Congressional Articles Mining

Paulina Gacek, Jakub Hulek, Filip Garbacki

November 28, 2024

1 Introduction

The United States House of Representatives, a cornerstone of American democracy, is composed of 435 elected officials. Each representative, also referred to as a congressman or congresswoman, serves a two-year term, advocating for the needs of their specific congressional district.

2 Project Overview

This project employs data mining and machine learning techniques to analyze textual data produced by U.S. Congress members, offering insights into political communication trends. By collecting a comprehensive dataset of articles and statements from various U.S. Congressmen and Congresswomen, we aim to explore key political patterns and their underlying motivations.

2.1 Considered Classification Tasks

2.1.1 Author Prediction

Initially, we considered developing a machine learning model to predict the author of an article based on linguistic and stylistic patterns. However, due to the frequent mention of the author's name within the text, this task proved to be trivial and not particularly insightful.

2.1.2 Political Affiliation Prediction

Another potential classification task was to predict the political party affiliation (Democrat or Republican) of the article's author. While this could be an interesting analysis, the limited number of classes (only two) and the often explicit mention of party affiliation within the text made this task too straightforward.

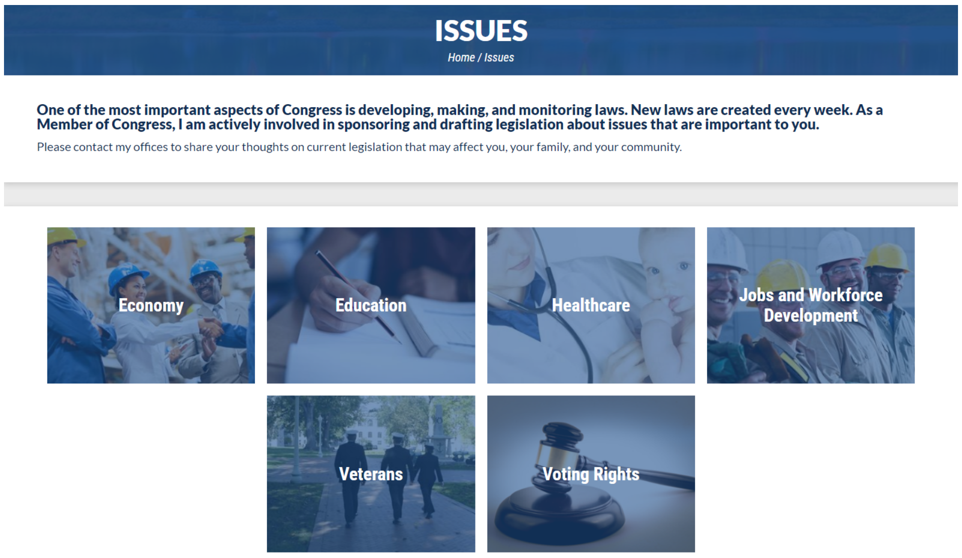
2.1.3 Topic Classification

Ultimately, we focused on the task of topic classification. This involved assigning articles to specific categories or topics based on their content. While the availability of predefined labels (the issues under which articles were posted) provided a solid foundation, we encountered challenges due to inconsistencies and non-standardization in the labeling process.

3 Data Acquisition

A list of all United States Representatives was compiled from the official House website: Directory of Representatives. This data includes each representative's full name, district, state, committee assignments, party affiliation, and a link to their individual webpage. Many of these individual webpages contain a dedicated "Issues" subpage where representatives outline the topics they address in their published articles.

For illustrative purposes, Figure 1 presents a screenshot of a sample congressman's (Rep. Terri Sewell) "Issues" page.



