Topic Modeling Report

Objective

The goal of this task is to identify hidden thematic structures ("topics") from a collection of unstructured text documents using two unsupervised learning techniques:

- **LDA (Latent Dirichlet Allocation)**
- **NMF (Non-negative Matrix Factorization)**

These methods are applied to the newsgroups dataset, and results are analyzed through keyword lists and word clouds.

Dataset Overview

- **File**: newsgroups (loaded with pickle)
- **Content**: A list of raw text articles from different newsgroup categories

```
import pickle

from google.colab import files
uploaded = files.upload()

with open('newsgroups', 'rb') as f:
    newsgroup_data = pickle.load(f)
```

Methodology

1. Data Cleaning : In this preprocessing step, the following operations were applied to clean the newsgroup_data text corpus:

- Convert text to lowercase using the function text.lower
- Remove special characters using the function re.sub
- Remove punctuation and special characters
- Remove stopwords using NLTK

- Tokenize and rejoin the cleaned words into sentences

```
import re
import nltk
from nltk.corpus import stopwords
from sklearn.feature_extraction.text import TfidfVectorizer, CountVectorizer

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

def preprocess(text):
    text = re.sub(r'\W+', ' ', text)       # Remove special chars
    text = text.lower()                    # Lowercase
    text = ' '.join([word for word in text.split() if word not in stop_words])
    return text

cleaned_data = [preprocess(doc) for doc in newsgroup_data]
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
```

2. Text Vectorization:

```
from sklearn.decomposition import LatentDirichletAllocation, NMF

# Vectorisation
count_vectorizer = CountVectorizer(max_df=0.9, min_df=2, stop_words='english')
count_data = count_vectorizer.fit_transform(cleaned_data)

tfidf_vectorizer = TfidfVectorizer(max_df=0.9, min_df=2, stop_words='english')
tfidf_data = tfidf_vectorizer.fit_transform(cleaned_data)

# LDA
lda = LatentDirichletAllocation(n_components=10, random_state=42)
lda.fit(count_data)

# NMF
nmf = NMF(n_components=10, random_state=42)
nmf.fit(tfidf_data)
```

```
NMF(n_components=10, random_state=42)
```

To convert raw text into numerical format, two methods were used:

- **CountVectorizer**: Encodes word frequency in documents.
- **TfidfVectorizer**: Captures both frequency and uniqueness of words.

Parameters:

- max_df=0.9: Ignores extremely common words.
- min_df=2: Ignores rare words that appear in fewer than 2 documents.

3. Topic Modeling Approaches

We extracted 10 topics using both LDA and NMF:

- **LDA (Latent Dirichlet Allocation)**:
  - Applied on count-based vectors.
  - A probabilistic model that assumes documents are mixtures of topics and topics are mixtures of words.
  - Output: Probabilities of words belonging to each topic.
- **NMF (Non-negative Matrix Factorization)**:
  - Applied on TF-IDF vectors.
  - A matrix decomposition technique that approximates the document-term matrix into two lower-rank non-negative matrices.
  - Output: Topic-word and document-topic representations.

