```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly
remount, call drive.mount("/content/drive", force_remount=True).
```

```
import os
```

```
folder_path = "/content/drive/MyDrive/Natural Language
Processing/HW3/emb_data"
filenames = os.listdir(folder_path)
documents = []

for file in filenames:
  try:
    with open(os.path.join(folder_path, file), 'r', encoding='utf-8')
as f:
        documents.append(f.read())
  except:
    print(file)
```

```
article_100.txt
```

```
len(documents)
doc_str = ' '.join(documents)
```

```
!pip install sentence_transformers
```

```
Requirement already satisfied: sentence_transformers in
/usr/local/lib/python3.10/dist-packages (3.1.1)
Requirement already satisfied: transformers<5.0.0,>=4.38.0 in
/usr/local/lib/python3.10/dist-packages (from sentence_transformers)
(4.44.2)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-
packages (from sentence_transformers) (4.66.5)
Requirement already satisfied: torch>=1.11.0 in
/usr/local/lib/python3.10/dist-packages (from sentence_transformers)
(2.4.1+cu121)
Requirement already satisfied: scikit-learn in
/usr/local/lib/python3.10/dist-packages (from sentence_transformers)
(1.5.2)
Requirement already satisfied: scipy in
/usr/local/lib/python3.10/dist-packages (from sentence_transformers)
(1.13.1)
Requirement already satisfied: huggingface-hub>=0.19.3 in
/usr/local/lib/python3.10/dist-packages (from sentence_transformers)
(0.24.7)
Requirement already satisfied: Pillow in
/usr/local/lib/python3.10/dist-packages (from sentence_transformers)
(10.4.0)
Requirement already satisfied: filelock in
```

```
/usr/local/lib/python3.10/dist-packages (from huggingface-hub>=0.19.3-
>sentence_transformers) (3.16.1)
Requirement already satisfied: fsspec>=2023.5.0 in
/usr/local/lib/python3.10/dist-packages (from huggingface-hub>=0.19.3-
>sentence_transformers) (2024.6.1)
Requirement already satisfied: packaging>=20.9 in
/usr/local/lib/python3.10/dist-packages (from huggingface-hub>=0.19.3-
>sentence_transformers) (24.1)
Requirement already satisfied: pyyaml>=5.1 in
/usr/local/lib/python3.10/dist-packages (from huggingface-hub>=0.19.3-
>sentence_transformers) (6.0.2)
Requirement already satisfied: requests in
/usr/local/lib/python3.10/dist-packages (from huggingface-hub>=0.19.3-
>sentence_transformers) (2.32.3)
Requirement already satisfied: typing-extensions>=3.7.4.3 in
/usr/local/lib/python3.10/dist-packages (from huggingface-hub>=0.19.3-
>sentence_transformers) (4.12.2)
Requirement already satisfied: sympy in
/usr/local/lib/python3.10/dist-packages (from torch>=1.11.0-
>sentence_transformers) (1.13.3)
Requirement already satisfied: networkx in
/usr/local/lib/python3.10/dist-packages (from torch>=1.11.0-
>sentence_transformers) (3.3)
Requirement already satisfied: jinja2 in
/usr/local/lib/python3.10/dist-packages (from torch>=1.11.0-
>sentence_transformers) (3.1.4)
Requirement already satisfied: numpy>=1.17 in
