

Assignment 2

34721242- Hafiz Samran Elahi



June 2, 2024

Murdoch University

ICT606- Machine Learning

## **Introduction:**

## The Twitter dataset is a handpicked selection of tweets extracted from the incomprehensible vastness of the Twittersphere. This detailed collection of data may be useful, particularly to researchers, data analysts, as well as enthusiasts who are studying the patterns of social media communication. The dataset covers a vast array of topics, industries, and areas and gives a clear insight into trends and sentiments as well as capturing the dynamics of the Twitter verse. Evaluate engagement data, map the changes in hashtags, identify key individuals, and get a better understanding of the progressive dynamics of the online world. Twitter dataset is a treasure trove of valuable data that can be used as the driving force for new research, as well as expand our knowledge about digital society through the prism of Twitter.

### **Data Source:**

The data used in this work was obtained from a source named Kaggle (Twitter-Dataset, n.d.). The Twitter data consists of 10,000 tweets scrapped from Twitter in order to perform analysis.

### **Tools and Libraries:**

1. **Python:** The programming language used for implementing the topic modeling.

2. **NLTK** (Natural Language Toolkit): Even more useful for tasks like tokenizing, removing stopwords, and lemmatization from the raw text data (Natural Language Toolkit, n.d.).

3. **Pandas:** It is primarily used when data manipulation and analysis are required, especially, while working with text data (Pandas, n.d.).

4. **Word Cloud:** Used in creating word clouds to compare the number of occurrences of the elements in the text data.

5. **Matplotlib:** Used for the visualization (Matplotlib, n.d.).

6. **Scikit-learn:** Particularly, the TfidfVectorizer class is applied to perform the TF-IDF vectorization process, while the LatentDirichletAllocation class is used to implement the LDA method (Scikit-learn, n.d.).

### Data preprocessing:

To analyze the text data, it is necessary to clean the data and make it ready for further analysis.

The following steps were executed:

**Lowercasing:** It changes all the text to the lower case as to ensure consistent representation after the-text analysis.

**URL Removal:** To clean the text, with the help of regular expressions, all the URLs like fi:/were searched by, and if found they were removed. This was done through the use of patterns, focusing on those beginning with “HTTP,” “www,” or “https,” and applying an empty pattern to them.

**Mentions and Hashtags Removal:** The textual analysis approach also excluded mentions and hashtags as strings that start with ‘@’ and ‘#’ symbols respectively were removed using regular expressions.

**Special Character Removal:** Apart from the alphanumeric characters, spaces, new lines, and tab characters the original text had other special characters which were eliminated from the full text. As for the symbols –, #, \*, and @ – these were successfully eliminated with the help of the regular expressions that followed their search and the subsequent substitution of the expressions with empty strings.

After the above steps of preprocessing the desired output text was saved in a new column named ‘Clean\_Text’ for further analysis.

**Tokenization:** To facilitate the analysis of textual data, we employed the Natural Language Toolkit (NLTK), specifically utilizing the `word\_tokenize` function. This function is designed to split strings of text into individual words, known as tokens. By applying `word\_tokenize` to each text entry, it converts the text into lists of words. These lists are then stored in a new column named 'Tokens', enabling detailed textual analysis at the word level.

**Stop Words Removal:** A lambda function is applied to the 'Tokens' column to iterate over each list of tokens and filter out any words present in the set of stop words. The resulting lists, now containing only meaningful words, were then stored back in the 'Tokens' column.

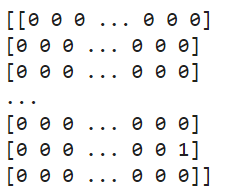
**Lemmatization:** Itinvolves reducing words to their base or root form and is useful in text analysis for normalizing the text and ensuring that different forms of a word are treated as a single item. A lambda function to the 'Tokens' column to achieve this. This function iterates over each list of tokens and applies the lemmatizer to each word. The resulting lemmatized words, now in their base forms, are stored back in the 'Tokens' column.

## Topic Modeling Building:

Latent Dirichlet Allocation (LDA) is used for topic modeling. This is a quick rundown of the procedures involved:

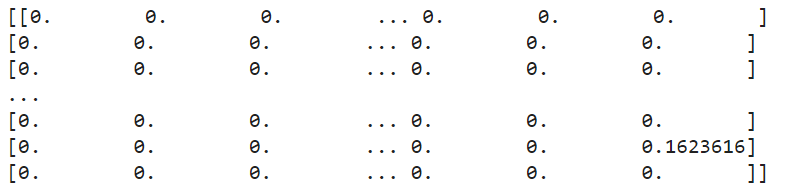
**BoW Matrix:**

The Bag-of-Words (BoW) matrix reflects the count of how often the specific word is presented in the dataset of tweets. First, we organize the extracted features in a matrix in which each row holds one tweet, and each column corresponds to a specific word in the corpus. The matrix is therefore largely empty because many of the cells contain zeros, reflecting the fact that it only provides information on whether or not words exist in the text. These tweets also form this matrix of data from which subsequent analysis can build.



**TF-IDF Matrix:**

The TF-IDF matrix measures the weightage of the words in the corpus by considering the frequency of appearance of a word within the context of a particular tweet and the rarity of such a word in all tweets. This matrix also aids in pinpointing those relevant words that are specific to each tweet as they are assigned with higher values. Similar to the case of the BoW matrix, it is sparse and retains the features of the data while focusing on important words related to tweets.



**BERT:**

BERTopic represents an innovative topic modeling algorithm that leverages BERT (Bidirectional Encoder Representations from Transformers) embeddings for document representation and clustering (BERTopic, n.d.). In contrast to conventional topic modeling methods reliant on bag-of-words representations, BERTopic accounts for the semantic context of words by embedding them into dense vector spaces. It commences by embedding the input text data through pre-trained BERT models, which capture the intricate contextual information of words. Subsequently, it employs dimensionality reduction techniques such as UMAP (Uniform Manifold Approximation and Projection) (McInnes & Healy, 2018). Visualizing the documents' high-dimensional embeddings and clusters using density-based clustering algorithms like HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise) (Malzer & Baum, 2019). This method facilitates the extraction of meaningful and interpretable topics from the data by considering the semantic relationships between words and documents. Consequently, it enables a more profound comprehension of the underlying themes prevalent within the text corpus.