ISSUES

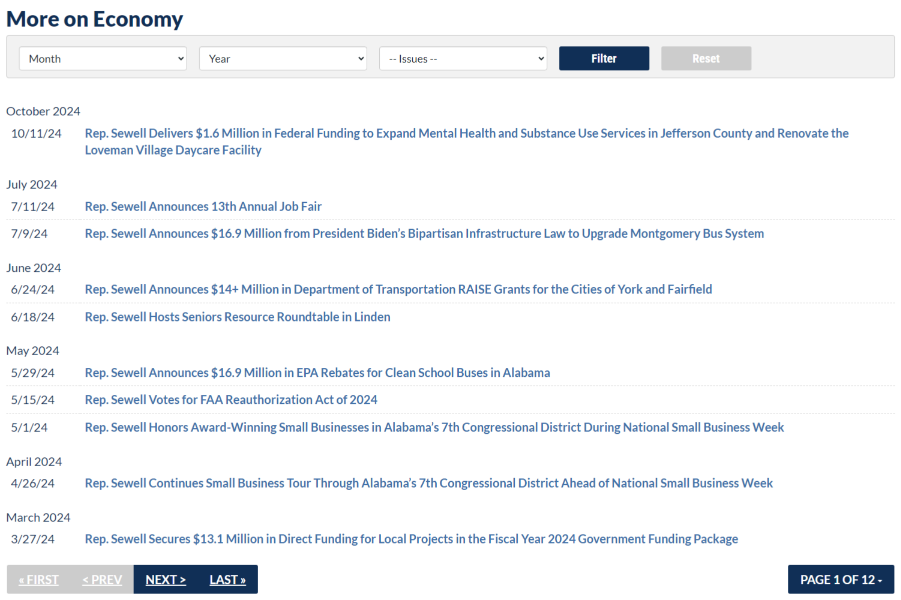
Home / Issues

One of the most important aspects of Congress is developing, making, and monitoring laws. New laws are created every week. As a Member of Congress, I am actively involved in sponsoring and drafting legislation about issues that are important to you.

Please contact my offices to share your thoughts on current legislation that may affect you, your family, and your community.

Figure 1: A screenshot of Rep. Terri Sewell issues page.

Each listed issue includes links to recently published articles and navigation buttons that allow access to older articles. Figure 2 showcases Rep. Terri Sewell's articles on the topic of Economy.



More on Economy

October 2024

10/11/24 Rep. Sewell Delivers $1.6 Million in Federal Funding to Expand Mental Health and Substance Use Services in Jefferson County and Renovate the Loveman Village Daycare Facility

July 2024

7/11/24 Rep. Sewell Announces 13th Annual Job Fair

7/9/24 Rep. Sewell Announces $16.9 Million from President Biden's Bipartisan Infrastructure Law to Upgrade Montgomery Bus System

June 2024

6/24/24 Rep. Sewell Announces $14+ Million in Department of Transportation RAISE Grants for the Cities of York and Fairfield

6/18/24 Rep. Sewell Hosts Seniors Resource Roundtable in Linden

May 2024

5/29/24 Rep. Sewell Announces $16.9 Million in EPA Rebates for Clean School Buses in Alabama

5/15/24 Rep. Sewell Votes for FAA Reauthorization Act of 2024

5/1/24 Rep. Sewell Honors Award-Winning Small Businesses in Alabama's 7th Congressional District During National Small Business Week

April 2024

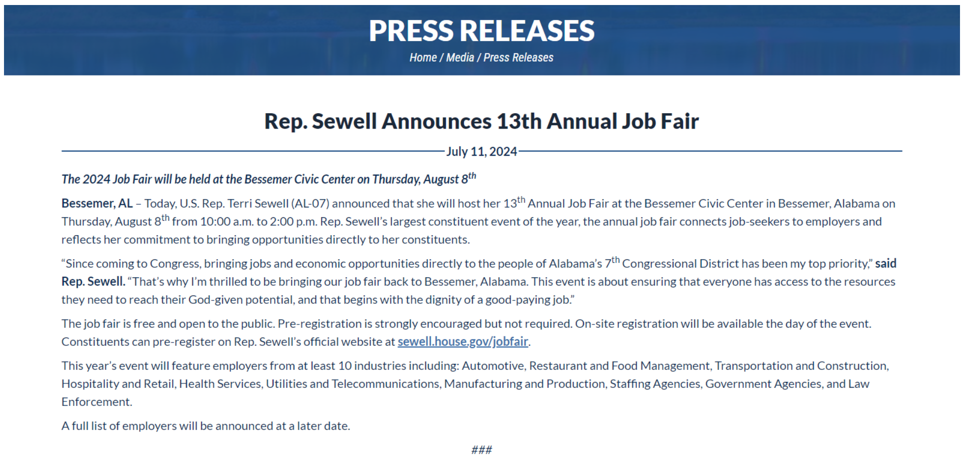
4/26/24 Rep. Sewell Continues Small Business Tour Through Alabama's 7th Congressional District Ahead of National Small Business Week

March 2024

3/27/24 Rep. Sewell Secures $13.1 Million in Direct Funding for Local Projects in the Fiscal Year 2024 Government Funding Package

Figure 2: Rep. Terri Sewell's economy articles

We extracted and saved the articles linked within these issue-specific lists. An example article is presented in Figure 3.



