Portrayal of Terrorist Organizations in Traditional Media

A Topic Modeling Analysis of Wall Street Journal and New York Times Articles

• Manish Rawat [1], • Temidara Agbesanwa[2], and • Abdul Aziz Mohammed[3]

[1]Department of Engineering and Data Science, University of Houston

April 28, 2023

Abstract

Topic modeling is a technique used in natural language processing (NLP) that automatically identifies topics or themes present in an extensive collection of documents. This report uses topic modeling to analyze the portrayal of terrorist organizations in traditional media outlets.

We explore various preprocessing methods to transform the data into a more practical format before feeding it into our model. We have used Exploratory Data Analysis (EDA) to visualize our data and extract vital features and trends.

1 Introduction

The basic idea behind topic modeling is to analyze the words that appear in a document and group them based on their frequency and context. By looking at word usage patterns, the algorithm can identify clusters of words that often appear together and assign them a label or topic.

For example, we were analyzing a collection of news articles. In that case, the topic modeling algorithm might identify a cluster of words like "president," "election," "vote," and "campaign" and label it as a "political" topic. Another cluster of words like "restaurant," "menu," "food," and "chef" might be labeled as a "food and dining" topic.

Topic modeling can be helpful in many applications, such as information retrieval, text mining, and recommendation systems. It can help us understand the main themes and ideas in a large corpus of text data and make searching and organizing that data easier.

2 Methodology

Preprocessing and EDA are essential steps in text mining that help improve the data qual-

ity, reduce noise, and extract valuable insights. They are crucial to making informed decisions about algorithms, parameters, and modeling techniques and producing reliable results.

2.1 Data Collection

Within this study, we used a large Global News database called Factiva as the source for the data collected, which contained articles from the New York Times and the Wall Street Journal in 2017.

2.2 Data Pre-processing

We begin by importing the dataset and splitting it into individual articles to build a corpus. After creating the corpus, we must separate the metadata from the actual articles. To accomplish this, we can use regular expressions to match and remove any headers or tags that appear in the text.

We use the nltk library to tokenize, lowercase, and filter out stop words and the gensim library to remove words that appear too frequently or infrequently within the text. Once we have completed cleaning up the corpus and extracting features, we can further explore the data by creating summaries on the features.

2.3 Data Visualisation

We have created a function to calculate coherence score to train LDA models with different topics and calculate coherence scores for each model. We use matplotlib and pyLDAvis to visualize the most coherence topic models from the analysis by creating a word cloud and histogram of the most frequent words in the corpus.

Word clouds are a popular visualization technique used in text mining to display the most frequent words in a corpus. Word clouds can quickly and easily overview a corpus's main topics and themes and help identify essential keywords and trends. They are often used as a starting point for further analysis, such as topic modeling or sentiment analysis. Word clouds can also be customized to include or exclude specific words, adjust the size and color of the words, and display the words in different layouts, making them a versatile and flexible tool for visualizing text data.

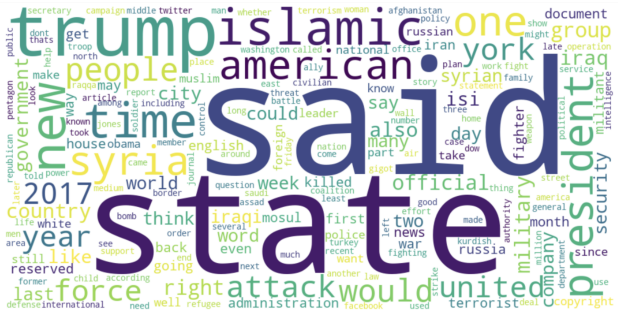

Figure 2: Top 20 word Frequency

Histograms are a type of visualization used in text mining to display the frequency distribution of words or phrases in a corpus. In a histogram, the x-axis represents the range of values or frequencies of the words or phrases, while the y-axis

represents the number of occurrences of those values or frequencies. Histograms can help identify the most common and rare words or phrases in a corpus and for detecting patterns and trends in the data (Refer: Figure 2). They can also be used to identify potential outliers or anomalies in the data. Histograms can be customized to adjust the size and number of bins, as well as the colors and labels of the axes, making them a flexible and versatile tool for visualizing text data.

