good metric in high dimensions since the ratio between the nearest and farthest points approaches 1. This means that points in high dimensions are essentially equidistant from each other which makes it hard to perform clustering. As a result, increasing the number of components after the “Elbow point” adds no new information. Since no new information is given to the k-means algorithm, the measures remain constant and do not change.

Question 7: Visualizing clustering results after dimensionality reduction

For SVD, r=50 is chosen and for NMF, r=2 is chosen.

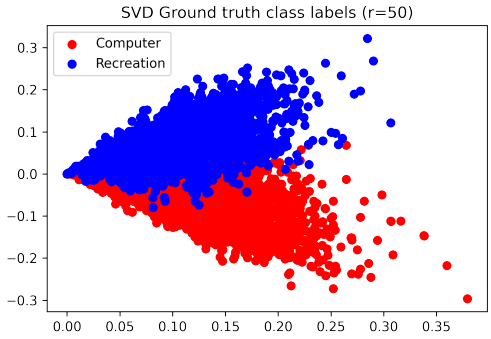

Figure 8: SVD Ground Truth Class Labels

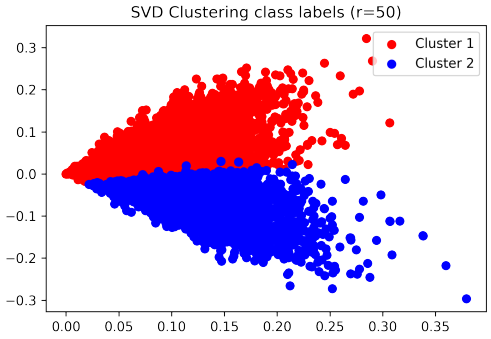

Figure 9: SVD Clustering Class Labels

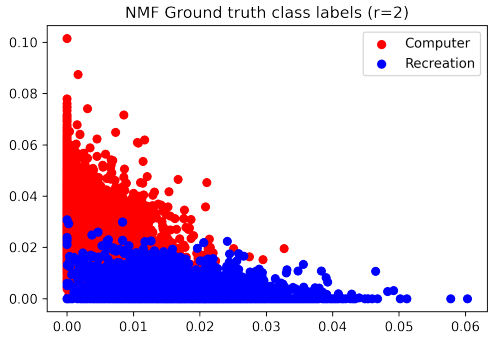

Figure 10: NMF Ground Truth Class Labels

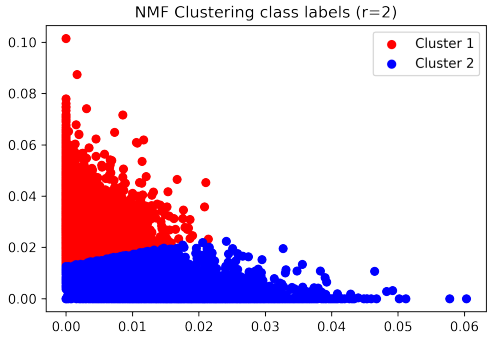

Figure 11: NMF Clustering Class Labels

Question 8: Analyzing clustering results after dimensionality reduction

In the visualization, the data points distributed in the clustering class and ground truth class are very close. While the ground truth plot would have category data points fall under another data range, the clustering class has a clear and straight boundary line, which monotonically divides the categories. For low dimensions, the data distribution is ideal for K-Means clustering.

Question 9: More challenging clustering

The dimension of the TF-IDF matrix is $(18846, 45365)$

Homogeneity	Completeness	V-measure	Adjusted Rand Index	Adjusted Mutual Info.
0.346	0.434	0.385	0.110	0.383

