```
# Install necessary resources
import nltk
nltk.download('stopwords')

# Import libraries
import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import fetch_20newsgroups
from sklearn.feature_extraction.text import TfidfVectorizer,CountVectorizer
from sklearn.cluster import KMeans
from sklearn.decomposition import LatentDirichletAllocation
from sklearn.metrics import homogeneity_score,completeness_score,normalized_mutual_info_sco
from sklearn.manifold import TSNE

# Step 1: Load Dataset
print("Loading dataset...")
newsgroups=fetch_20newsgroups(remove=('headers','footers','quotes'))
documents=newsgroups.data
true_labels=newsgroups.target

# Step 2: TF-IDF Vectorization for K-Means
print("Vectorizing documents using TF-IDF...")
tfidf_vectorizer=TfidfVectorizer(stop_words='english',max_df=0.5,min_df=2)
tfidf=tfidf_vectorizer.fit_transform(documents)

# Step 3: Apply K-Means Clustering
print("Applying K-Means clustering...")
num_clusters=20
kmeans_model=KMeans(n_clusters=num_clusters,random_state=42)
kmeans_model.fit(tfidf)

# Step 4: Display Top Terms per KMeans Cluster
print("\nTop terms per KMeans cluster:")
terms=tfidf_vectorizer.get_feature_names_out()
for i in range(num_clusters):
    top_indices=kmeans_model.cluster_centers_[i].argsort()[-10:][::-1]
    top_terms=[terms[ind] for ind in top_indices]
    print(f"Cluster {i}:{top_terms}")

# Step 5: Evaluate Clustering
print("\nEvaluation Metrics for K-Means:")
print("Homogeneity Score:",homogeneity_score(true_labels,kmeans_model.labels_))
print("Completeness Score:",completeness_score(true_labels,kmeans_model.labels_))
print("NMI Score:",normalized_mutual_info_score(true_labels,kmeans_model.labels_))

# Step 6: Count Vectorization for LDA
print("\nVectorizing documents using CountVectorizer for LDA...")
count_vectorizer=CountVectorizer(stop_words='english',max_df=0.9,min_df=2)
doc_term_matrix=count_vectorizer.fit_transform(documents)
```

```
# Step 7: Apply Latent Dirichlet Allocation (LDA)
print("Applying Latent Dirichlet Allocation (LDA)...")
lda_model=LatentDirichletAllocation(n_components=20,random_state=42)
lda_model.fit(doc_term_matrix)

# Step 8: Display Top Terms per LDA Topic
print("\nTop terms per LDA topic:")
terms=count_vectorizer.get_feature_names_out()
for idx,topic in enumerate(lda_model.components_):
    top_terms=[terms[i] for i in topic.argsort()[-10:]]
    print(f"Topic {idx}:{top_terms}")

# Step 9: Visualize KMeans Clusters with t-SNE
print("\nVisualizing KMeans clusters using t-SNE (this may take a few minutes)...")
tsne_model=TSNE(n_components=2,perplexity=50,n_iter=300,random_state=42)
tsne_result=tsne_model.fit_transform(tfidf.toarray())

# Plot the t-SNE results
plt.figure(figsize=(12,6))
scatter=plt.scatter(tsne_result[:,0],tsne_result[:,1],c=kmeans_model.labels_,cmap='tab20',s
plt.title("KMeans Clustering (t-SNE Visualization)")
plt.xlabel("t-SNE Dimension 1")
plt.ylabel("t-SNE Dimension 2")
plt.colorbar(scatter,label='Cluster ID')
plt.tight_layout()
plt.show()
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
Loading dataset...
Vectorizing documents using TF-IDF...
Applying K-Means clustering...