/usr/local/lib/python3.10/dist-packages (from
transformers<5.0.0,>=4.38.0->sentence_transformers) (1.26.4)
Requirement already satisfied: regex!=2019.12.17 in
/usr/local/lib/python3.10/dist-packages (from
transformers<5.0.0,>=4.38.0->sentence_transformers) (2024.9.11)
Requirement already satisfied: safetensors>=0.4.1 in
/usr/local/lib/python3.10/dist-packages (from
transformers<5.0.0,>=4.38.0->sentence_transformers) (0.4.5)
Requirement already satisfied: tokenizers<0.20,>=0.19 in
/usr/local/lib/python3.10/dist-packages (from
transformers<5.0.0,>=4.38.0->sentence_transformers) (0.19.1)
Requirement already satisfied: joblib>=1.2.0 in
/usr/local/lib/python3.10/dist-packages (from scikit-learn-
>sentence_transformers) (1.4.2)
Requirement already satisfied: threadpoolctl>=3.1.0 in
/usr/local/lib/python3.10/dist-packages (from scikit-learn-
>sentence_transformers) (3.5.0)
Requirement already satisfied: MarkupSafe>=2.0 in
/usr/local/lib/python3.10/dist-packages (from jinja2->torch>=1.11.0-
>sentence_transformers) (2.1.5)
Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.10/dist-packages (from requests->huggingface-
hub>=0.19.3->sentence_transformers) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in
/usr/local/lib/python3.10/dist-packages (from requests->huggingface-
hub>=0.19.3->sentence_transformers) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/usr/local/lib/python3.10/dist-packages (from requests->huggingface-
hub>=0.19.3->sentence_transformers) (2.2.3)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.10/dist-packages (from requests->huggingface-
hub>=0.19.3->sentence_transformers) (2024.8.30)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in
/usr/local/lib/python3.10/dist-packages (from sympy->torch>=1.11.0-
>sentence_transformers) (1.3.0)
```

```python
from sentence_transformers import SentenceTransformer

model = SentenceTransformer('all-MiniLM-L6-v2')
embeddings = model.encode(documents)
```

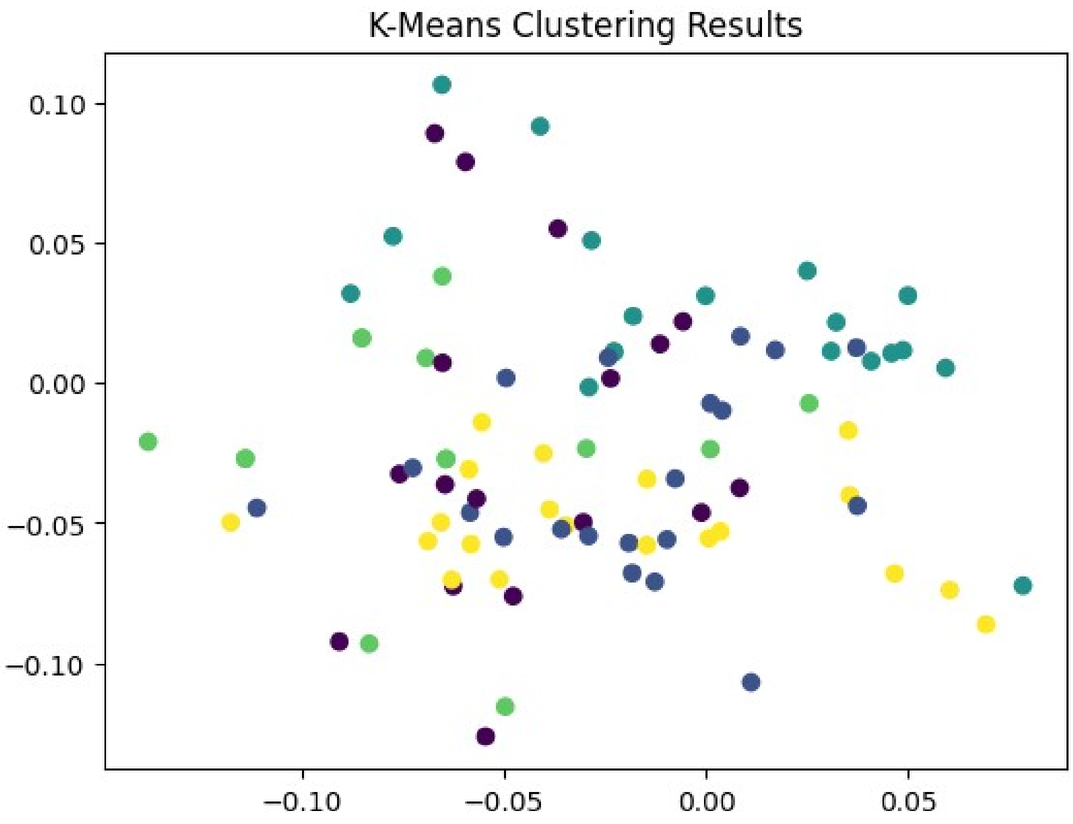
```
/usr/local/lib/python3.10/dist-packages/transformers/
tokenization_utils_base.py:1601: FutureWarning:
`clean_up_tokenization_spaces` was not set. It will be set to `True`
by default. This behavior will be depracted in transformers v4.45, and
will be then set to `False` by default. For more details check this
issue: https://github.com/huggingface/transformers/issues/31884
  warnings.warn(
```

I chose all-MiniLM-L6-v2, a BERT-based model, because it offers a great balance between efficiency and performance. It's lightweight, fast, and ideal for generating high-quality, contextual embeddings for large document corpora. Despite its small size, it captures semantic similarities well, making it suitable for tasks like document clustering. Its pre-training for sentence and document embeddings ensures good scalability and effectiveness in resource-constrained environments like Colab.