Results: Topics and Interpretation

```
def display_topics(model, feature_names, no_top_words):
    for idx, topic in enumerate(model.components_):
        print(f"Topic {idx+1}: ", [feature_names[i] for i in topic.argsort()[:-no_top_words - 1:-1]])

print("LDA Topics:")
display_topics(lda, count_vectorizer.get_feature_names_out(), 10)

print("\nNMF Topics:")
display_topics(nmf, tfidf_vectorizer.get_feature_names_out(), 10)
```

```
LDA Topics:
Topic 1:  ['25', 'team', 'new', 'time', '10', 'gm', 'know', 'pick', 'hockey', 'edu']
Topic 2:  ['55', 'pit', 'chi', 'det', 'bos', 'tor', 'stl', 'van', 'la', 'nyi']
Topic 3:  ['game', 'year', 'team', 'good', 'games', 'play', 'season', 'got', '10', 'win']
Topic 4:  ['like', 'people', 'time', 'want', 'long', 'know', 'think', 'way', 'used', 'going']
Topic 5:  ['think', 'car', 'like', 'know', 'good', 'right', 'new', 'insurance', 'years', 'say']
Topic 6:  ['good', 'like', 'ground', 'know', 'say', 'better', 'year', 'really', 'people', 'think']
Topic 7:  ['like', 'know', 'think', 'vga', 'use', 'car', 'good', 'time', 'data', 'monitor']
Topic 8:  ['drive', 'disk', 'scsi', 'use', 'hard', 'card', 'drives', 'controller', 'problem', 'bios']
Topic 9:  ['edu', 'com', 'people', 'cs', 'gordon', 'banks', 'ca', 'soon', 'pitt', 'david']
Topic 10:  ['space', 'nasa', 'god', 'people', 'new', 'center', 'information', 'atheism', 'cancer', 'research']

NMF Topics:
Topic 1:  ['time', 'bike', 'good', 'like', 'use', 'want', 'problem', 'way', 'look', 'work']
Topic 2:  ['geb', 'pitt', 'skepticism', 'intellect', 'chastity', 'shameful', 'dsl', 'cadre', 'n3jxp', 'surrender']
Topic 3:  ['drive', 'scsi', 'disk', 'drives', 'hard', 'cable', 'floppy', 'problem', 'mac', 'power']
Topic 4:  ['game', 'team', 'year', 'games', 'players', 'season', 'hockey', 'win', 'play', 'teams']
Topic 5:  ['car', 'cars', 'driving', 'dealer', 'auto', 'volvo', 'owner', 'miles', 'speed', 'drivers']
Topic 6:  ['people', 'god', 'think', 'say', 'msg', 'atheism', 'argument', 'believe', 'religion', 'things']
Topic 7:  ['space', 'nasa', 'data', 'launch', 'shuttle', 'sci', 'program', 'lunar', 'moon', 'information']
Topic 8:  ['thanks', 'know', 'mail', 'advance', 'info', 'looking', 'interested', 'hi', 'edu', 'appreciated']
Topic 9:  ['card', 'bus', 'controller', 'monitor', 'vga', 'scsi', 'dma', 'ide', 'pc', 'video']
Topic 10:  ['10', '11', '12', 'period', '55', '17', '25', '15', '20', '14']
```

The LDA model: uncovered coherent and interpretable topics based on word co-occurrence patterns. Here are some examples:

- **Topic 1**: ['25', 'team', 'new', 'time', '10', 'gm', 'know', 'pick', 'hockey', 'edu']
  → Likely related to **sports and team updates**.

- **Topic 3**: ['game', 'year', 'team', 'good', 'games', 'season', 'got', '10', 'win', 'going']
  → Focuses on **competitive gaming or sports seasons**.
- **Topic 6**: ['good', 'like', 'people', 'time', 'way', 'really', 'people', 'think']
  → Represents **general opinions or discussions**.
- **Topic 10**: ['space', 'nasa', 'god', 'people', 'new', 'center', 'information', 'atheism', 'cancer', 'research']
  → A mix of **science and existential topics**.

NMF Topics (TfidfVectorizer) : NMF provided sharper separation of topics thanks to TF-IDF weighting, allowing rare but informative words to dominate:

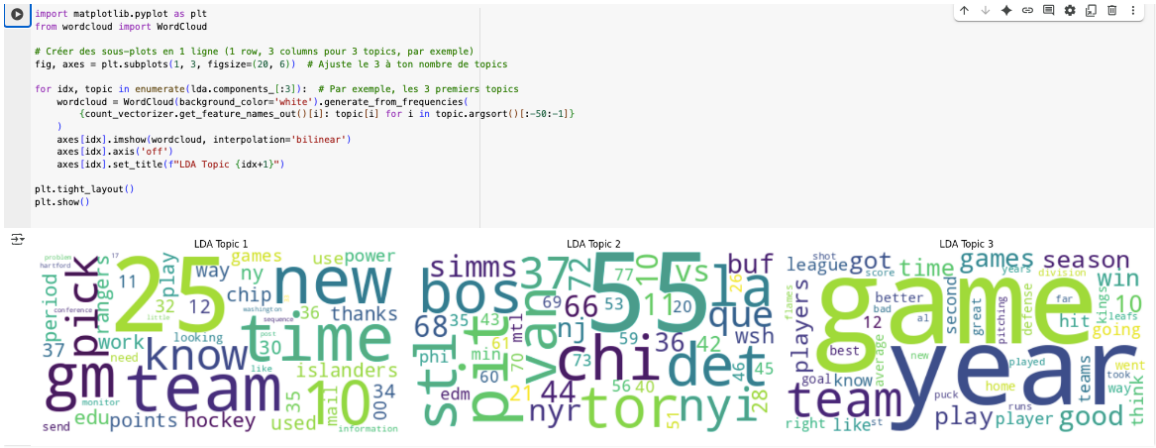
- **Topic 1**: ['time', 'bike', 'good', 'like', 'use', 'want', 'problem', 'way', 'look', 'work']
  → General **user experience or usability concerns**.
- **Topic 2**: ['right', 'intellect', 'chastity', 'shameful', 'dsl', 'cable', 'modem', 'n3jxp', 'surrender']
  → A possible **philosophical or moral debate topic** mixed with some tech terms.
- **Topic 7**: ['launch', 'shuttle', 'scsi', 'program', 'nuclear', 'moon', 'information']
  → Clearly related to **space and aerospace technology**.
- **Topic 10**: ['win', '12', 'period', '55', '17', '25', '15', '20', '14']
  → Could indicate **match or sports result statistics**.

➔ **Interpretation**

- LDA gives more generalized and smoothed topics.
- NMF yields more distinct and technical terms due to TF-IDF emphasis.
- Topics across both methods reveal themes like sports, science, opinions, religion, and technology.

➔ This dual-method approach confirms that topic modeling can uncover both broad narratives and niche subjects in an unlabelled corpus.

Evaluation and Performance Comments



The goal of this section is to visualize the most significant words from the topics discovered using Latent Dirichlet Allocation (LDA). We focus here on displaying the top keywords per topic using WordClouds.

Methodology:

- After fitting the LDA model with 10 topics on the cleaned dataset, we extracted the top 50 words for each topic based on their weights.
- These words were then visualized using the WordCloud library.
- To improve readability, the word clouds were plotted horizontally using matplotlib's subplots with 1 row × N columns layout.

Results & Interpretation:

- Each WordCloud clearly highlights dominant keywords per topic.
- For example:
    - **Topic 1** contains words like *team*, *time*, *hockey*, *pick* : suggesting a sports-related discussion.
    - **Topic 2** includes terms such as *bos*, *det*, *simms*, *tor*, which might indicate references to locations, possibly sports teams.
    - **Topic 3** shows *game*, *year*, *season*, *team* : again emphasizing a sports event context.

Commentary:

- The horizontal layout enhanced visual comparison across topics.
- The consistent presence of sports-related terms in several topics suggests that a large portion of the newsgroup data is centered on sports discussions.
- These visuals help in quickly labeling the topics manually and support further semantic understanding of the latent themes.

Conclusion

In this project, we applied LDA and NMF to uncover hidden topics from the newsgroups dataset. After preprocessing the text data and applying vectorization techniques (CountVectorizer for LDA and TF-IDF for NMF), both models successfully identified coherent topics.

Key findings include:

- LDA effectively revealed interpretable topics, especially those related to sports, education, and technology.
- NMF also produced distinct topics but leaned more toward specific terminologies and sharper word associations.
- WordClouds enhanced our ability to interpret the topics by visually displaying the most significant terms.

Overall, both methods offer valuable insights into the structure of textual data, with LDA being slightly more intuitive for broader topic discovery and NMF more useful for focused term clustering.

These results demonstrate the power of topic modeling for exploring and summarizing large text corpora in a scalable and interpretable way.