However, it is essential to note that word clouds and histograms only display the frequency of words and do not provide any information about the context or relationships between words. This is why combining them or other visualization techniques and analysis methods is important for a more comprehensive understanding of the data.

3 Experimental Results

3.1 Preprocessing and EDA

The initial step in our analysis involves importing the dataset and converting it into a corpus of individual articles. We perform data preprocessing and exploratory data analysis (EDA) on the corpus to draw observations and insights from the data as follows:

3.1.1 Observations

1. The corpus was loaded as a list of articles and combined into a single string object.

2. Meta-data and tags were removed from the corpus using regular expressions to ensure that the analysis focuses only on the main textual content.

3. The text was tokenized using the word tokenize function from the Natural Language Toolkit (nltk) library, which breaks the text into individual words.

4. Stop words, such as common prepositions, articles, and conjunctions, were removed from the corpus using the stopwords.words('english') function from the nltk library to focus on more meaningful words.

5. Words that appeared too frequently or infrequently in the corpus were filtered out using a frequency filter with a lower count threshold of 10 and an upper count threshold of 1000. This step helped in retaining only the most relevant words for further analysis.

Table 1: Word frequencies in corpus

Word	**Frequency**
said	11688
state	10422
trump	7717
islamic	6287
new	6228
time	5888
president	4372
one	4290
syria	4081
american	4080

6. The top 10 words in the corpus and their frequencies are presented in Table 1. These words indicate some of the most common themes and topics present in the articles. Similarly, we have used a Histogram plot to visualize the top 20 words (Refer: Figure 2).

7. A word cloud was generated using the filtered tokens to visualize the most common words in the corpus (Refer: Figure 1). The word cloud visually represents the words' relative frequencies and highlights the most prominent themes and topics within the articles.

3.2 Topic modeling

Topic Modeling is a statistical technique used to identify topics or themes within a large corpus of text data. Latent Dirichlet Allocation (LDA) is a commonly used algorithm for Topic Modeling, a probabilistic model used for topic modeling in natural language processing.

Assumptions:

1. Text data is a collection of documents.
2. Each document is a mixture of one or more topics.
3. Each topic is a probability distribution over a fixed vocabulary of terms.
4. Each word in a document is generated by one of the topics according to its probability distribution over the topics.

3.2.1 LDA Algorithm

1. Initialize the number of topics and the number of words in each topic.
2. Randomly assign each word in each document to a topic.
3. For each document and each topic, calculate the proportion of words assigned to that topic.
4. For each topic and each word, calculate the proportion of assignments to that topic.
5. Iterate steps 3 and 4 until convergence.

3.2.2 Observations

1. The documents were preprocessed by tokenizing them into words and removing stop words. The preprocessed documents are stored in the "texts" variable. We have used necessary libraries like gensim for topic modeling and matplotlib and pyLDAvis for visualization.
2. Topic modeling is implemented using Latent Dirichlet Allocation (LDA) for 10 topics and calculating the coherence score to evaluate the quality of the generated topics. The coherence score measures the quality of the topics generated by a model. We can compute the coherence score for each model using the CoherenceModel class.
3. The function is created, which takes in a corpus, a dictionary of words, and a list of texts as input. The function uses the Gensim library's LdaModel and CoherenceModel classes to generate LDA models and calculate coherence scores. It loops through a range of numbers of topics (specified by the input arguments) and generates an LDA model for each number of topics.
4. It then generates multiple LDA models with different numbers of topics and computes the coherence score for each model. The function returns a list of the generated LDA models and their corresponding coherence scores.
5. Computing the Coherence score:

(a) No's of topics: 2,
Coherence Score: 0.3296

(b) No's of topics: 3,
Coherence Score: 0.3258

(c) No's of topics: 4,
Coherence Score: 0.3328

(d) No's of topics: 5,
Coherence Score: 0.3217

(e) No's of topics: 6,
Coherence Score: 0.3170

(f) No's of topics: 7,
Coherence Score: 0.3326

(g) No's of topics: 8,
Coherence Score: 0.3318

(h) No's of topics: 9,
Coherence Score: 0.3327

(i) No's of topics: 10,
Coherence Score: 0.3224

6. The coherence scores showed that the optimal number of Topic 4 with a Coherence Score: 0.3328 (Refer: Figure 3).

7. The function creates word clouds() takes a list of topic models as input and generates a word cloud for each model. For each model, it creates a dictionary of word frequencies for the words in the topics and then generates a word cloud from this dictionary. The word cloud is plotted using the matplotlib library and saved as an image file. Each word cloud is given a title indicating which topic model it corresponds to. The function is designed to work with models generated using the Gensim library for topic modeling. (Refer: Figure 4 to 12).

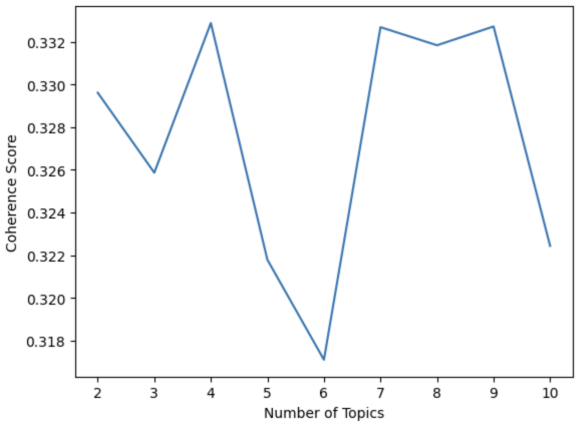

Figure 3: Optimal Model from Coherence Score

8. Topic modeling identified two dominant topics in the corpus, with varying degrees of frequency of appearance. The topics were:

   Topic 1: ['said', 'trump', 'mr', 'state', 'new', 'islamic', 'president', 'isis', 'american', 'york']

   Topic 2: ['mr', 'said', 'new', 'islamic', 'trump', 'state', 'american', 'york', 'one', 'times']

   Topic 3: ['said', 'mr', 'trump', 'state', 'islamic', 'new', 'president', 'times', 'syria', 'would']

   Topic 4: ['said', 'mr', 'state', 'trump', 'new', 'islamic', 'one', 'syria', 'people', 'would']

   The topic distribution and frequency of appearance suggest that the media focused heavily on topics related to Trump, Islamic State, ISIS, and Syria. However, the topics related to cybersecurity, the economy, and crime received relatively less attention.