```python
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
from sklearn.metrics.pairwise import cosine_similarity

num_clusters = 5

kmeans = KMeans(n_clusters=num_clusters, random_state=42)
kmeans.fit(embeddings)

labels = kmeans.labels_

silhouette_avg = silhouette_score(embeddings, labels)
print(f'K-Means Silhouette Score: {silhouette_avg}')

plt.scatter(embeddings[:, 0], embeddings[:, 1], c=labels,
cmap='viridis')
```

```
plt.title('K-Means Clustering Results')
plt.show()
```

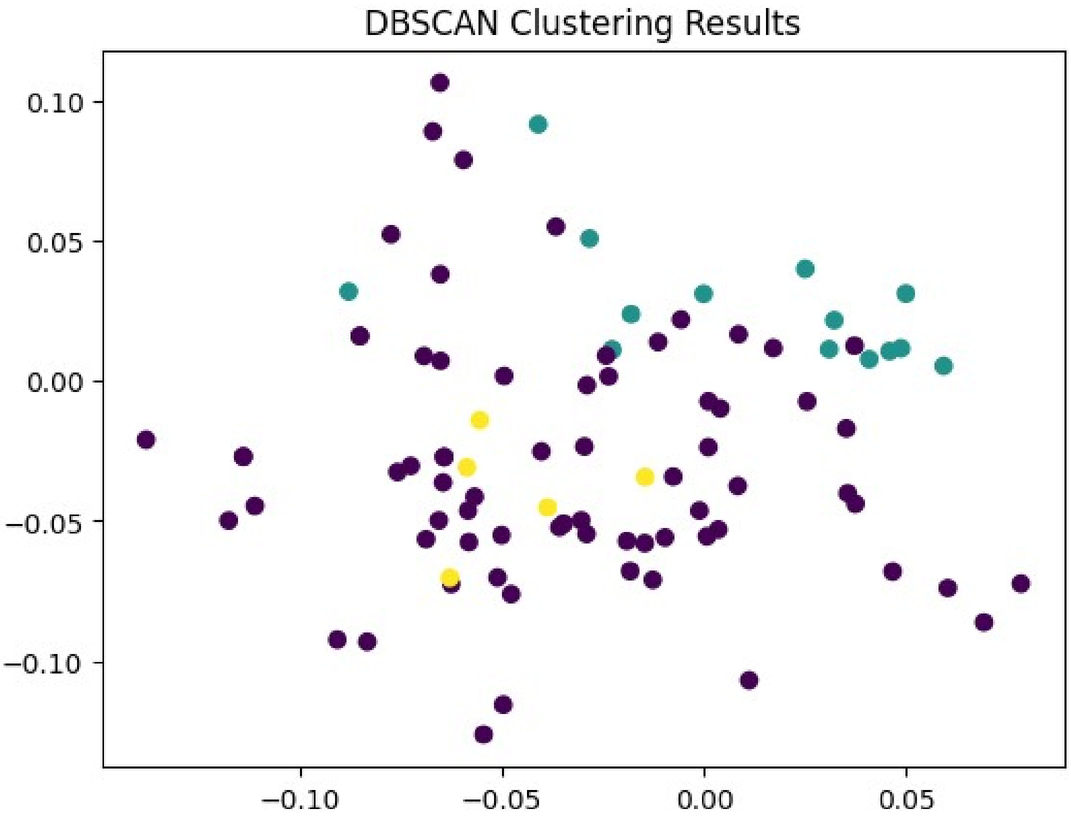
```
K-Means Silhouette Score: 0.12284903973340988
```



I chose cosine similarity because it measures how similar two documents are based on their content, regardless of their size or length. It's a simple and effective way to group documents that are closely related in meaning, which makes it ideal for clustering tasks like this one.