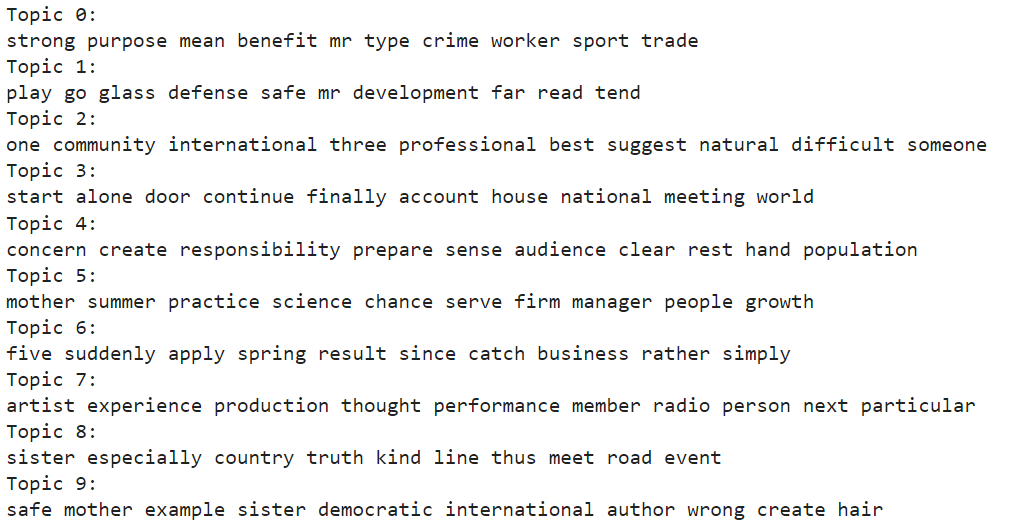
Results:

The LDA model identifies underlying topics in the text data based on word distributions and assigns words to these topics (sklearn.lda.LDA, n.d.).

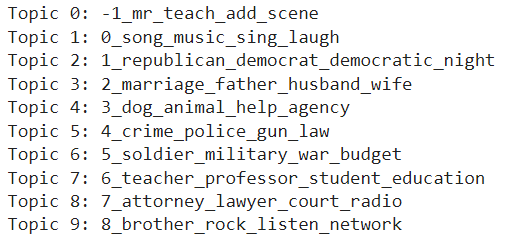
**Perplexity Evaluation:**

The value of perplexity of the LDA model on the TF IDF matrix is calculated as 2188. 93. Perplexity is a word used to denote the level of accuracy of a given model in predicting a sample and the lower, the better it is.

**Top 10 Topics Generated by the LDA Model:**



**Top 10 Topics Predicted by the BERTopic Model:**



### Visualization:

In these functions, the pre-specified topics are given, and the multiple-word keyword clusters associated with each topic are displayed using the show\_topics function which shows the top words.

They do this using the generate\_wordclouds function and present the word clouds of all the topics studied as shown below.

### Word cloud

Following are the word clouds for each predicted topic.

Topic 1:



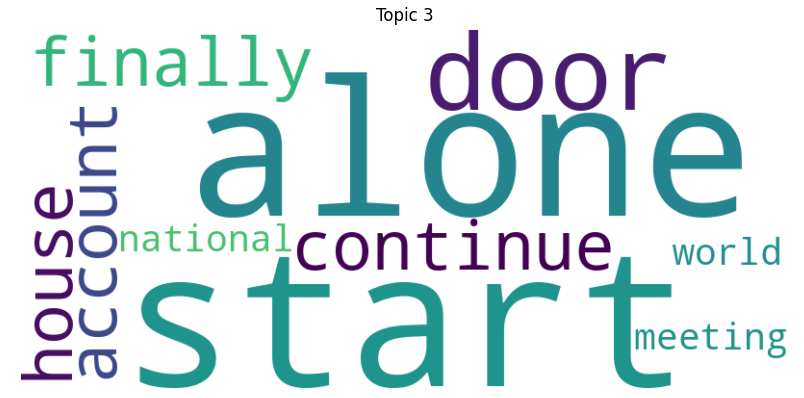
Topic 2:



Topic 3:



Topic 4:



Topic 5:

:

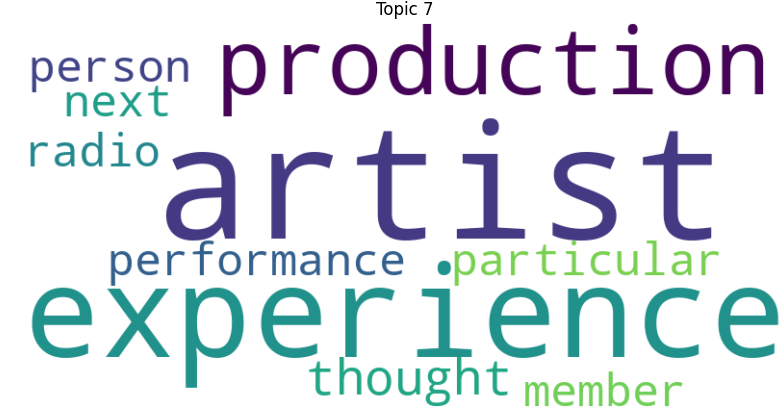
Topic 6:



Topic 7:

:

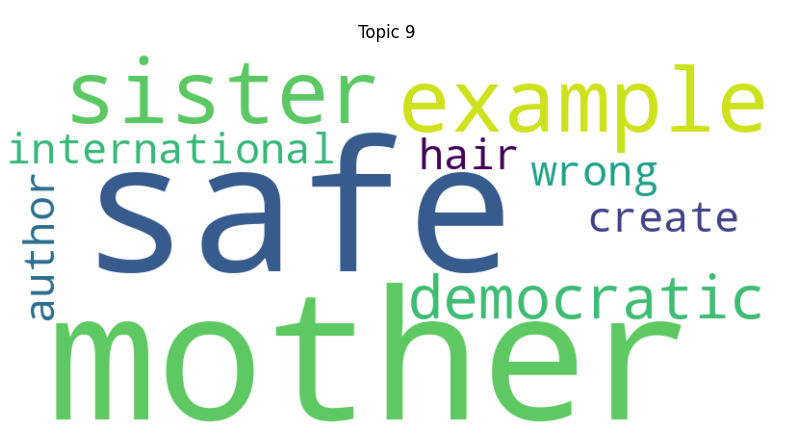
Topic 8:



Topic 9:



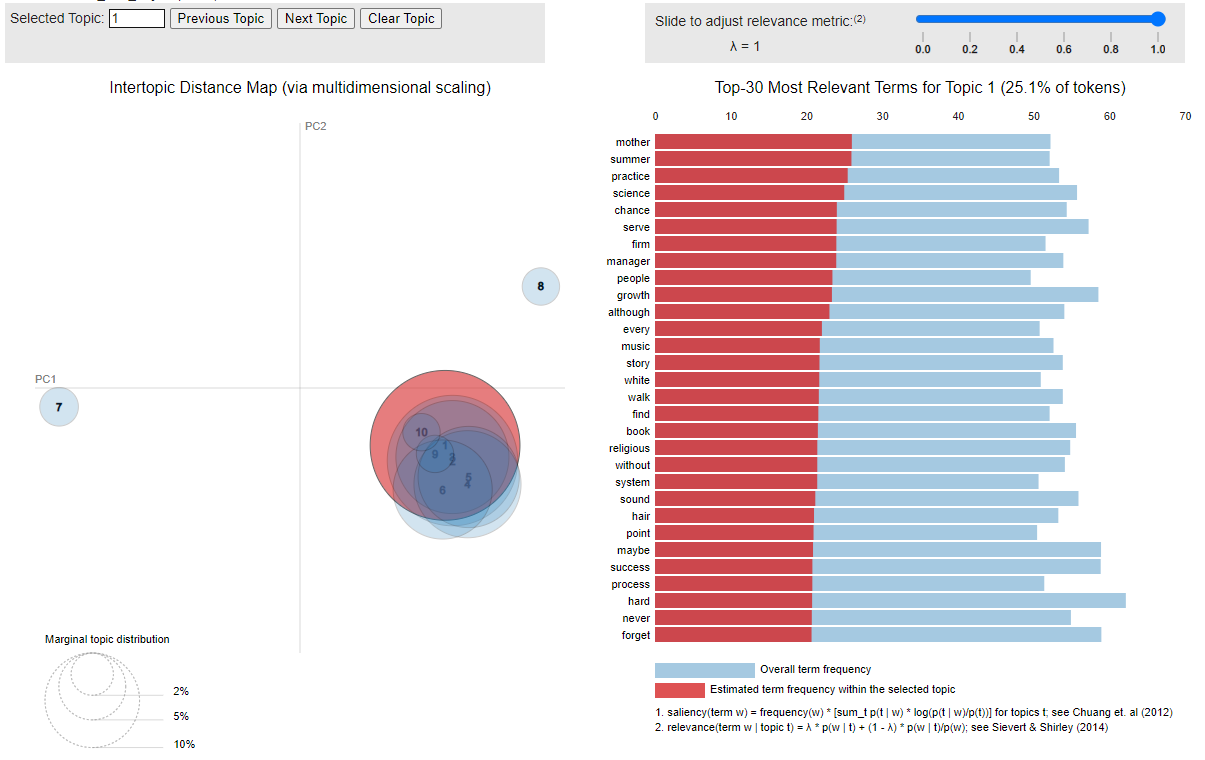
Topic 10:



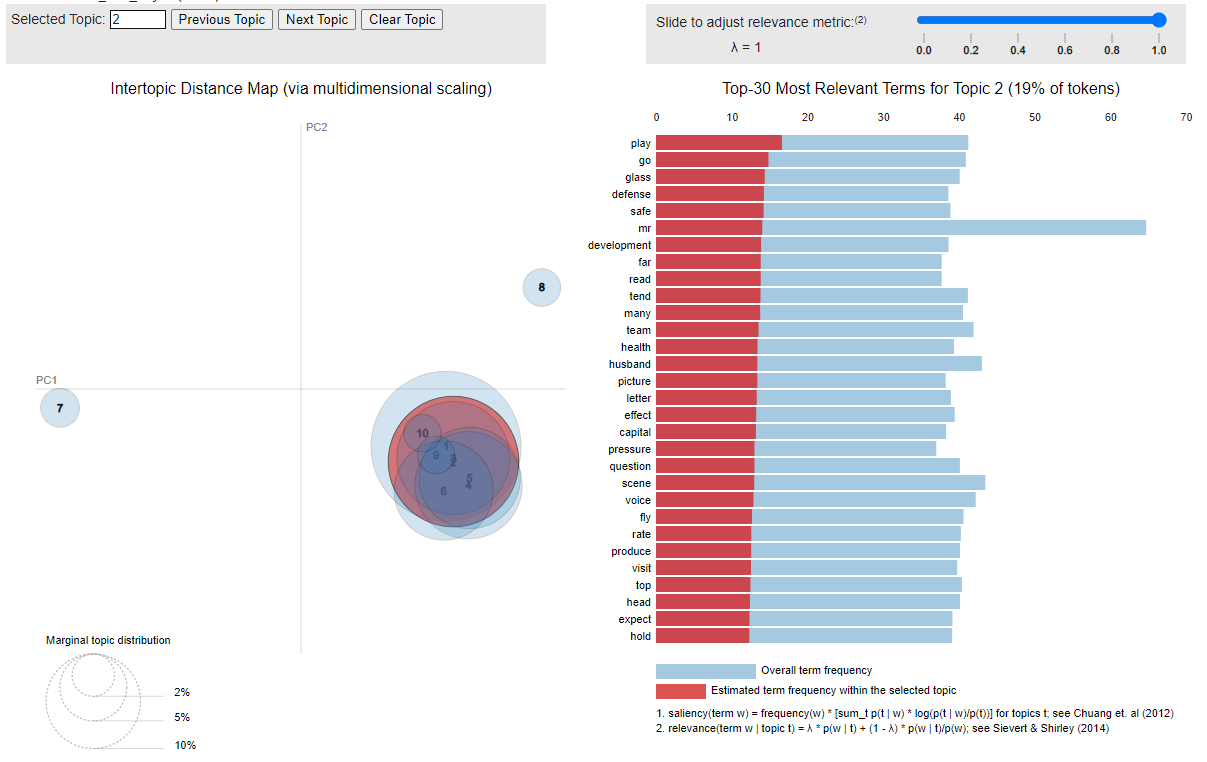
### LDA Visualization

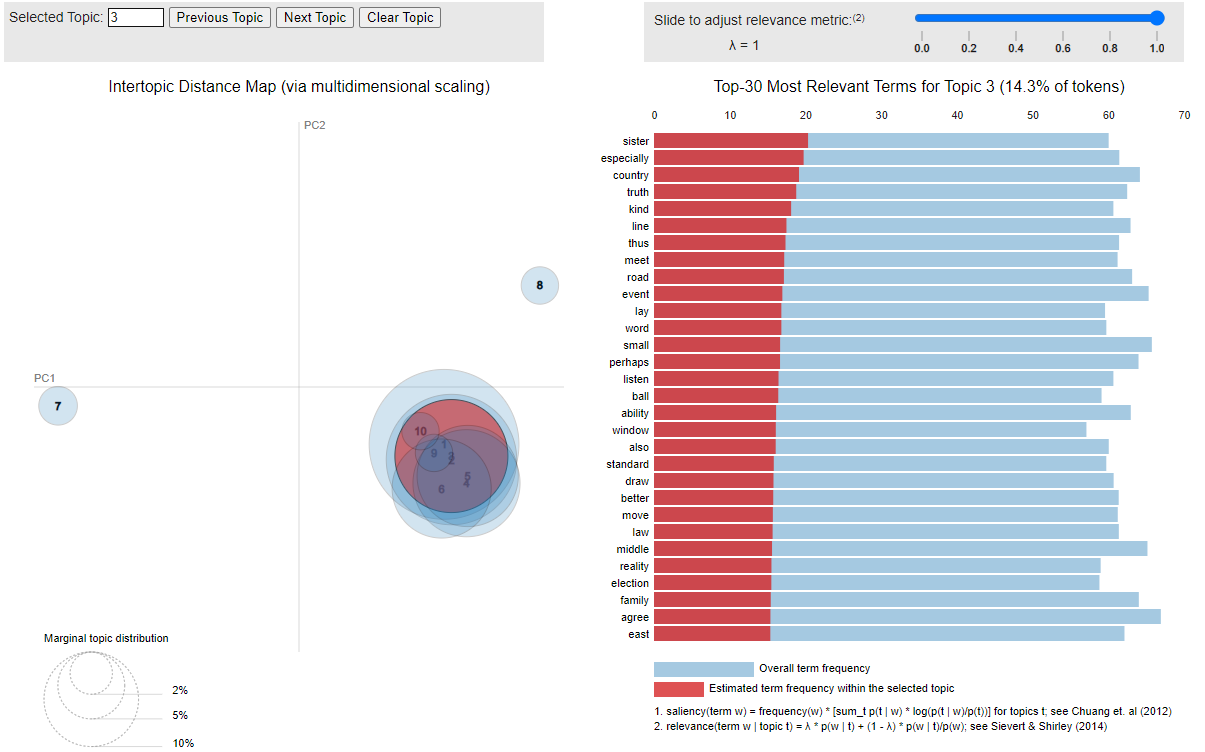
In our quest to understand the underlying topics within our textual data, we turned to the pyLDAvis library (pyLDAvis, n.d.), a robust tool renowned for visualizing topic models. Through its capabilities, we generated a visualization that effectively captures the intricate relationships between topics and terms. This interactive display offers a comprehensive overview of the discovered topics. It enhances exploration and interpretation by enabling users to delve deeper into our textual dataset's underlying structures and connections.