Table 2: Metrics for K-Means clustering for all 20 categories

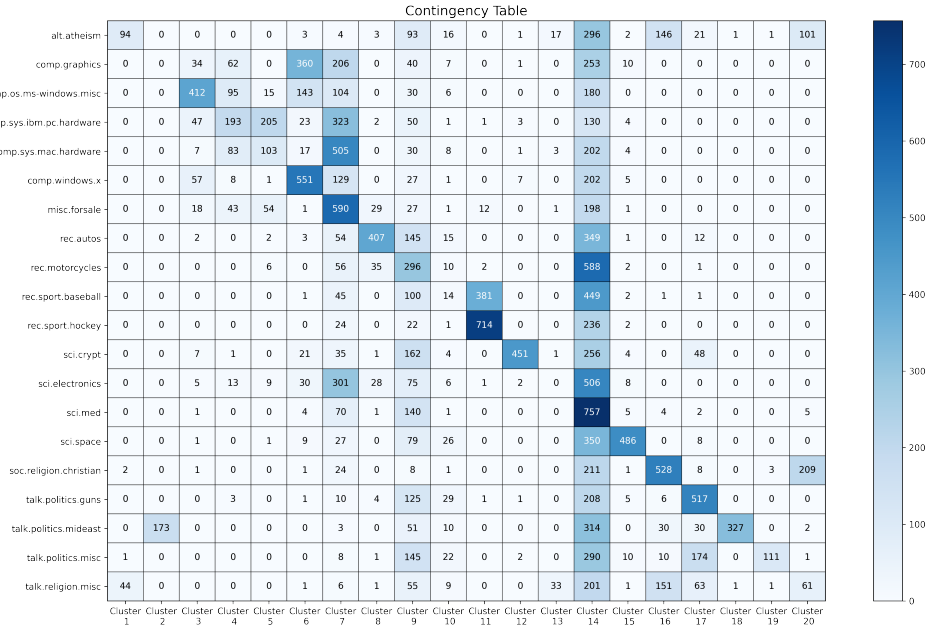

Figure 12: Contingency matrix for all 20 categories

Question 10: Kullback-Leibler Divergence for NMF

Homogeneity	Completeness	V-measure	Adjusted Rand Index	Adjusted Mutual Info.
0.191	0.205	0.198	0.058	0.195

Table 3: Metrics for K-Means clustering using NMF with Frobenius cost function.

Homogeneity	Completeness	V-measure	Adjusted Rand Index	Adjusted Mutual Info.
0.203	0.228	0.215	0.066	0.212

Table 4: Metrics for K-Means clustering using NMF with Kullback-Leibler divergence cost function.

From the above, it can be seen that using Kullback-Leibler divergence over Frobenius norm as the cost function for NMF gives better results in clustering the text data.

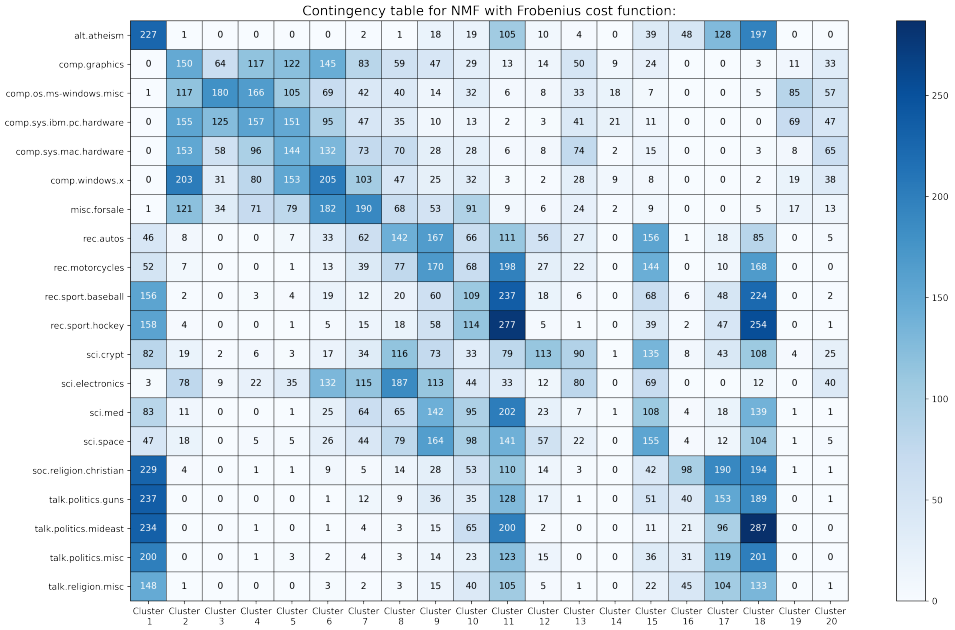

Figure 13: Contingency table for K-Means clustering using NMF with Frobenius cost function.