9. A summary of each topic has been generated in each model to a separate output file for

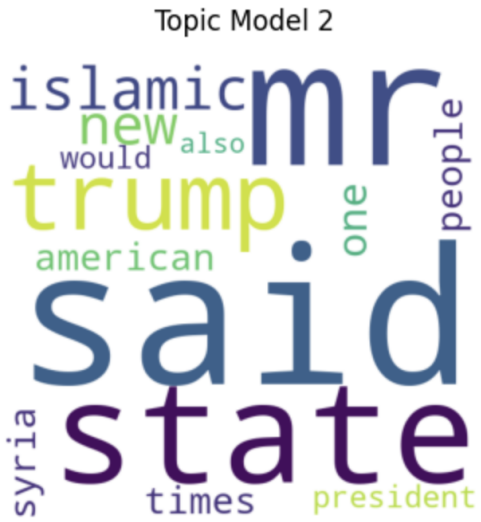

Figure 4: Word Cloud: Topic Model 2

Figure 5: Word Cloud: Topic Model 3

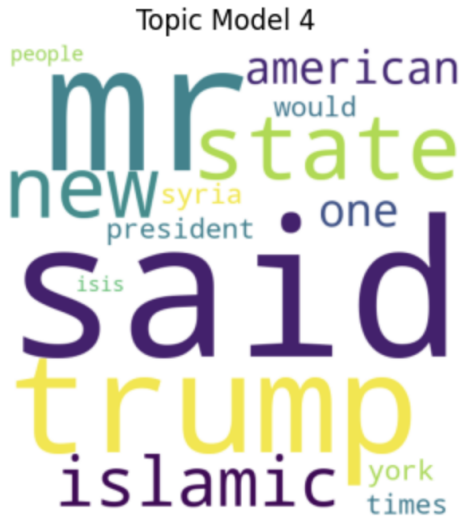

Figure 6: Word Cloud: Topic Model 4

Figure 7: Word Cloud: Topic Model 5

Figure 8: Word Cloud: Topic Model 6

Figure 10: Word Cloud: Topic Model 8

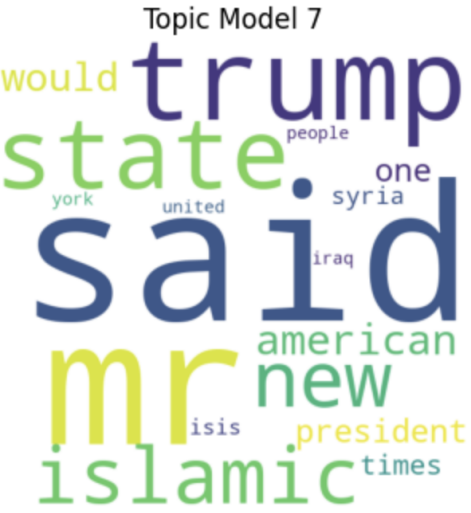

Figure 9: Word Cloud: Topic Model 7

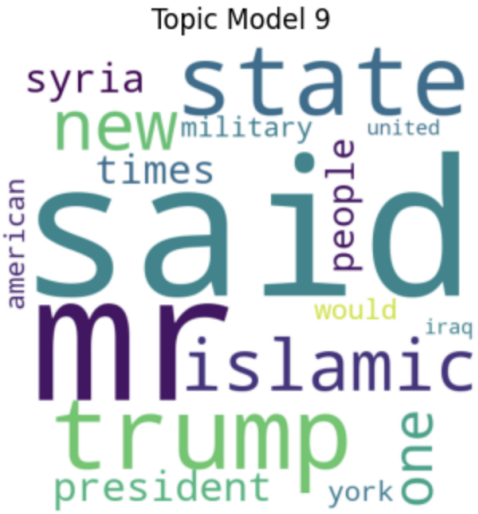

Figure 11: Word Cloud: Topic Model 9

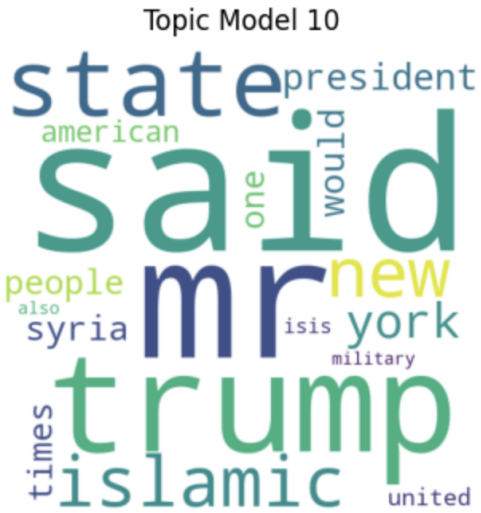

Figure 12: Word Cloud: Topic Model 10

further analysis (Refer: Figure 13) with the file name 'summaryXtopics.txt,' where X is the number of topics.

10. Finally, the best model is being used to visualize further with an interactive visualization library pyLDAvis, which shows the topics generated by the LDA model (Refer: Figure 14), along with the most representative words for each topic and the relationships between topics. pyLDAvis can help explore the topics' structure and coherence and identify areas of overlap or ambiguity.