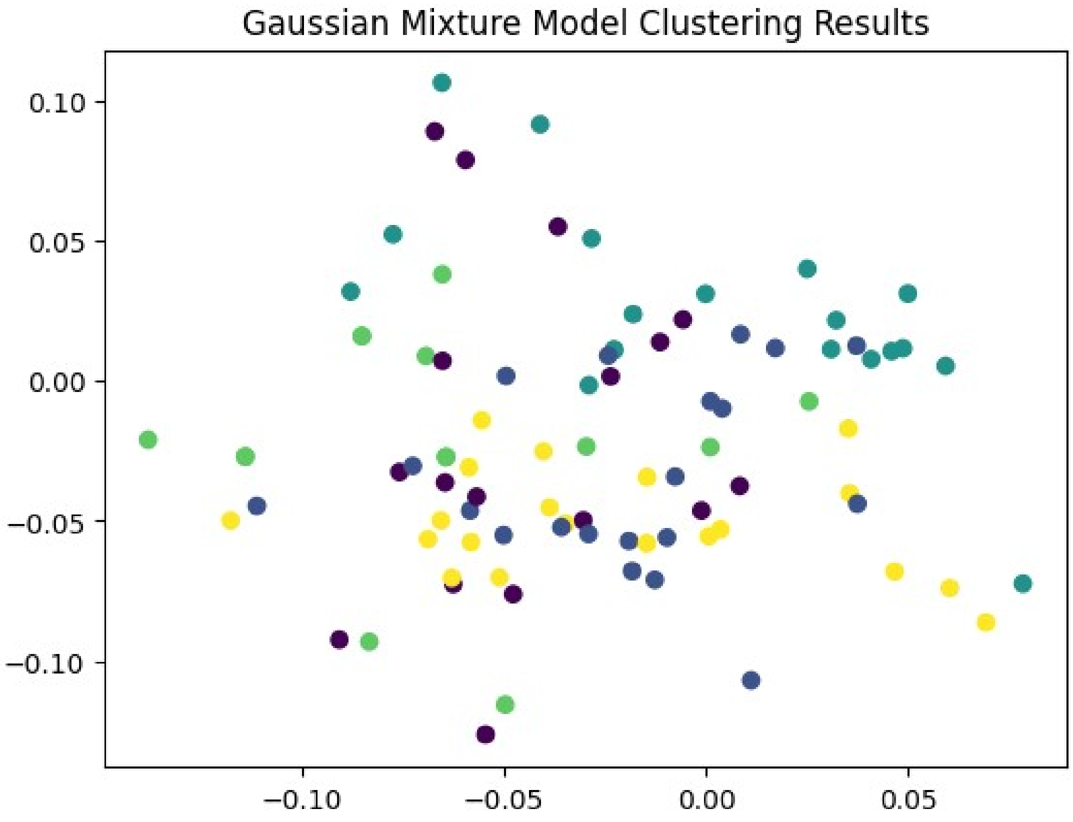
```
from sklearn.cluster import DBSCAN

dbscan = DBSCAN(eps=0.5, min_samples=5, metric='cosine')  # 'eps' and
'min_samples' can be tuned
dbscan_labels = dbscan.fit_predict(embeddings)

dbscan_silhouette_score = silhouette_score(embeddings, dbscan_labels)
print(f'DBSCAN Silhouette Score: {dbscan_silhouette_score}')

plt.scatter(embeddings[:, 0], embeddings[:, 1], c=dbscan_labels,
cmap='viridis')
plt.title('DBSCAN Clustering Results')
plt.show()
```

```
DBSCAN Silhouette Score: 0.04281741753220558
```



```
from sklearn.mixture import GaussianMixture

gmm = GaussianMixture(n_components=5, covariance_type='full',
random_state=42)
gmm_labels = gmm.fit_predict(embeddings)

gmm_silhouette_score = silhouette_score(embeddings, gmm_labels)
print(f'GMM Silhouette Score: {gmm_silhouette_score}')

plt.scatter(embeddings[:, 0], embeddings[:, 1], c=gmm_labels,
cmap='viridis')
plt.title('Gaussian Mixture Model Clustering Results')