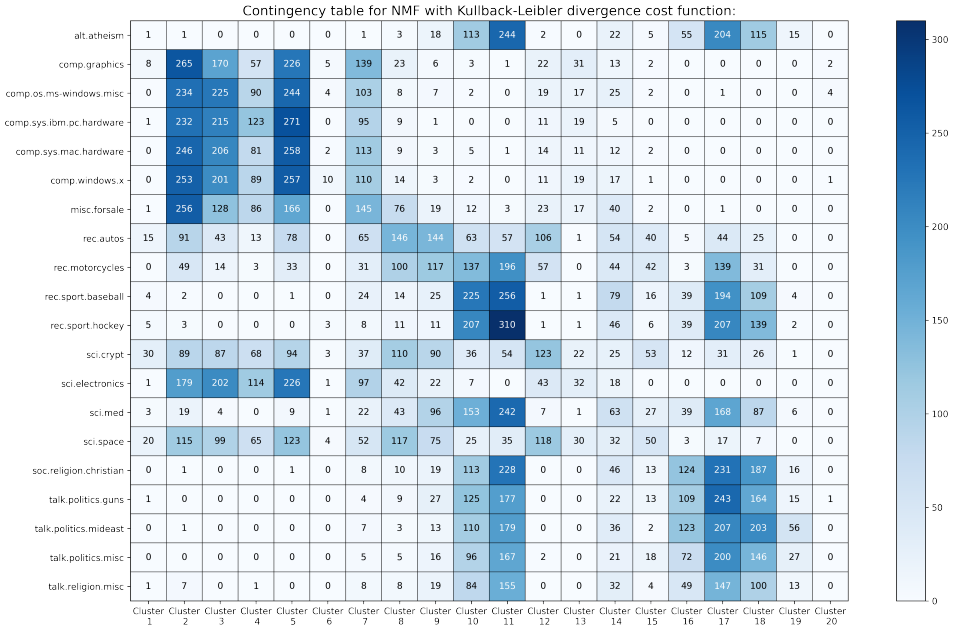

Figure 14: Contingency table for K-Means clustering using NMF with Kullback-Leibler divergence cost function.

Question 11: UMAP - Results

Homogeneity	Completeness	V-measure	Adjusted Rand Index	Adjusted Mutual Info.
0.008	0.008	0.008	0.002	0.005

Table 5: Metrics for K-Means clustering using UMAP dimensionality reduction with Euclidean distance.

UMAP is used to reduce the dimensionality of the 20 categories of the TF-IDF matrix to `n_components` $= 30$. The contingency tables and clustering evaluation metrics using the Euclidean distance and cosine similarity metrics are shown in Figures 15 and 16 and Tables 5 and 6 respectively.

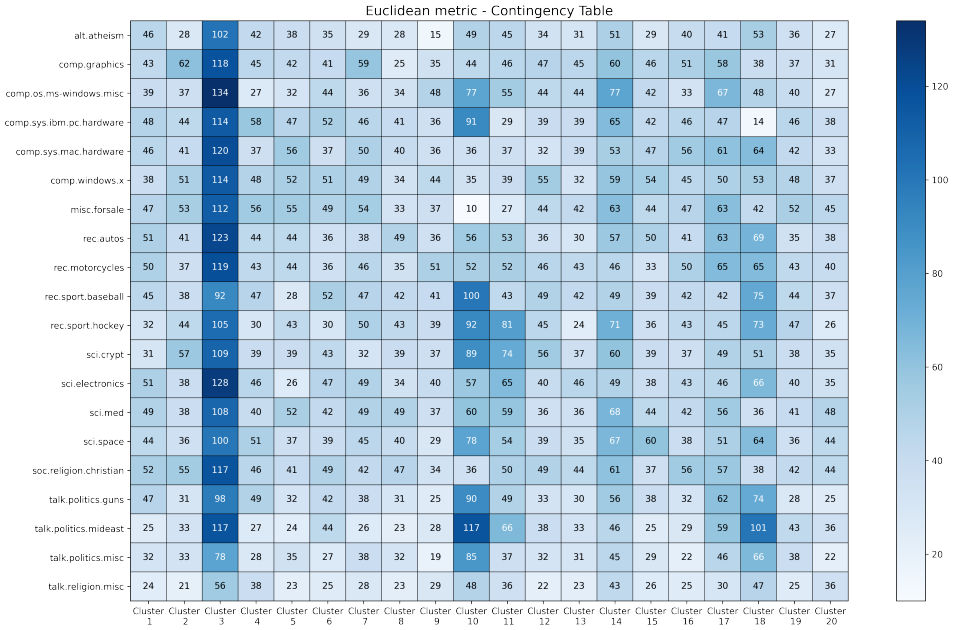

Figure 15: Contingency table for K-Means clustering using UMAP dimensionality reduction with Euclidean distance.