Top terms per KMeans cluster:
Cluster 0: ['mileage', 'highway', 'covers', 'charles', 'affect', 'travel', 'comment',
Cluster 1: ['cda', 'warrenty', 'educational', 'repair', 'sales', 'emptor', 'tax', 'or
Cluster 2: ['objective', 'objectively', 'society', 'ideal', 'blame', 'innocent', 'mur
Cluster 3: ['visualization', 'video', 'frame', 'software', '8mm', 'rs6000', 'vhs', 'e
Cluster 4: ['aa', 'mle', 'aaa', 'mitchell', 'stat', 'keith', 'improve', 'does', 'move
Cluster 5: ['just', 'like', 'know', 'don', 'people', 'does', 'think', 'use', 'thanks'
Cluster 6: ['jagr', 'francis', 'bowman', 'ice', 'points', 'jaro', 'jaromir', 'better'
Cluster 7: ['limit', 'drive', 'hard', 'power', 'tell', 'pds', 'upgrades', 'limitation
Cluster 8: ['empire', 'constantinople', 'sack', 'battle', 'hordes', 'heretics', 'anat
Cluster 9: ['glock', 'revolver', 'dangerous', 'moment', 'hrumph', 'safeties', 'hammer
Cluster 10: ['memes', 'flawless', 'cosmetic', 'genes', 'harmful', 'units', 'store', '
Cluster 11: ['bought', 'earphone', 'reciever', '901', '910', 'onkyo', 'zeos', 'old',
Cluster 12: ['brian', 'curran', 'meaddata', 'mead', 'central', 'data', 'com', 'seatir
Cluster 13: ['dad', 'dream', 'told', 'buy', 'car', 'true', 'come', 'probably', 'dwex'
Cluster 14: ['dxf', 'doc', 'updated', 'location', 'host', 'rw', 'file', '1991', 'pub'
Cluster 15: ['joe', 'months', 'big', 'little', 'mercury', 'henry', 'film', 'built', '
Cluster 16: ['caste', 'profiting', 'pgf', 'srl03', 'blare', 'cacs', 'fraering', 'twil
Cluster 17: ['chevrolet', 'lemon', 'transmission', 'defective', '1992', 'idaho', 'rap
Cluster 18: ['testing', 'openvms', 'rerun', 'record', 'regression', 'verification', '
Cluster 19: ['argument', 'evidence', 'astounding', 'adda', 'supportable', 'fluffy', '

Evaluation Metrics for K-Means:
Homogeneity Score:  0.0023701646173742716
Completeness Score: 0.3005010652926979
NMI Score:          0.004703233071371646

Vectorizing documents using CountVectorizer for LDA...
Applying Latent Dirichlet Allocation (LDA)...

Top terms per LDA topic:
Topic 0: ['teams', 'league', 'players', 'year', 'hockey', 'play', 'season', 'games',
Topic 1: ['new', 'people', 'information', 'mail', 'privacy', 'anonymous', 'use', 'pub
Topic 2: ['send', 'pub', 'graphics', 'software', 'com', 'available', 'mail', 'ftp', '
Topic 3: ['1t', '0t', '1d9', '145', 'pl', 'a86', 'b8f', 'g9v', 'max', 'ax']
Topic 4: ['conference', 'paradox', 'captain', 'germany', 'type', 'phone', 'version',
Topic 5: ['w1', 'hz', 'lk', 'ck', 't7', 'chz', 'uw', 'c_', 'w7', 'cx']
Topic 6: ['know', 'think', 'say', 'christian', 'bible', 'people', 'does', 'believe',
Topic 7: ['13', '14', '16', '20', '12', '15', '11', '25', '00', '10']
Topic 8: ['time', 'government', 'know', 'right', 'make', 'like', 'just', 'think', 'do
Topic 9: ['does', 'bit', 'like', 'disk', 'problem', 'scsi', 'windows', 'card', 'use',
Topic 10: ['files', 'help', 'advance', 'hi', 'file', 'does', 'like', 'windows', 'know
Topic 11: ['data', '10', 'center', 'use', 'research', 'earth', 'launch', '1993', 'nas
Topic 12: ['clipper', 'said', 'know', 'chip', 'number', 'keys', 'stephanopoulos', 'pr
Topic 13: ['palestinian', 'arabs', 'jews', 'pope', 'church', 'land', 'state', 'arab',
Topic 14: ['set', 'use', 'server', 'widget', 'motif', 'output', 'entry', 'program', '
Topic 15: ['medical', 'people', 'just', 'edu', 'patients', 'disease', 'doctor', 'don'
Topic 16: ['went', 'didn', 'killed', 'war', 'jews', 'turkish', 'armenians', 'said', '
Topic 17: ['government', 'technology', 'need', 'used', 'phone', 'new', 'chip', 'encry
Topic 18: ['m0', 'mw', 'mt', 'mu', 'mv', 'mk', 'mr', 'mj', 'mm', 'mp']
```

```
Topic 19: ['time', 've', 'got', 'year', 'think', 'don', 'like', 'car', 'just', 'good'

Visualizing KMeans clusters using t-SNE (this may take a few minutes)...
/usr/local/lib/python3.11/dist-packages/sklearn/manifold/_t_sne.py:1164: FutureWarnir
  warnings.warn(
```