plt.show()
```

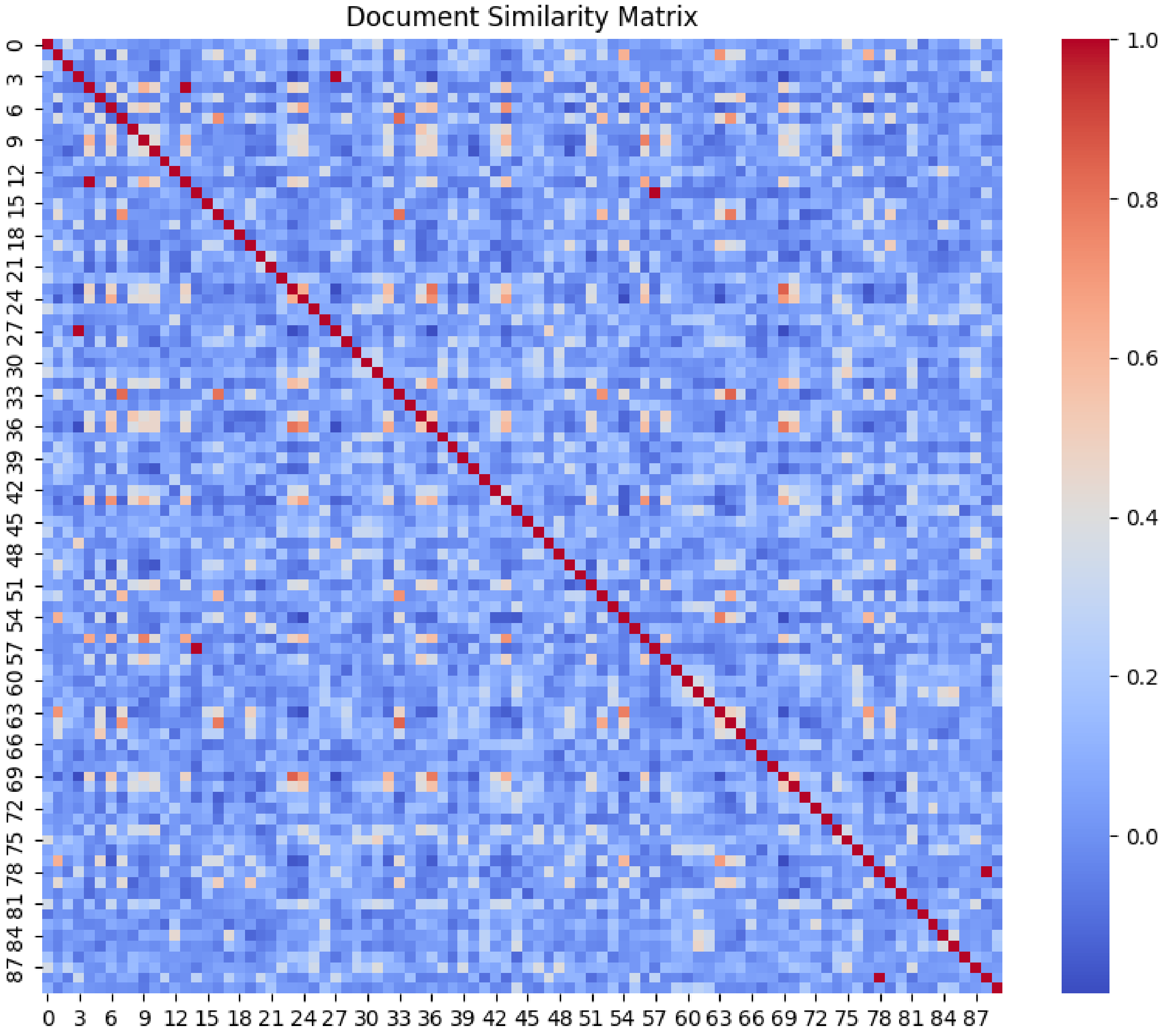
```
GMM Silhouette Score: 0.12284903973340988
```



```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

similarity_matrix = cosine_similarity(embeddings)
similarity_df = pd.DataFrame(similarity_matrix)
print(similarity_df)

plt.figure(figsize=(10, 8))
sns.heatmap(similarity_df, annot=False, cmap='coolwarm')
plt.title('Document Similarity Matrix')
plt.show()

          0         1         2         3         4         5
6   \
0   1.000000  0.043651  0.329527  0.030252  0.067896  0.044202
0.084180
1   0.043651  1.000000  0.109277  0.151102 -0.043264  0.357018 -
0.131528
2   0.329527  0.109277  1.000000 -0.030321 -0.062749  0.064486
0.074606
3   0.030252  0.151102 -0.030321  1.000000 -0.076568  0.046116 -
0.067596
```

```
4   0.067896 -0.043264 -0.062749 -0.076568  1.000000 -0.007720
0.452269
..       ...       ...       ...       ...       ...       ...       .