PRESS RELEASES

Home / Media / Press Releases

Rep. Sewell Announces 13th Annual Job Fair

July 11, 2024

The 2024 Job Fair will be held at the Bessemer Civic Center on Thursday, August 8th

Bessemer, AL – Today, U.S. Rep. Terri Sewell (AL-07) announced that she will host her 13th Annual Job Fair at the Bessemer Civic Center in Bessemer, Alabama on Thursday, August 8th from 10:00 a.m. to 2:00 p.m. Rep. Sewell's largest constituent event of the year, the annual job fair connects job-seekers to employers and reflects her commitment to bringing opportunities directly to her constituents.

"Since coming to Congress, bringing jobs and economic opportunities directly to the people of Alabama's 7th Congressional District has been my top priority," **said Rep. Sewell.** "That's why I'm thrilled to be bringing our job fair back to Bessemer, Alabama. This event is about ensuring that everyone has access to the resources they need to reach their God-given potential, and that begins with the dignity of a good-paying job."

The job fair is free and open to the public. Pre-registration is strongly encouraged but not required. On-site registration will be available the day of the event. Constituents can pre-register on Rep. Sewell's official website at sewell.house.gov/jobfair.

This year's event will feature employers from at least 10 industries including: Automotive, Restaurant and Food Management, Transportation and Construction, Hospitality and Retail, Health Services, Utilities and Telecommunications, Manufacturing and Production, Staffing Agencies, Government Agencies, and Law Enforcement.

A full list of employers will be announced at a later date.

###

Figure 3: Rep. Terri Sewell's sample article

The data scraping process began by accessing the Directory of Representatives' main page and creating individual scripts for each representative. These scripts were designed to extract information on issues, article lists, and article content.

Initially, we aimed to collect data from 100 representatives across both parties and all states by randomly selecting state representatives. However, this approach proved to be inefficient. We transitioned to a more targeted approach, scraping articles from individual representatives one by one. We developed scripts for 25 representatives and collected approximately 18,000 articles. Fortunately, we discovered that approximately 40% of the representatives utilized a common webpage provider, while another 10% shared a different provider. By applying our scraping algorithms to the websites of these providers for all representatives listed in the Directory of Representatives, we were able to acquire an additional 32,000 articles, resulting in a total of 50,000 articles.

To accelerate the processing, we implemented both multiprocessing and multithreading techniques. Multiple processes were employed to optimize CPU utilization for HTML processing, while multiple threads ensured that page downloads did not hinder ongoing computations. Figure 4 illustrates the parallel processing architecture utilized in our data scraping pipeline. Multiple processes, each employing numerous threads, were employed to concurrently download and process webpages. This parallel approach maximizes the utilization of both processing power and internet bandwidth, significantly accelerating the overall scraping efficiency. As a result, the downloading and scraping of 5000 pages was accomplished in a mere 6 minutes.

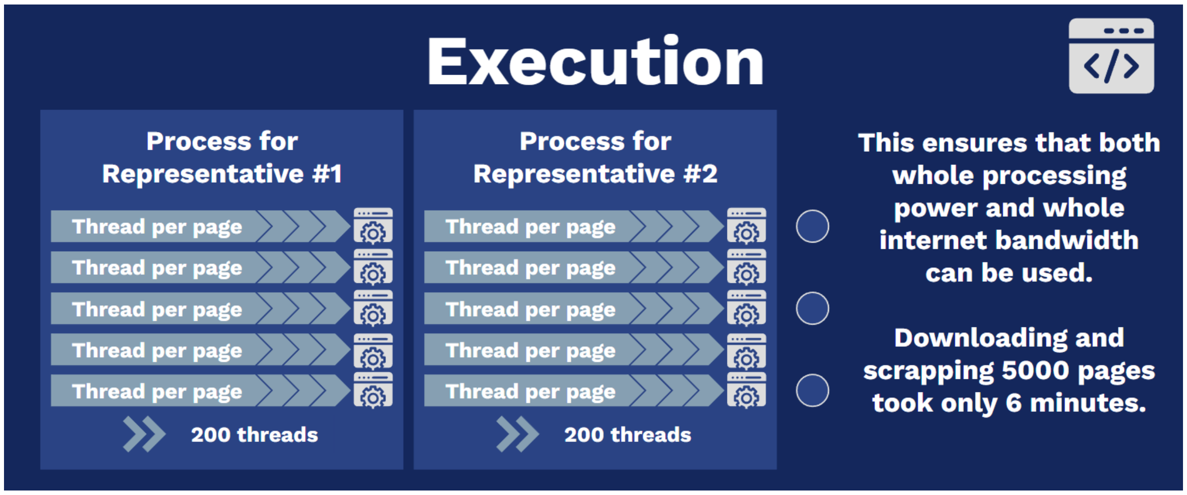

Execution
Process for Representative #1
Thread per page
Thread per page
Thread per page
Thread per page
Thread per page
200 threads
Process for Representative #2
Thread per page
Thread per page
Thread per page
Thread per page
Thread per page
200 threads
This ensures that both whole processing power and whole internet bandwidth can be used.
Downloading and scrapping 5000 pages took only 6 minutes.