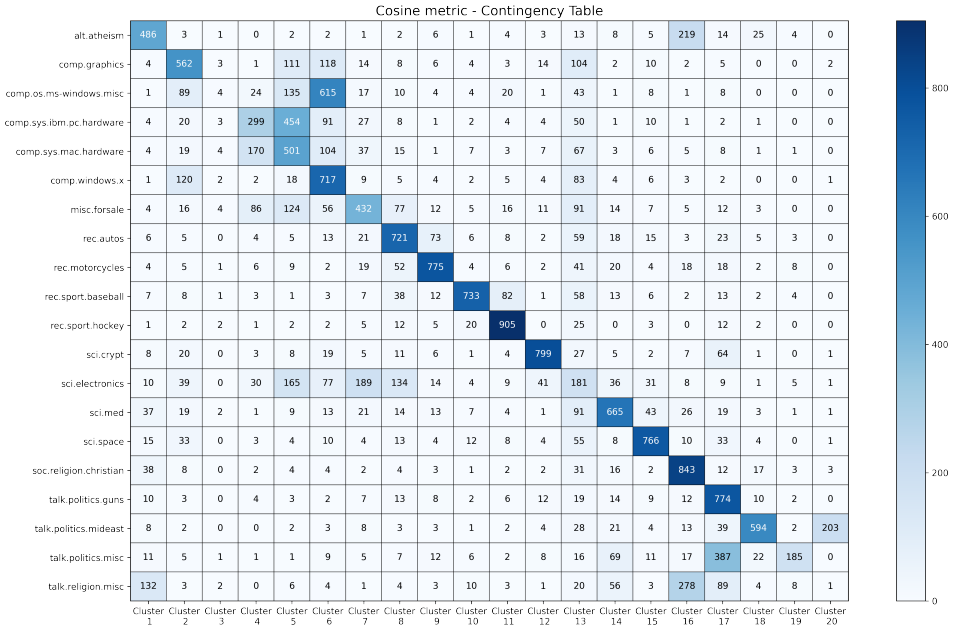

Figure 16: Contingency table for K-Means clustering using UMAP dimensionality reduction with cosine similarity.

Homogeneity	Completeness	V-measure	Adjusted Rand Index	Adjusted Mutual Info.
0.527	0.551	0.539	0.407	0.537

Table 6: Metrics for K-Means clustering using UMAP dimensionality reduction with cosine similarity.

We can clearly observe that the Euclidean distance metric performs far worse than the cosine similarity metric. This is likely because the Euclidean distance metric does not ignore the magnitude of the vectors, causing documents of different lengths to be placed at vastly different locations.

Question 12: UMAP - Analysis

Observing the contingency table using the cosine similarity metric (Figure 16), we can see the related category pairs `comp.os.ms-windows.misc` & `comp.windows.x`, `comp.sys.ibm.pc.hardware` & `comp.sys.mac.hardware`, `soc.religion.christian` & `talk.religion.misc`, and `talk.politics.guns` & `talk.politics.misc` are clustered together (see Clusters 5, 6, 16 and 17).

With regards to confusion, we see that no single cluster is formed for the `sci.electronics` category – instead, this category is largely split between clusters 5-8, which contain categories in `comp.sys.ibm.pc.hardware`, `comp.sys.mac.hardware`, `misc.forsale`, and `rec.autos`. These are all categories that generally relate to electronics. Clusters for the `comp` categories (clusters 2, 4, 5 and 6) contain errors either from related categories from other `comp` clusters or from the`misc.forsale` category. The confusion from the `misc.forsale` category is likely due to the fact that computer parts can be put for sale.