..
85  0.017698 -0.044869  0.049815  0.008985 -0.019509 -0.091058
0.072717
86  0.183678 -0.007777  0.124173 -0.080611  0.029962  0.070473
0.098416
87  0.356776  0.099581  0.282884  0.116816  0.002204 -0.008272
0.146582
88 -0.001036  0.059715 -0.064609  0.330057 -0.002670  0.062332 -
0.006251
89  0.108533 -0.047002  0.053265  0.088378  0.008183 -0.000291 -
0.015039

          7         8         9  ...        80        81        82
83  \
0   0.047567  0.016183 -0.009011  ...  0.037603  0.343042 -0.082673
0.097666
1   0.383585 -0.029208 -0.009600  ... -0.043318 -0.003457  0.008166
0.041348
2   0.106423  0.017195  0.022993  ...  0.063701  0.213822  0.007145
0.143186
3   0.133840  0.081855 -0.057708  ...  0.057514 -0.045854 -0.044630 -
0.021355
4  -0.050698  0.260163  0.612110  ... -0.105126  0.165703  0.049074 -
0.034708
..       ...       ...       ...  ...       ...       ...       ...
...
85  0.001277  0.009117  0.015971  ...  0.227672  0.042555  0.266522
0.158957
86 -0.009180  0.145628  0.107678  ...  0.023136  0.132571  0.075502
0.013444
87  0.038532  0.151473  0.059850  ... -0.025144  0.385573  0.024276
0.033127
88  0.089579  0.079662 -0.080880  ...  0.117392 -0.081734  0.034707
0.028315
89 -0.003309 -0.008884 -0.069414  ...  0.190337 -0.012629 -0.018447 -
0.041141

          84        85        86        87        88        89
0   0.029048  0.017698  0.183678  0.356776 -0.001036  0.108533
1  -0.042181 -0.044869 -0.007777  0.099581  0.059715 -0.047002
2   0.014774  0.049815  0.124173  0.282884 -0.064609  0.053265
3   0.038723  0.008985 -0.080611  0.116816  0.330057  0.088378
4   0.071727 -0.019509  0.029962  0.002204 -0.002670  0.008183
..       ...       ...       ...       ...       ...       ...
85  0.385321  1.000000  0.030636  0.061231  0.004859  0.205784
86  0.030254  0.030636  1.000000  0.139710  0.091644  0.011861
```

```
87  0.040914  0.061231  0.139710  1.000000  0.029602  0.033524
88 -0.000210  0.004859  0.091644  0.029602  1.000000  0.060520
89  0.234716  0.205784  0.011861  0.033524  0.060520  1.000000

[90 rows x 90 columns]
```



```
print(f'Number of filenames: {len(filenames)}')
print(f'Number of labels: {len(labels)}')
filenames.remove(file)
print(len(filenames))

Number of filenames: 91
Number of labels: 90
90
```

```python
import pandas as pd

output_df = pd.DataFrame({
    'filename': filenames,
    'category': labels
})

output_df.to_csv('document_clusters.csv', index=False)
```