Figure 4: Multiprocessing and multihreading overview

4 Data Preprocessing

4.1 Cleaning and formatting

The first step of the text preprocessing pipeline involves cleaning and formatting the raw textual data obtained from representatives' articles. This step aims to remove irrelevant information and standardize the format to ensure consistency for further analysis. We applied the following steps:

1. Multiple consecutive spaces were replaced with a single space, and leading/trailing spaces were removed. Additionally, newline characters and tabs were replaced with a single space to create a consistent text format.

2. Articles containing HTML code, often resulting from faulty formatting on representative websites, were removed to avoid interference with text analysis.

3. All articles' creation dates were standardized to a consistent date format.

4. Non-English articles, primarily Spanish, were excluded to maintain focus on English-language content and avoid potential language-related biases in the analysis.

4.2 Filtering by Character Count

After the initial cleaning and formatting, we filtered out articles that were either too short (less than 150 characters) or excessively long (more than 10,000 characters). This step removed noise and outliers, focusing the analysis on articles with substantial content. The figure 5 illustrates the character count distributions in the dataset before and after filtering. Initially, the distribution is highly skewed, with a concentration of very long texts, indicating the presence of outliers. Post-filtering, the distribution becomes more normalized and centered around a manageable range, making the dataset more suitable for analysis.

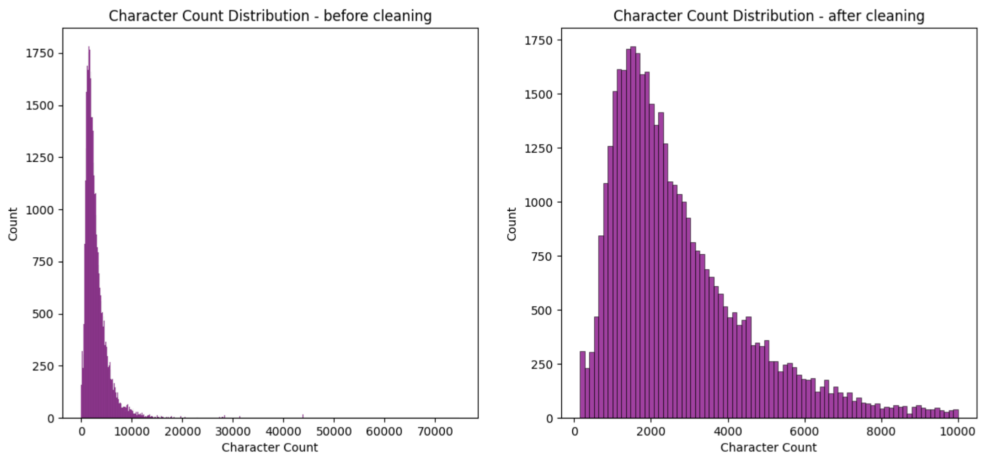


Figure 5: Distribution of number of characters in a single article.

4.3 Topics standardization

The next step involved standardizing the diverse range of topics assigned to articles by different representatives. While the original dataset contained **358** distinct topics, many of these were highly similar or overlapping. We manually categorized these topics into **19** standardized categories. For example, topics like "Health", "Healthcare", "What Rep. Flood is doing to expand healthcare options", "Social Security and Medicare," and "Improving Access to Affordable Healthcare" were all grouped under the broader category of "Healthcare and Social Security".

However, some topics proved challenging to categorize due to their specificity or ambiguity. Examples of such topics include "Back the Blue," "Getting Things Done," and "Crumbling Foundations."

These topics, which often reflected personal initiatives or local concerns, were classified as "Undefined" and subsequently removed from the dataset.

The figure 6 illustrates the distribution of articles across various categories by political party affiliation. It is evident that Democrats and Republicans exhibit distinct publishing patterns. Democrats tend to focus more on issues such as "Healthcare and Social Security," "Education" and "Equality and Civil Rights". In contrast, Republicans prioritize topics like "National Security, Defense, Foreign Affairs", "Government and Law" and "Pro-Life/Abortion and Family Values."