More Clustering Algorithms

Question 13: Agglomerative clustering

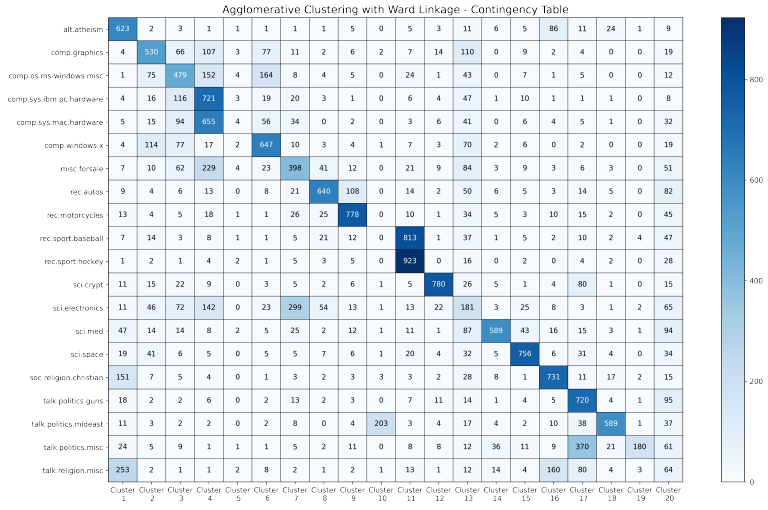

Figure 17: Contingency table for agglomerative cluster with ward linkage

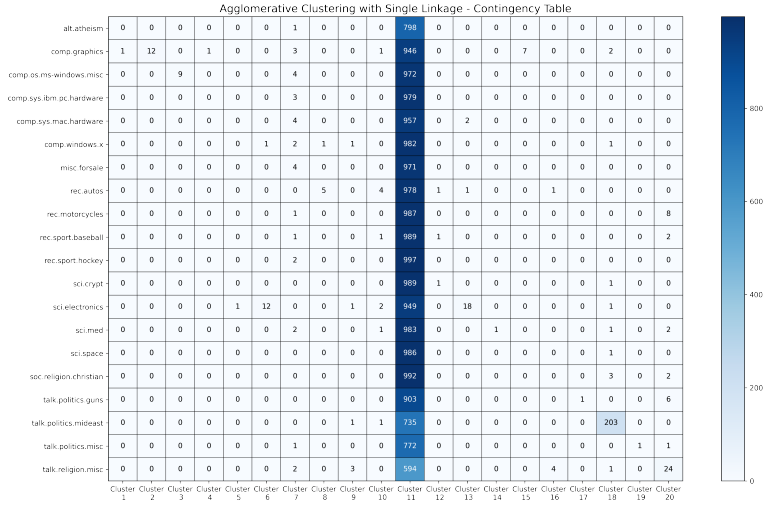

Figure 18: Contingency table for agglomerative cluster with single linkage

The contingency tables and clustering metrics for agglomerative clustering with ward and single linkage are shown in Figures 17 and 18 and Table 7. Using agglomerative clustering with single linkage performs poorly because the data is globular in nature.

	Homogeneity	Completeness	V-measure	Adjusted Rand Index	Adjusted Mutual Info.
Ward	0.520	0.547	0.533	0.396	0.532
Single	0.017	0.369	0.032	0.001	0.027

Table 7: Performance comparison for agglomerative clustering with ward and single linkage

Question 14: DBSCAN and HDBSCAN - Results

	Homogeneity	Completeness	V-measure	Adjusted Rand Index	Adjusted Mutual Info.
DBSCAN	0.454	0.557	0.500	0.280	0.499
HDBSCAN	0.397	0.604	0.479	0.226	0.478

Table 8: Performance Comparison for DBSCAN and HDBSCAN

The DBSCAN hyperparameters were set as follows: `eps`$= 0.6$, `min_samples`$= 100$.

The HDBSCAN hyperparameters were set as follows: `min_cluster_size`$= 100$, `min_samples`$= 15$, `clustering_selection_epsilon`$= 0.5$.