3.3 GitHub Repository

GitHub repository can be accessed by Link.

Figure 13: Output Summaries of Topic Model 10

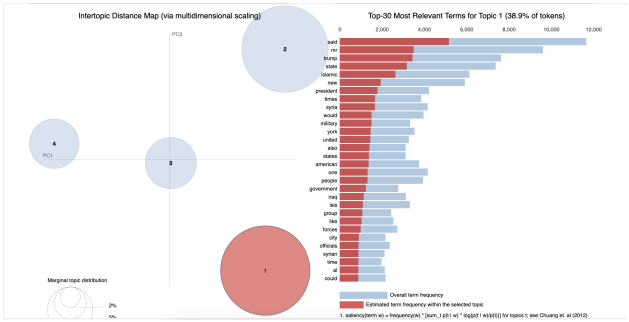

Figure 14: Topic Modeling Visualization

4 Conclusion

The data used in this study is collected from Factiva, a large Global News database, in 2017, and contains articles from Wall Street Journal and the New York Times. The articles are preprocessed by tokenizing and removing stop words. The study uses topic modeling with the LDA algorithm to extract topics from the corpus. The coherence score is used to determine the optimal number of topics. Finally, the topics are visualized using word clouds, and the best model is selected based on the coherence score.

The coherence score for different numbers of topics is shown in the coherence score plot. The best coherence score was obtained with 4 topics,

and it was 0.3328. However, the coherence scores for the number of topics between 2 and 10 are very close, and there is no clear optimal number of topics.

The word clouds for each topic are created using the word frequency dictionary for the topic. In the word clouds, we can see the most frequent words in each topic, which gives an idea about the main theme of the topic.

The best model is selected based on the coherence score, and the number of topics in the best model is four.

The topics and their corresponding theme were:

Topic 1 is related to politics, with words like Trump, President, and American.

Topic 2 is related to terrorism, with words like Terrorism, attacks, and group.

Topic 3 is related to the military and conflicts, with words like syria, militants, and forces.

Finally, Topic 4 is also related to politics and military, with words like Trump, President, and military.

Thus, topic modeling is valuable in analyzing how terrorist organizations are portrayed in traditional media outlets. The study found that the most common words in the corpus suggest a focus on American politics and foreign policy, with articles related to the Islamic State and its activities. The study also identified that the corpus has ten dominant topics related to the Islamic State, terrorism, and foreign policy. Future studies could utilize similar methods to understand better how terrorist organizations are portrayed in traditional media outlets.

5 Author Contributions

Manish Rawat: Conceptualization, Methodology, Software, Validation, Writing- Original draft, Reviewing and Editing, Supervision; **Temidara Agbesanwa**: Conceptualization, Formal Analysis, Methodology, Software, Writing- Original draft, Reviewing and Editing, Supervision; **Abdul Aziz Mohammed**: Formal Analysis, Visualization, Reviewing and Editing, Supervision.

6 References

1. *Real-time topic classification of Twitter messages.* Journal of the American Society for Information Science and Technology, 63(9), 1632-1649. Newman, D., Karimi, S., Cavedon, L., Baldwin, T. (2012).

2. *Exploring the Space of Topic Coherence Measures* by Michael Röder, Andreas Both, and Alexander Hinneburg. Proceedings of the Eighth ACM International Conference on Web Search and Data Mining, 2015.

3. *Probabilistic Topic Models* by Mark Steyvers and Tom Griffiths. Journal of Machine Learning Research, 2007.

4. *Dynamic Topic Models* by David Blei and John Lafferty. Proceedings of the International Conference on Machine Learning, 2006.

5. *Hierarchical Topic Models and the Nested Chinese Restaurant Process* by David M. Blei, Thomas L. Griffiths, and Michael I. Jordan. Advances in Neural Information Processing Systems, 2004.