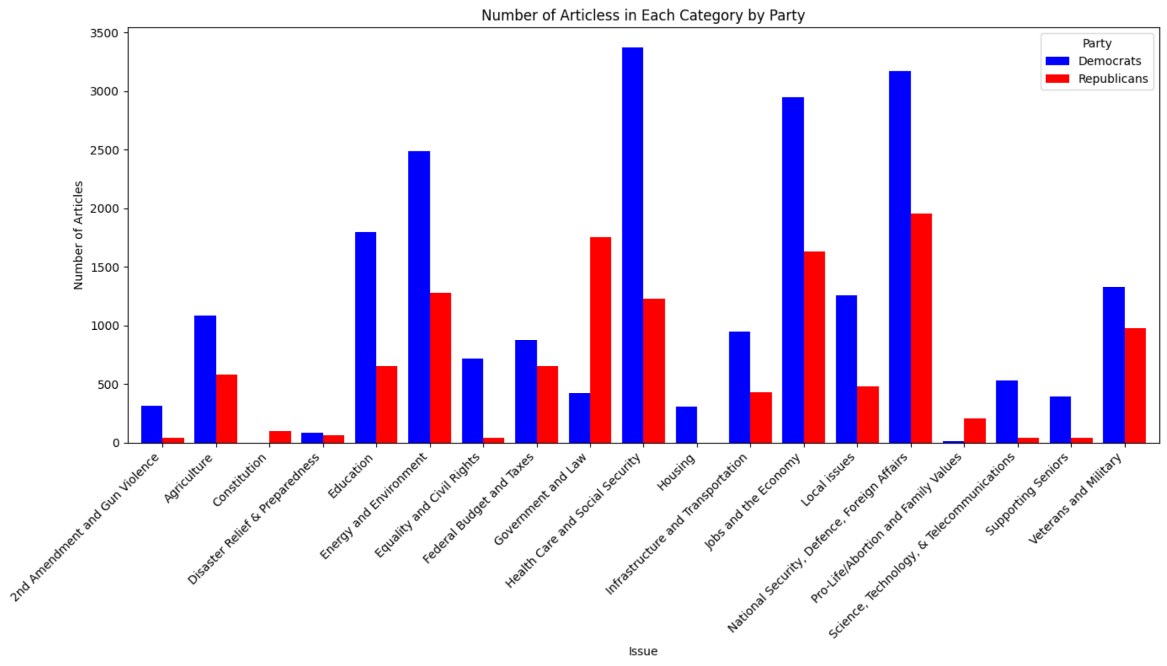


Figure 6: Number of articles in each topic by party.

4.4 Data Consolidation: Merging duplicated articles

Some articles were relevant to multiple categories and were consequently assigned to each of them. To avoid over-representation of certain textual pattern, duplicated articles were identified and merged into a single record with multiple associated labels.

4.5 Dataset Reduction Across Preprocessing Stages

The initial dataset comprised **50 303** articles. After applying the cleaning and formatting steps, the number of articles was reduced to **42 618**. Subsequent filtering based on character count further reduced the dataset to **41 662** articles. Topic standardization resulted in bringing the total count down to **34 230**. Finally, merging duplicate articles reduced the dataset to **25 644**, among which **19 765** articles were assigned to a single topic.

5 Explorative Data Analysis

Having the data prepared and ready, some EDA was performed to find insights into the dataset's structure and values, so that better understanding of the data could be achieved.

5.1 Topic's quantities

As the articles differ in the matter of topics that are associated with them, it was crucial to delve into the distribution of topics per article.