Question 15: DBSCAN and HDBSCAN - Analysis

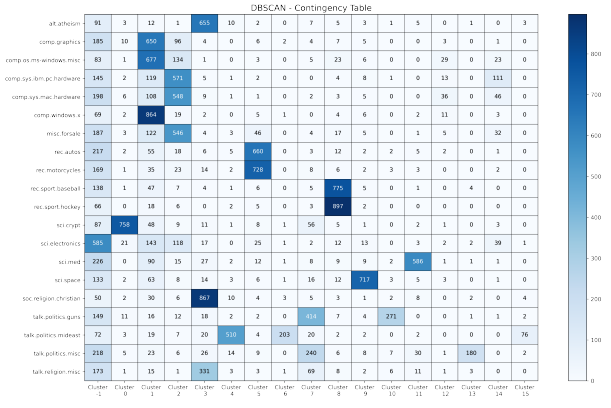

Figure 19: DBSCAN contingency matrix

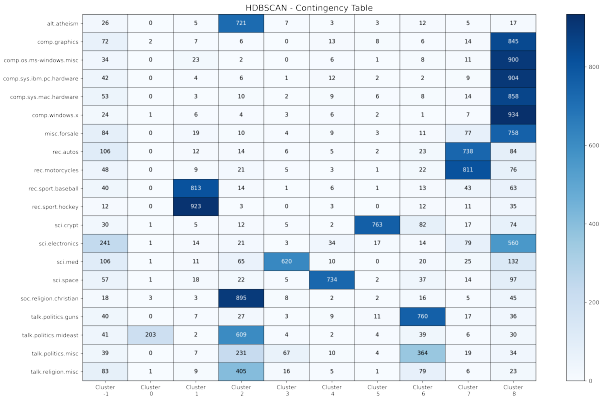

Figure 20: HDBSCAN contingency matrix

The contingency tables for DBSCAN and HDSCAN clustering are given in Figures 19 and 20 respectively. There are 16 clusters obtained using DBSCAN, and 9 clusters obtained using HDBSCAN. The cluster corresponding to label "-1" contains data points considered to be noisy i.e. points that do not have enough neighbors within range to form or be part of a cluster.

Both DBSCAN and HDBSCAN are able to successfully cluster related categories. For example, DBSCAN and HDBSCAN cluster the categories `rec.sport.baseball` and `rec.sport.hockey` together, as well as the categories `talk.politics.guns` and `talk.politics.misc`. Very interestingly, HDBSCAN clusters `talk.politics.mideast`, `soc.religion.christian` and `talk.religion.misc` categories together, whereas DBSCAN can separate the `talk.politics.mideast` category from the `soc.religion.christian` and `talk.religion.misc` categories. The more challenging categories from `comp` are clustered with documents from the `misc.forsale` category for both HDBSCAN and DBSCAN likely because many computer-related parts are posted for sale. Overall, given the aforementioned hyperparameters, DBSCAN is able to better separate related categories into different clusters in comparison to HDBSCAN; however, it consequently labels more data as noise.

Part 2 - BBCNews Dataset

Question 16: Clustering BBCNews Dataset

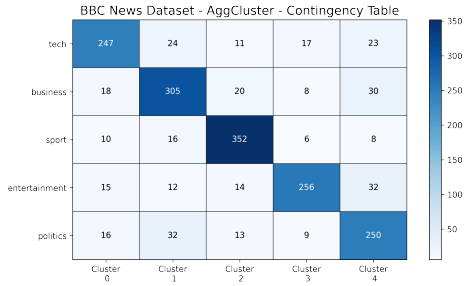

Figure 21: Contingency table for agglomerative clustering on BBC News dataset.

Homogeneity	Completeness	V-measure	Adjusted Rand Index	Adjusted Mutual Info.
0.543	0.544	0.544	0.586	0.542

Table 9: Metrics for Agglomerative clustering on BBC News dataset.

The data acquisition process is carried out in the functions `load_bbc_news` and `lemmatize_data`. The Python library `pandas` is utilized to easily collect the text and label for all the documents. Although the BBC News Classification dataset is pre-split into training and testing sets, we merge the sets as the clustering is performed completely unsupervised. The text data is filtered by removing punctuation, symbols and numbers, and through lemmatization.

The feature engineering process is achieved by constructing a TF-IDF matrix, and utilizing UMAP with a cosine similarity metric to reduce the dimension down to 30 components.

In order to cluster the data, we perform agglomerative clustering with ward linkage.

The performance of clustering is evaluated using a contingency table (Figure 21) and the homogeneity, completeness, V-measure, adjusted Rand index, and adjusted mutual information score metrics (Table 9).