Topic 1:

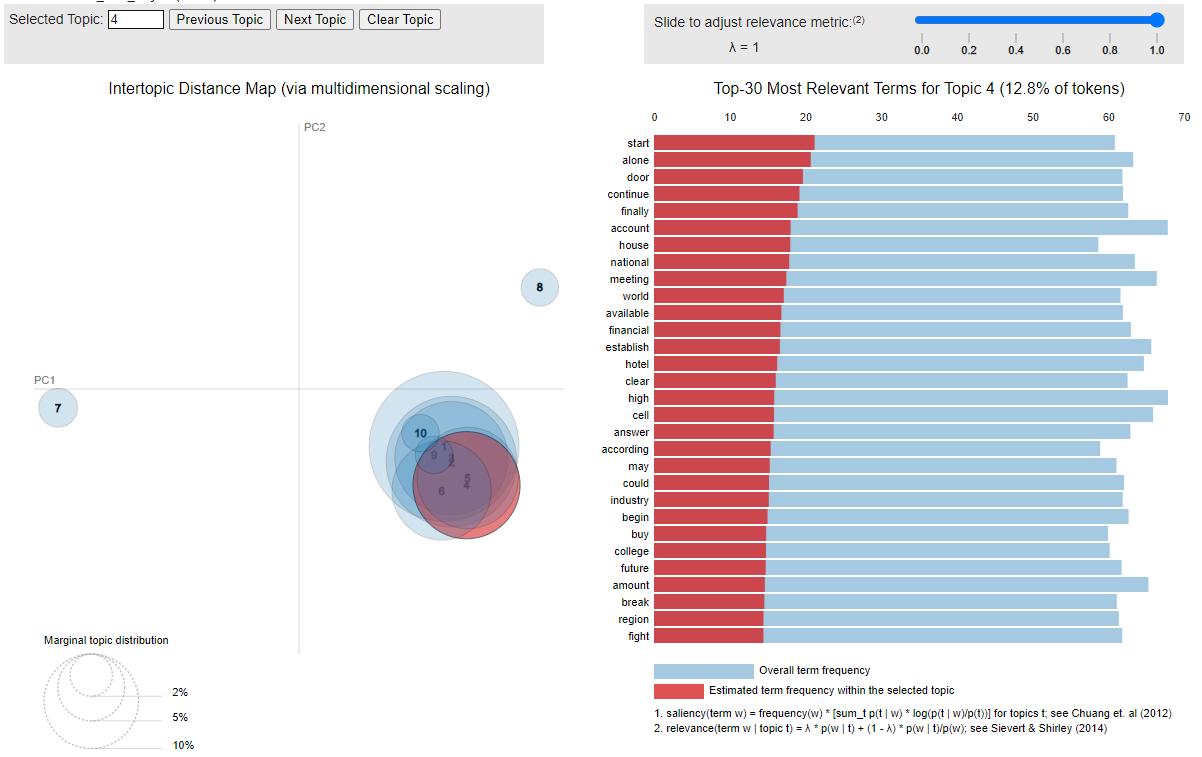


Topic 2:

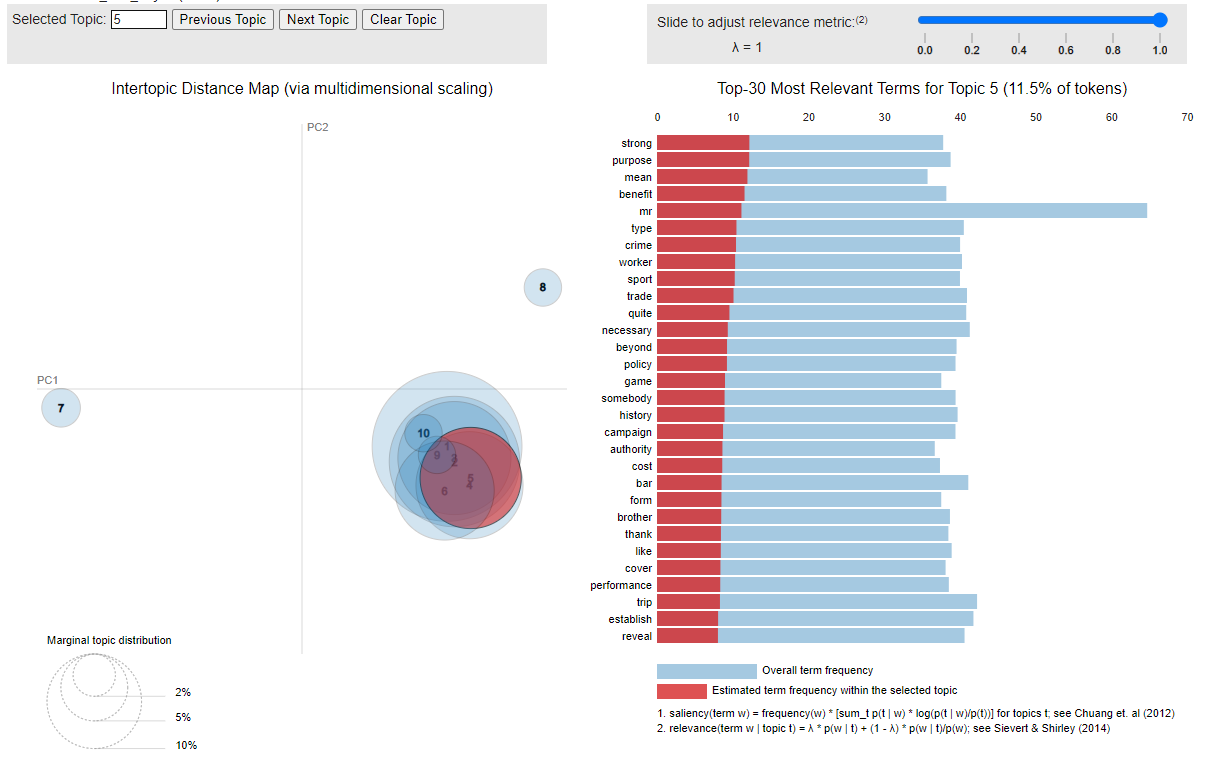


Topic 3:  