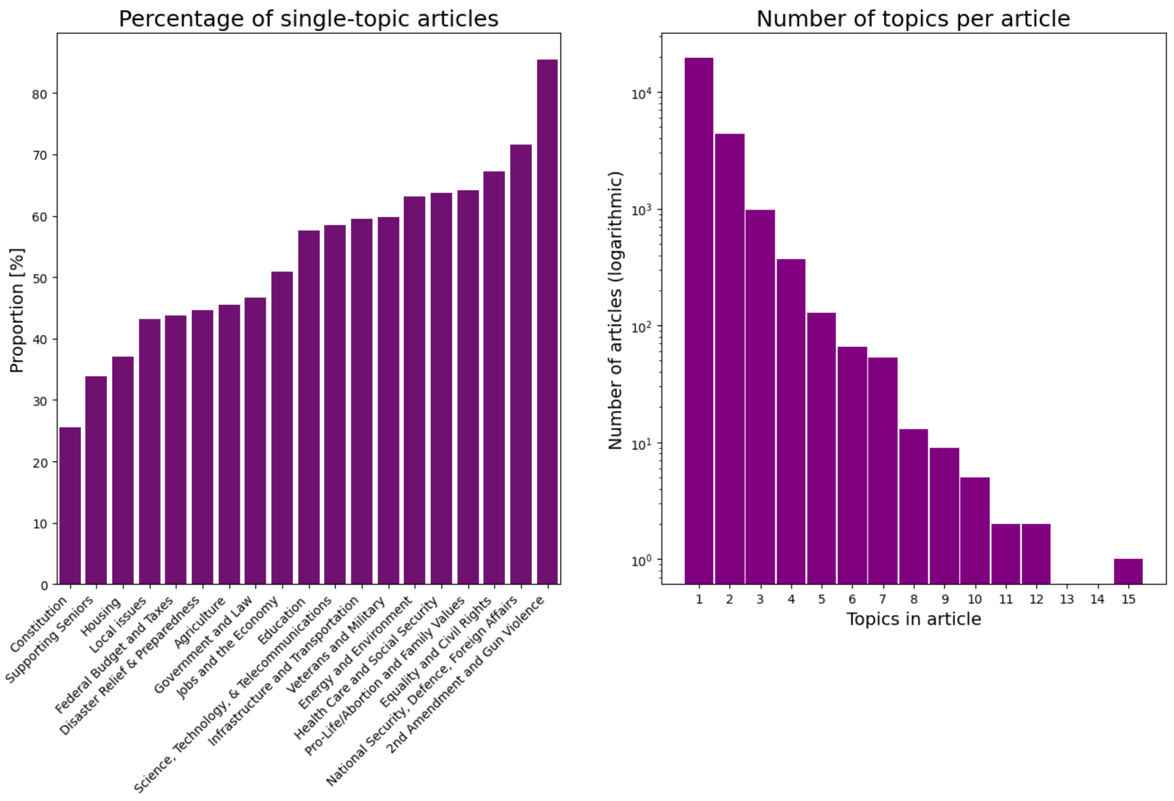


Figure 7: Topics per article distributions

It turned out that there are certain topics that tend to appear separately, while others, like "Constitution", "Supporting Seniors" or "Housing" more often than not happen to be a part of an article that contains multiple categories. On the other hand, over 80% of articles that regarded "2nd Amendment and Gun Violence" did occur on their own, without any other issues being mentioned.

What is also notable is the distribution of articles that relate to given amount of issues. While majority of articles refer only to a single topic, there is a significant number of articles that do tackle up to 4 topics at the same time. While there are examples of articles with 5 or more topics mentioned, their quantity is not that significant.

