# Topic Modeling on NYT Articles Dataset

## Introduction

In today's digital landscape, vast amounts of unstructured textual data are continuously generated through platforms such as news outlets, blogs, and social media. Analyzing this data to extract meaningful insights presents both a significant challenge and a valuable opportunity. Topic modeling serves as an effective unsupervised machine learning approach to identify underlying themes and semantic structures within large text corpora. This project employs Latent Dirichlet Allocation (LDA), a widely recognized topic modeling technique, to analyze a dataset of New York Times (NYT) articles. The primary objective is to uncover the dominant topics present across the dataset and to provide a thematic summary, highlighting the key terms associated with each topic.

## Related Work

A substantial body of research has explored the application of topic modeling techniques to news datasets for purposes such as trend detection, article clustering, and the development of recommendation systems. Latent Dirichlet Allocation (LDA), introduced by Blei et al. (2003), is a foundational generative probabilistic model designed for uncovering latent structures in collections of discrete data. Since its introduction, LDA has undergone various enhancements and optimizations, including parallelized implementations such as *LdaMulticore* in the Gensim library. Tools like *pyLDAvis* (Sievert & Shirley, 2014) have gained widespread adoption for their ability to visually assess the coherence and overlap of extracted topics. Alternative topic modeling approaches include Non-negative Matrix Factorization (NMF), Correlated Topic Models (CTM), and more recent methods such as BERTopic, which leverage transformer-based embeddings. Nevertheless, LDA continues to be favored for its interpretability and computational efficiency, particularly when working with large-scale textual corpora.

## Methodology

This section documents our approach step-by-step without discussion or interpretation:

**1. Environment Setup:**

- Uninstall conflicting or partially installed packages.
- Install compatible versions of core scientific and NLP libraries, including NumPy, Pandas, SciPy, Gensim, scikit-learn, and pyLDAvis.

**2.Library Imports:**

- Import required libraries for data manipulation, visualization, natural language processing, and topic modeling.
- Download necessary NLTK datasets such as punkt, wordnet, stopwords, averaged_perceptron_tagger, and omw-1.4.

**3.Data Loading:**

- Mount Google Drive to access project files.
- Read the dataset stored in Parquet format and convert it to CSV for easier handling.

**4.Dataset Overview:**

- Print the shape, column names, data types, and memory usage of the dataset.
- Display the count of articles per year.
- Identify missing values in the dataset.
- Generate and display a line chart showing the number of articles per year.

**5.Chunked Data Loading and Preview:**

- Read the CSV file in manageable chunks using a defined chunk size.
- Preview the first chunk to verify the structure and columns of the dataset.

**6.Text Preprocessing:**

- Define a preprocessing function that converts text to lowercase, removes digits and punctuation, tokenizes, removes stopwords, and applies stemming.
- Read the entire CSV file in large chunks.
- Drop rows with missing values in the “excerpt” column.
- Apply the preprocessing function to each chunk and retain only the “excerpt” and “year” columns.
- Store and combine all processed chunks into a single DataFrame.
- Save the final preprocessed DataFrame as a new CSV file.

**7.LDA Model Training:**

- Select a sample of 100,000 records from the processed dataset.
- Tokenize the processed text into word lists.
- Create a Gensim Dictionary and filter extreme terms based on conditions as considering only top 20,000 most frequent words.
- Convert the text into a Bag-of-Words corpus i.e., a tuple of unique word id with its word count.
- Train an LDA topic model using LdaMulticore with defined parameters including number of topics, passes, iterations, and chunk size.
- Compute the coherence score using the c_v metric to evaluate topic quality.

**8.Topic Assignment:**

- Extract topic distributions for each document from the LDA model.
- Assign the dominant topic (with the highest probability) to each document in the sample dataset as an another column.

**9.Topic Trends Analysis:**

- Group documents by year and dominant topic, and count occurrences.
- Normalize topic frequencies by year to compute proportions.
- Generate a line plot showing how topic proportions change over time.

**10.Top Words per Topic:**

- Define and use a function to display the top keywords associated with each topic generated by the LDA model.

**11.Word Cloud Generation:**

- For each topic, generate a word cloud to visually highlight the most representative words.
- Display individual word clouds using matplotlib for better interpretability.

**12.Bar Chart of Top Words per Topic:**

- Define a function to plot the top keywords of each topic as horizontal bar charts.
- Generate and display these bar plots for all topics in the model.

**13.Interactive Topic Visualization:**

- Prepare and display an interactive visualization of the LDA model using pyLDAvis, which helps assess topic distribution and word relevance across topics.