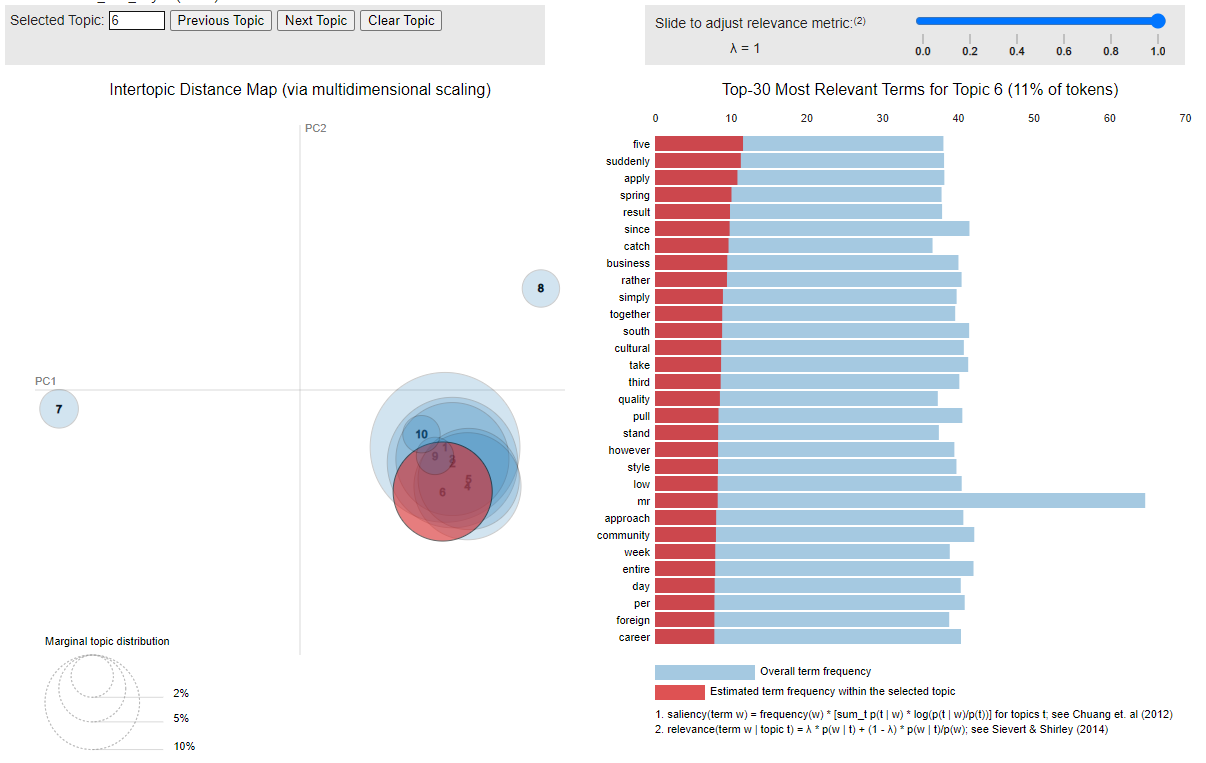
Topic 4:



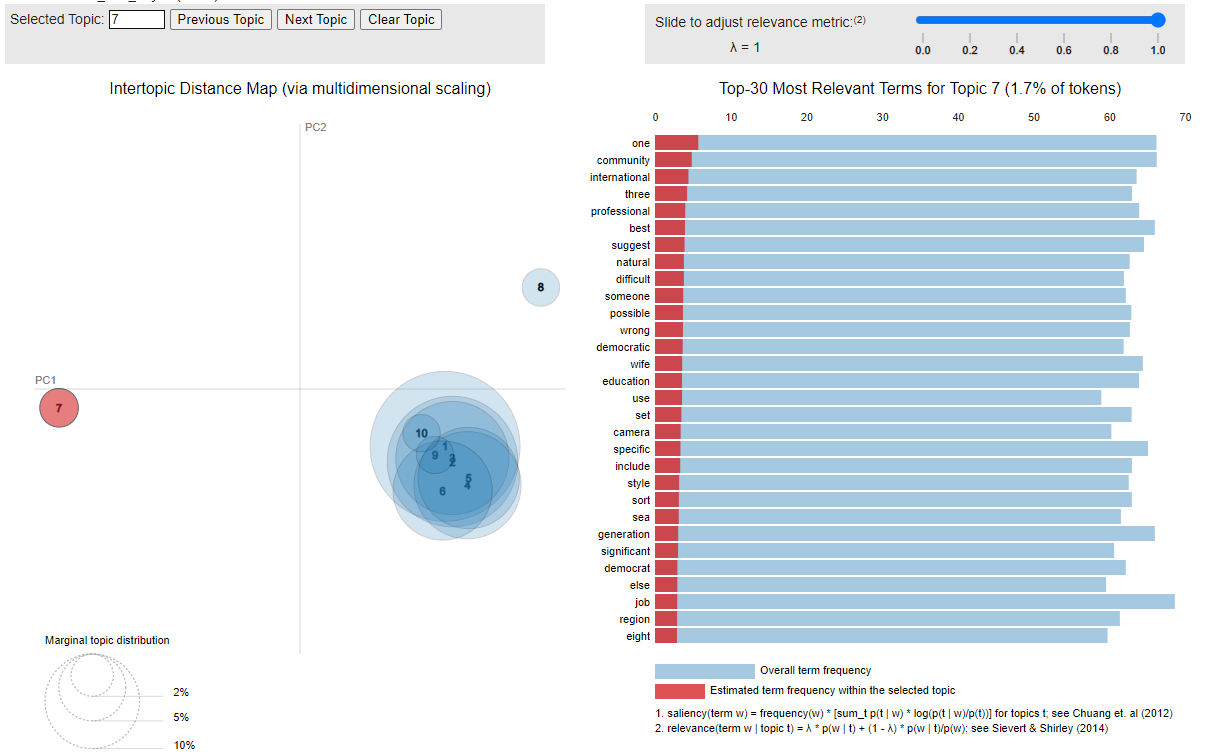
Topic 5:



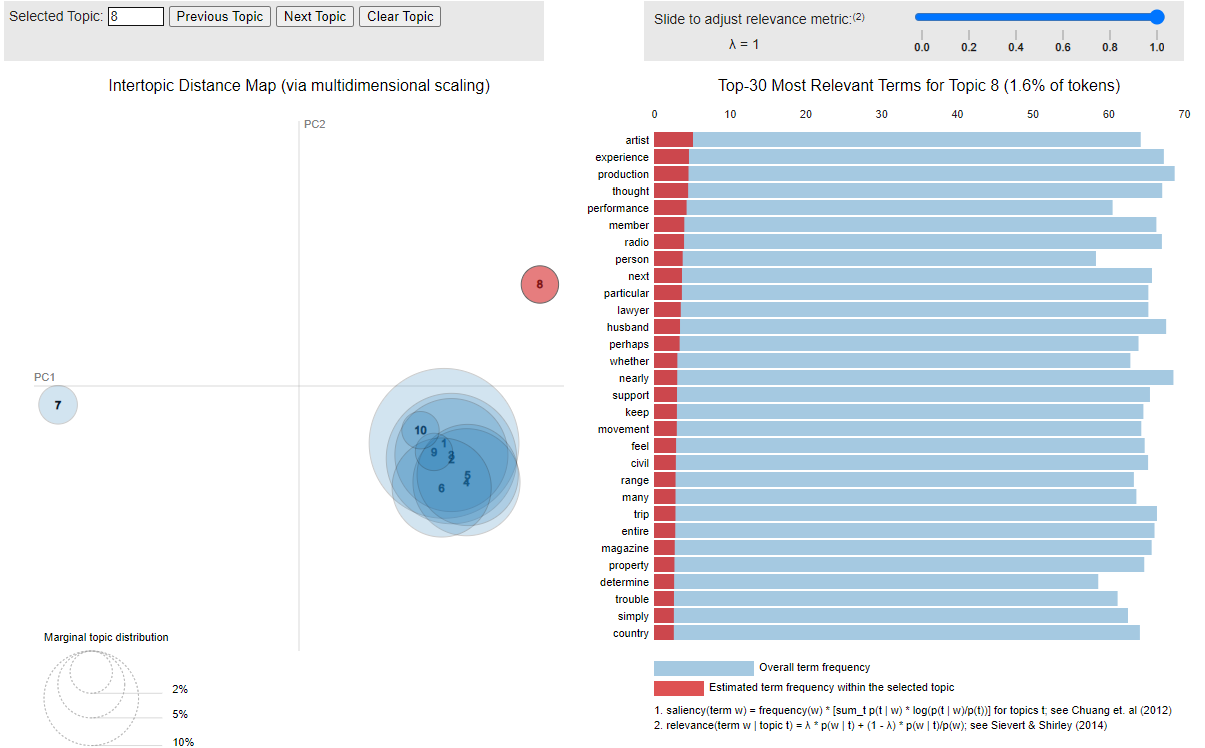
Topic 6:



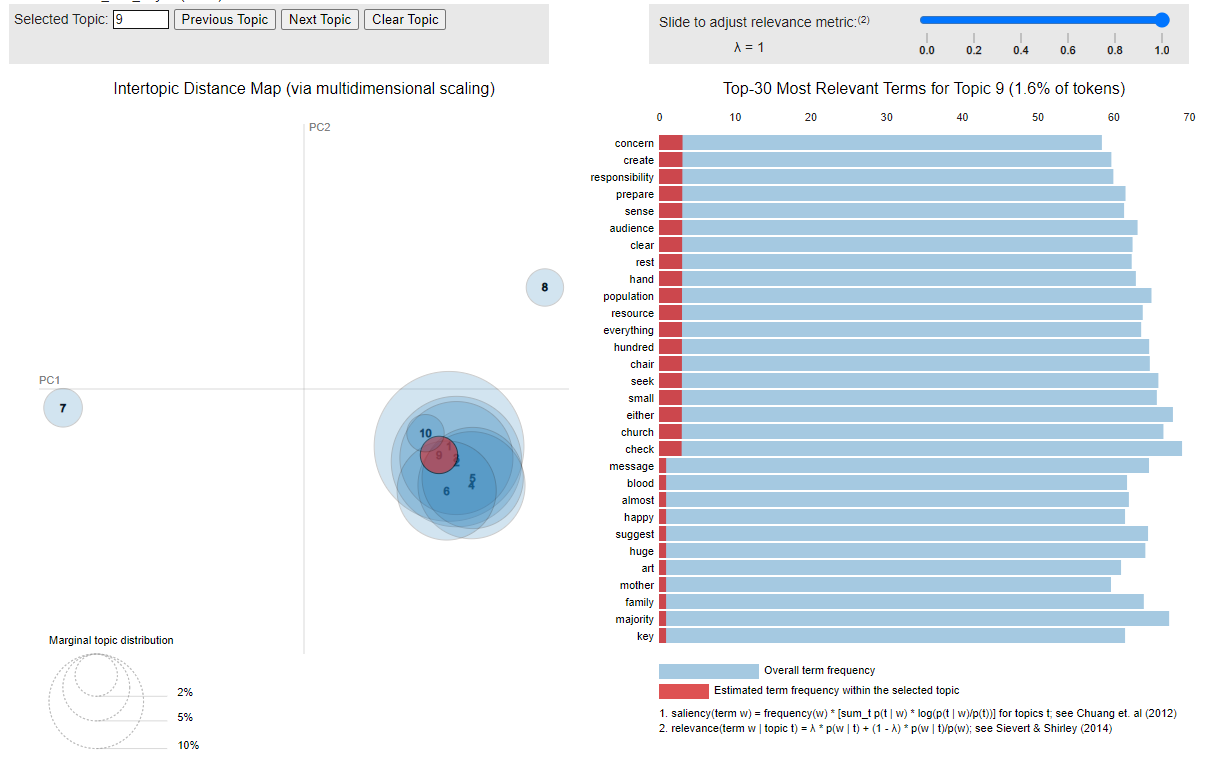
Topic 7:

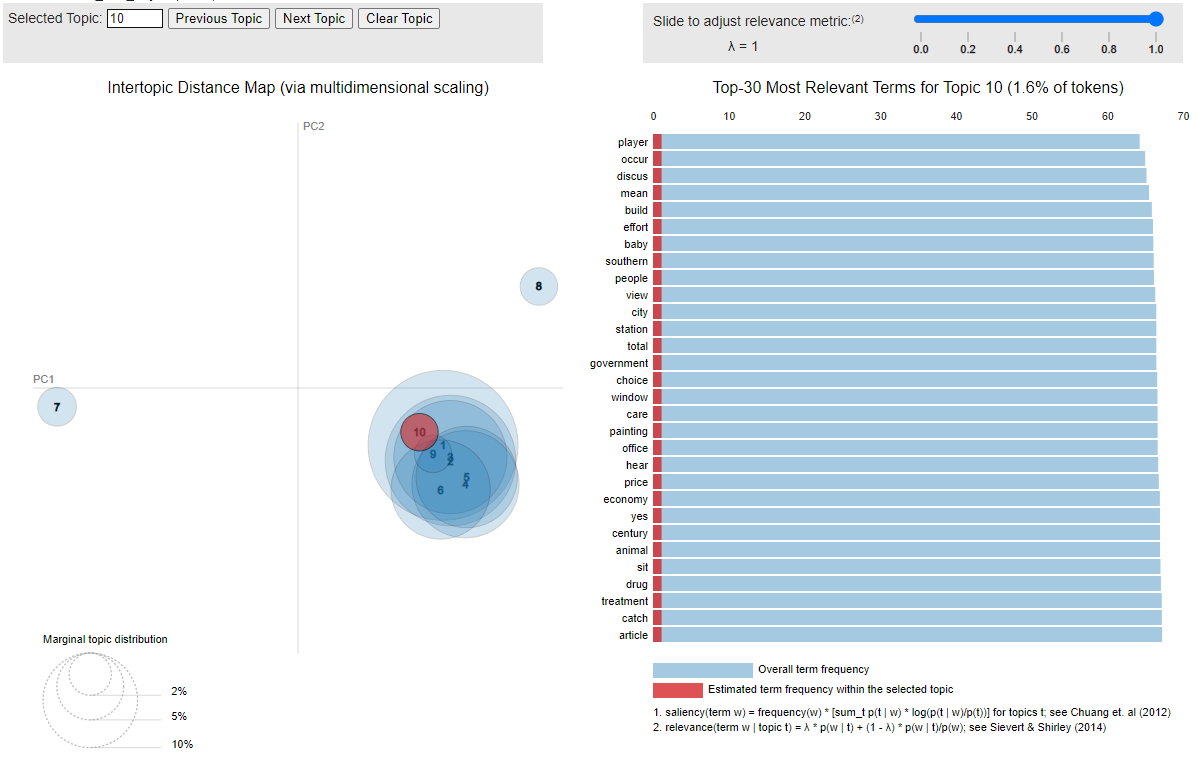


Topic 8:



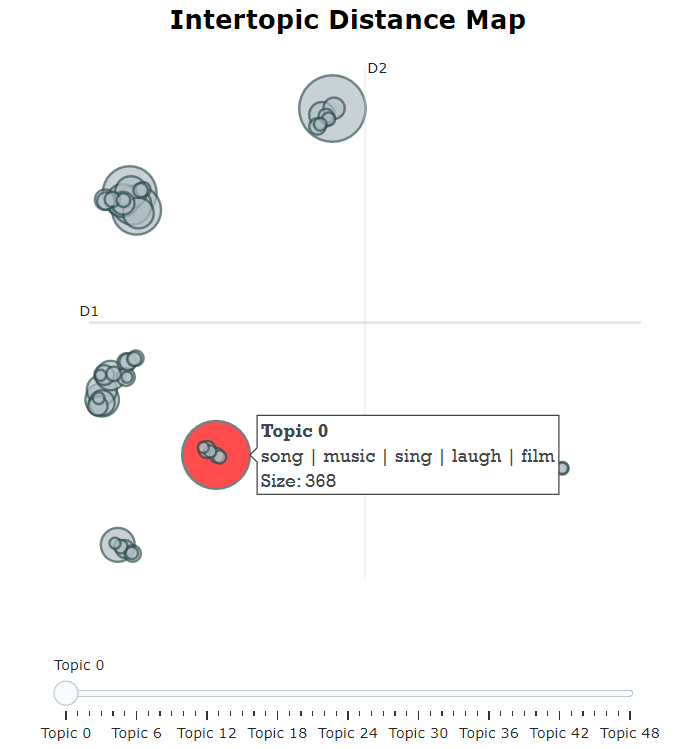
Topic 9:



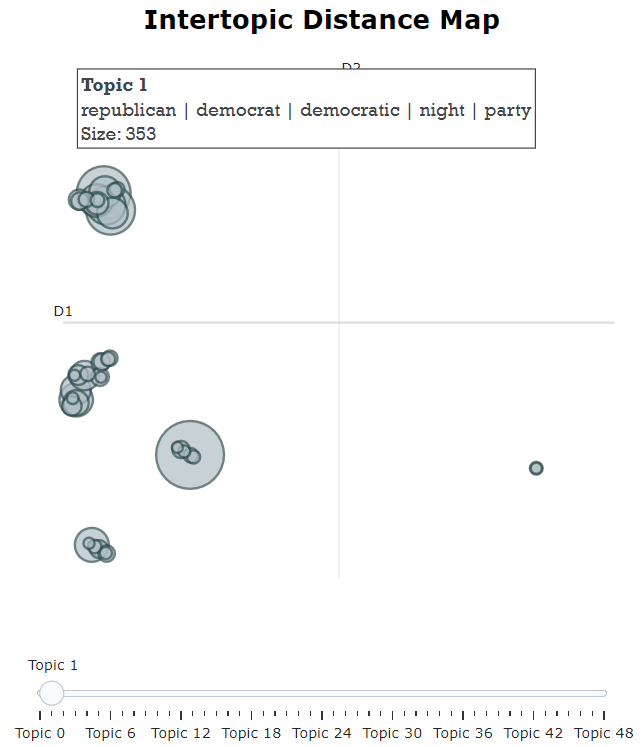
Topic 10:  


BERTopic Visualization:

To visualize the topics, pyLDAvis is used to interpret the clusters of topics. Following is the visualization for top 10 topics:  
Topic 1:



Topic 2:

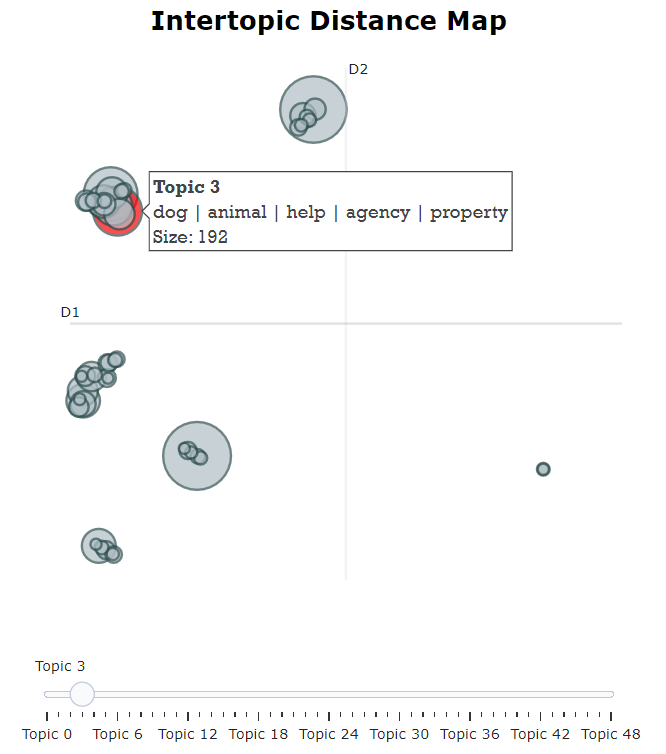


Topic 3:

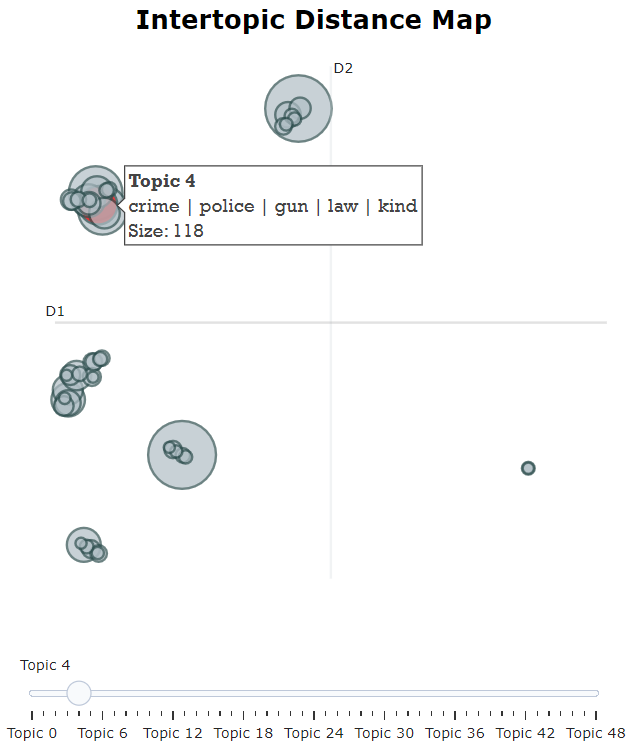
A screenshot of a video

Description automatically generated

Topic 4:



Topic 5:



Topic 6:

A screenshot of a computer

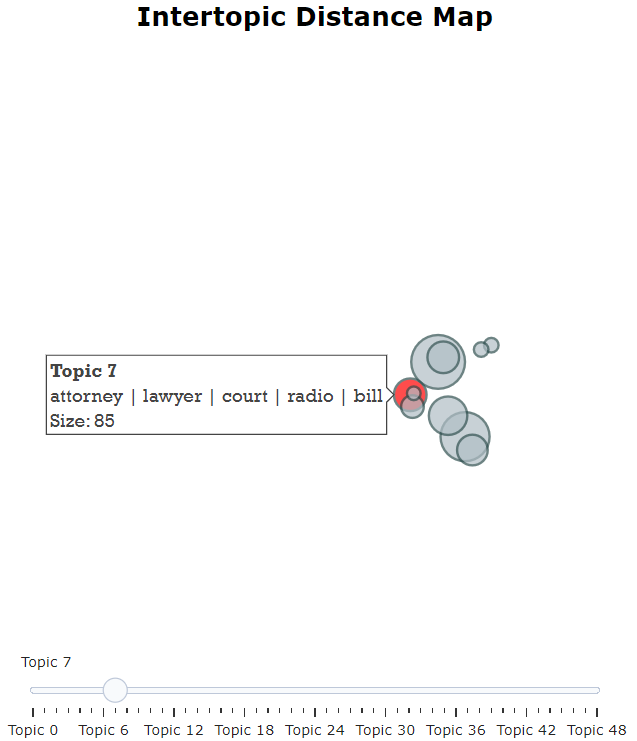
Description automatically generated

Topic 7:

A screenshot of a computer

Description automatically generated

Topic 8:



Topic 9:

A screenshot of a computer

Description automatically generated

Topic 10:

A screenshot of a computer

Description automatically generated

### Interpretation:

Interpreting the Top Words: In the same way as with Topic and Word, when referring to the general Top Words of the subjects described by the model or using word clouds to compare the frequencies of the terms related to each topic, one can deduce meanings or implications that provide further understanding of the topics or subjects which the model identifies.

These topics can make the use of structural analysis and can be characterized and labeled based on the most noticeable words which helps the reader to find the structure and contents of the text data easier.

Comparison:

The mean distance value of approximately 196.33 denotes the average dissimilarity between the topics produced by the LDA and BERTopic algorithms. This indicates substantial disparities in the topics identified by the two algorithms. Such dissimilarity may arise from differences in the underlying assumptions, feature representations, or clustering techniques employed by each algorithm. For example, LDA relies on bag-of-words representations and probabilistic modeling, whereas BERTopic utilizes BERT embeddings and density-based clustering.

### Future Work:

The aim is to use other topic modeling algorithms and, subsequently to determine the performance difference between them and analyze the various techniques in detail. Enhancing the results by training the hyperparameters to perform the topic modeling tasks in line with the data sets provided. Attractiveness of deepening the text analysis by integrating sentiment analysis or entity recognition techniques. To sum up, this is the beginning work on text processing and topic modeling required to proceed to future work on more accurate models and interpretations.

**Conclusion:**

The topic modeling techniques, LDA and BERTopic, were applied to a Twitter dataset, revealing distinct sets of topics. While LDA identified broad, abstract themes, BERTopic uncovered more specific and contextual topics.The mean distance between the two models' topics indicates substantial dissimilarity, likely due to their differing methodologies.Future work could explore other algorithms, hyperparameter tuning, and integrating sentiment analysis or entity recognition for enhanced text analysis.Overall, this study serves as a foundation for further research into accurate topic modeling and interpretation techniques.

# References:

*BERTopic*. (n.d.). Retrieved from maartengr.github.io.

Malzer, C., & Baum, M. (2019). HDBSCAN($\hat{\epsilon}$): An Alternative Cluster Extraction Method for HDBSCAN. *arXiv: Databases*. Retrieved 5 30, 2024, from https://arxiv.org/abs/1911.02282

*Matplotlib*. (n.d.). Retrieved from matplotlib.

McInnes, L., & Healy, J. (2018). UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. *arXiv: Machine Learning*. Retrieved 5 30, 2024, from https://arxiv.org/abs/1802.03426

*Natural Language Toolkit*. (n.d.). Retrieved from NLTK.

*Pandas*. (n.d.). Retrieved from pandas.pydata.

*pyLDAvis*. (n.d.). Retrieved from pypi.

*Scikit-learn*. (n.d.). Retrieved from scikit-learn.

*sklearn.lda.LDA*. (n.d.). Retrieved from scikit-learn.

*Twitter-Dataset*. (n.d.). Retrieved from Kaggle.