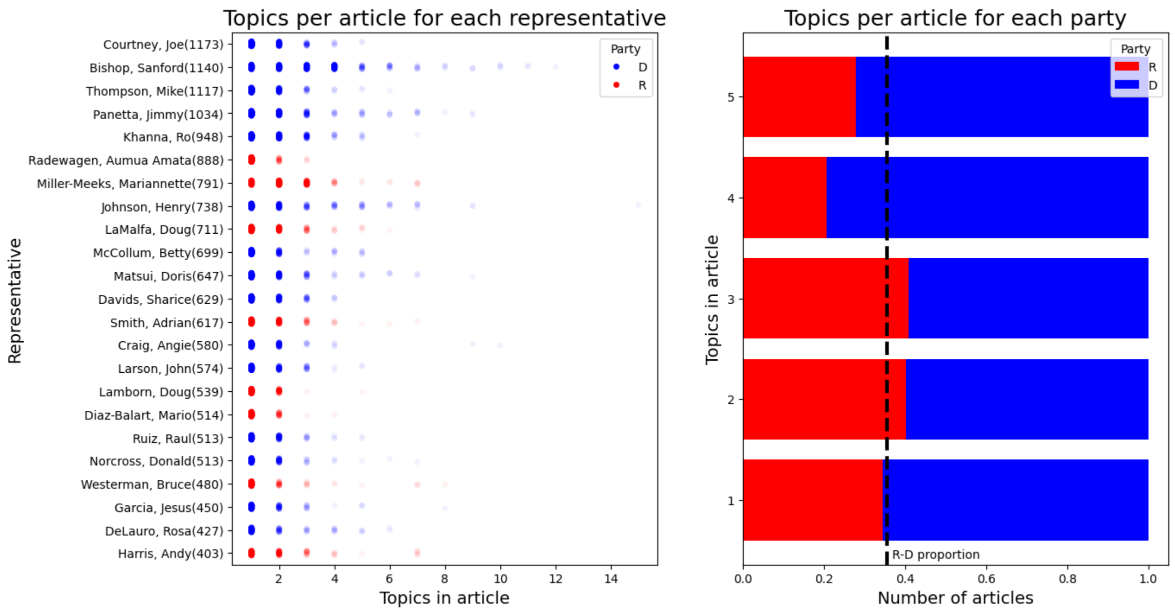


Figure 8: Topics per article in regard to representatives or parties. Only representatives with at least 400 articles are presented.

With further analysis it has been found that certain representatives tend to tackle multiple issues at once more often than the others. Considering Mrs. Aumua Amata Radewagen and Mrs. Marianette Miller-Meeks, it's clearly visible that although both congresswoman did contribute to the dataset in similar degree, the latter often includes multiple issues in one article, while Mrs. Radewagen rarely ever crosses the threshold of 2 topics per article.

There was also an attempt to see whether some parties tend to contain certain amount of topics more often than others. It turned out that there's higher likeliness to tackle multiple (4+) topics for Democrats, however it has to be said that the amount of records in those categories is rather little, thus this observation is not well-based in data. It is also interesting that proportion of single-topic articles is very similar to the articles proportion overall. It is actually corresponding to the distribution of topics per article - the well-fitted proportion is actually a consequence of what has been noticed before, that vast majority of articles is indeed only referring to one issue at a time.

5.2 Topic patterns

Since topics can coexist, there is a broad horizon of possibilities regarding pattern matching. A very important tool for taking a glimpse into that is the chord chart shown below.