## Results and Discussion

**Methodology Justification:**

To begin, the text data was carefully cleaned through a series of preprocessing steps—converting everything to lowercase, removing numbers and punctuation, filtering out common stopwords, and applying stemming to reduce words to their base form. These steps helped simplify the text and reduce noise, making it easier for the model to find patterns. Because the dataset was quite large, the data was processed in chunks to avoid memory issues and keep things efficient.

For topic modeling, we used Latent Dirichlet Allocation (LDA) because it offers a good balance between being interpretable and scalable. We chose the LdaMulticore implementation from Gensim, which takes advantage of multiple processor cores to speed up training—especially helpful with over 100,000 documents. The model's performance was fine-tuned by setting the number of topics to 15, running 20 passes, and increasing iterations to 1000 for better convergence and clearer topic definitions.

To evaluate how well the model performed, we calculated the c_v coherence score—a metric that checks how semantically related the top words in each topic are. This score gives a solid indication of whether the topics make intuitive sense.

**Results:**

- **Topic Identification**: The model was able to assign a clear dominant topic to each article. Each topic was defined by its most relevant keywords, which we displayed using both text lists and visual word clouds. The model captured a range of themes like politics, economics, health, and technology, with each topic clearly distinguished by its vocabulary.
- **Trends Over Time**: We visualized how these topics shifted over the years. This showed which themes became more or less prominent over time, offering insights into changing public interest and media focus.
- **Keyword Insights**: For each topic, we highlighted the most important words using bar charts and word clouds. These visual tools helped make the results easier to understand and provided a quick grasp of what each topic was about.
- **Interactive Exploration**: We also created an interactive visualization using pyLDAvis, which made it easier to explore how topics are related, how specific words contribute to each topic, and how well-separated the topics are from one another.

**Comparison with Other Approaches:**

While there are more advanced topic modeling techniques—like BERTopic, which uses transformer-based embeddings, or methods like Non-negative Matrix Factorization (NMF) and Correlated Topic Models (CTM)—we opted for LDA in this project for several reasons:

- It's easy to understand and interpret.
- It handles large datasets efficiently.
- It's well-supported by widely used open-source libraries.

BERTopic can capture deeper contextual meanings, but it requires significantly more computing power and can be harder to interpret. NMF doesn't offer a probabilistic understanding of the data, and CTM, though powerful, is more complex to implement and manage.

**Limitations:**

- **Loss of Word Order**: LDA works on the assumption that word order doesn't matter, which means it misses out on the meaning that comes from how words are arranged in a sentence.
- **Fixed Topic Number**: We had to decide the number of topics beforehand, which may not always match the natural structure of the dataset.
- **Preprocessing Trade-offs**: Removing very common or very rare words can simplify the data, but there's always a risk of losing meaningful information in the process.
- **Computational Demand**: Even with parallel processing, training the model and evaluating its quality can take a considerable amount of time on large datasets.

## Conclusion and Future Work

**Conclusion:**

This project successfully used Latent Dirichlet Allocation (LDA) to discover meaningful themes hidden within a large collection of New York Times articles. By combining thorough text cleaning and preprocessing with a well-tuned LDA model, we were able to identify and visualize dominant topics across thousands of documents. Each topic was described by its most important keywords and brought to life through word clouds and bar charts. The addition of an interactive visualization using pyLDAvis made it even easier to explore and interpret the results. Overall, this analysis highlighted how powerful unsupervised topic modeling can be when it comes to summarizing large volumes of unstructured text. It also offered a window into how news coverage and public conversations have shifted over time. The coherence score provided confidence that the topics we discovered were both logical and meaningful.

**Future Work:**

There are several ways this project could be expanded or improved:

- **Track Topic Evolution**: In the future, models like Dynamic Topic Modeling (DTM) could be used to show how specific topics change and develop over time.
- **Use Contextual Embeddings**: Incorporating advanced models like BERTopic, which are built on transformer embeddings, could help the model understand deeper context and more subtle patterns in the text.
- **Automate Topic Labels**: Right now, understanding each topic still requires manually looking at the top keywords. Future work could involve automating this process using external data or machine learning.
- **Add Metadata and Entities**: Including more context—such as the article's author, section, or named entities like people and places—could make the topic analysis even richer and more precise.
- **Tune the Model Further**: Exploring techniques like grid search or Bayesian optimization could help fine-tune LDA's parameters to get even better topics.
- **Improve Visualizations**: Finally, making the visual outputs more interactive and user-friendly—such as allowing users to filter by year or topic type—would make the results more accessible to a broader audience, especially those without technical backgrounds.