Abhay Nath (CT_CSI_DS_511)

Aim: Applying clustering algorithms like Latent Dirichlet Allocation (LDA) or K-means to group similar documents together for topic modeling and understanding large text corpora.

Code:

```
# Created by Abhay Nath (CT_CSI_DS_511)
import os
import glob
import tarfile
import urllib.request
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.cluster import KMeans
from sklearn.manifold import TSNE
import matplotlib.pyplot as plt
import nltk
from nltk.corpus import stopwords

url = "https://archive.ics.uci.edu/ml/machine-learning-databases/20newsgroups-mld/20_newsgroups.tar.gz"
urllib.request.urlretrieve(url, '20_newsgroups.tar.gz')

with tarfile.open('20_newsgroups.tar.gz', 'r:gz') as tar:
    tar.extractall()

def load_data(path):
    documents = []
    labels = []
    for label in os.listdir(path):
        class_path = os.path.join(path, label)
        if os.path.isdir(class_path):
            for file_path in glob.glob(os.path.join(class_path, '*')):
                with open(file_path, 'r', encoding='latin1', errors='ignore') as file:
                    documents.append(file.read())
                    labels.append(label)
    return documents, labels

documents, labels = load_data('20_newsgroups')

nltk.download('stopwords')
stop_words = stopwords.words('english')

vectorizer = TfidfVectorizer(stop_words=stop_words, max_df=0.5, max_features=10000)
X = vectorizer.fit_transform(documents)

num_clusters = 20
```

```
km = KMeans(n_clusters=num_clusters, random_state=42)
km.fit(X)
clusters = km.labels_

def print_top_terms_per_cluster(vectorizer, km, num_terms=10):
    terms = vectorizer.get_feature_names_out()
    for i in range(num_clusters):
        print(f"Cluster {i}:")
        cluster_terms = km.cluster_centers_[i].argsort()[-num_terms:]
        print(" ".join(terms[cluster_terms]))

print_top_terms_per_cluster(vectorizer, km)

tsne = TSNE(n_components=2, perplexity=30, random_state=42)
X_tsne = tsne.fit_transform(X.toarray())

plt.figure(figsize=(12, 8))
scatter = plt.scatter(X_tsne[:, 0], X_tsne[:, 1], c=clusters, cmap='viridis',
marker='o', s=50)
plt.colorbar(scatter, ticks=range(num_clusters))
plt.title('Visualization of 20 Newsgroups clusters')
plt.xlabel('feature 1')
plt.ylabel('feature 2')
plt.show()
```

Output:

```
Cluster 0:
objective de frank horus ap apple mchp sni sgi sandvik
Cluster 1:
apana comp cc uwa monu6 australia munnari monash oz au
Cluster 2:
university mechalas engineering sage misc cc noose mentor ecn purdue
Cluster 3:
n3jxp dsl uucp blue med pittsburgh gordon banks geb pitt
Cluster 4:
sci host posting one nntp would cc ohio rec state
Cluster 5:
system monitor ohio state se apple hardware comp sys mac
Cluster 6:
uwo sfu carleton uwaterloo bc canada hockey ubc bnr ca
Cluster 7:
motif microsoft file apps dos misc comp ms os windows
Cluster 8:
western reserve usenet po sw freenet ins stratus cleveland cwru
Cluster 9:
cv boi sc hpscit packard hewlett sdd apollo col hp
Cluster 10:
mideast armenia politics zuma armenians soviet armenian turkish soc culture
Cluster 11:
udel shipping offer ohio 00 state computers sale misc forsale
Cluster 12:
card ide austin drive scsi hardware comp sys pc ibm
Cluster 13:
nz comp za mantis pipex uknet demon co ac uk
Cluster 14:
toronto alaska astro henry access digex gov sci nasa space
Cluster 15:
alexia owner noise uxa ux1 urbana cobb illinois cso uiuc
Cluster 16:
dsg 94305 slac andy agate csd newshost headwall leland stanford
Cluster 17:
sci chip netcom encryption eff key security org crypt clipper
```

```
Cluster 18:
ohio would state religion people guns alt misc politics talk
Cluster 19:
aramis hedrick igor soc religion god geneva athos christian rutgers
```