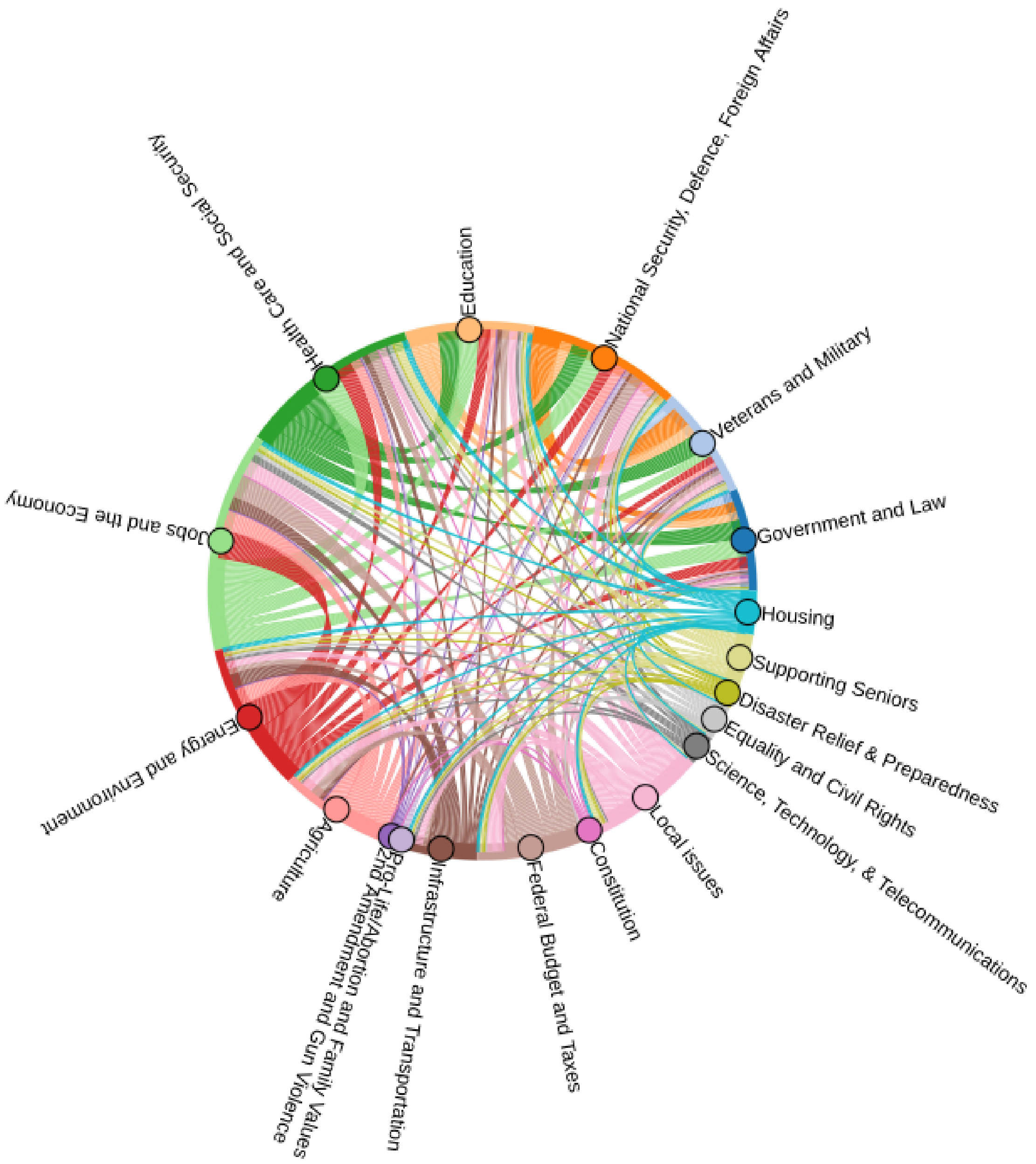


Figure 9: Chord chart presenting how frequently given topics appear together in an article.

While in its image version is rather unreadable, an interactive chord chart that Holoview library provides is an extraordinary tool that allows to easily track each topic's dependencies. An analysis of said chart has given some rather intuitive patterns - like "Education" does not often appear with "Agriculture, Energy and Environment", or that "Supporting Seniors" is often mentioned when talking about healthcare, social securities or veterans.

5.3 How the times change

The very nature of the politics is that it mirrors current national and global events. A very simple word frequency analysis was performed, based solely on counting occurrences of the word and some of its most derivative phrases or synonyms.

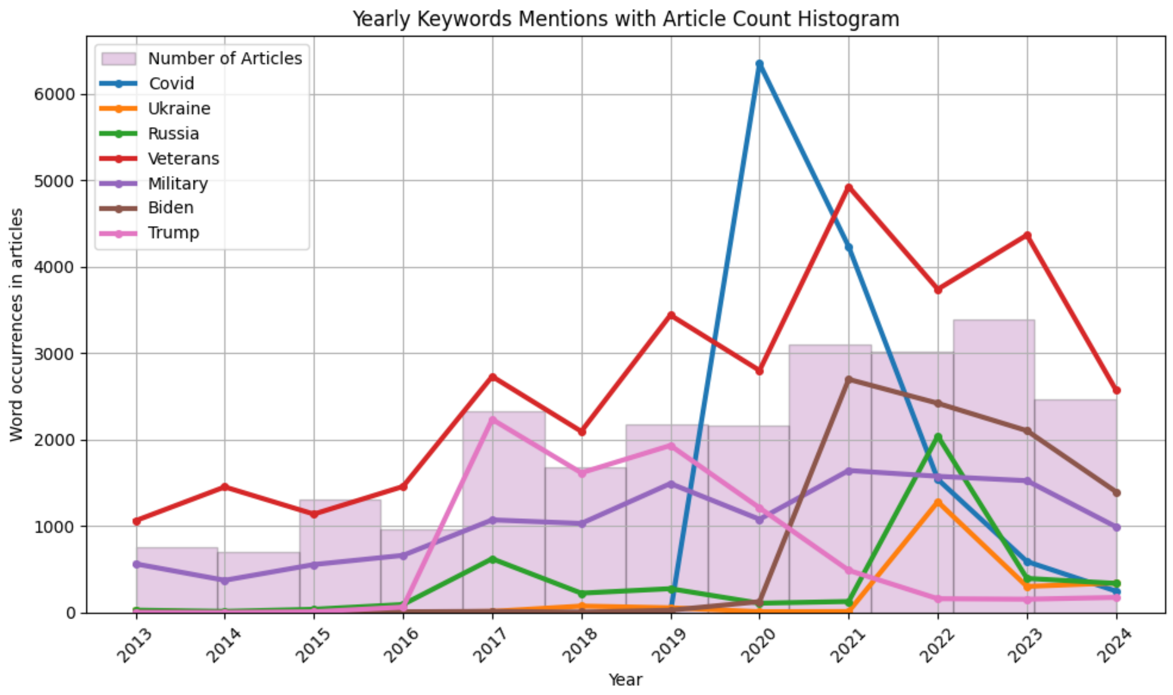


Figure 10: Time-based distribution of articles with frequencies of some example keywords

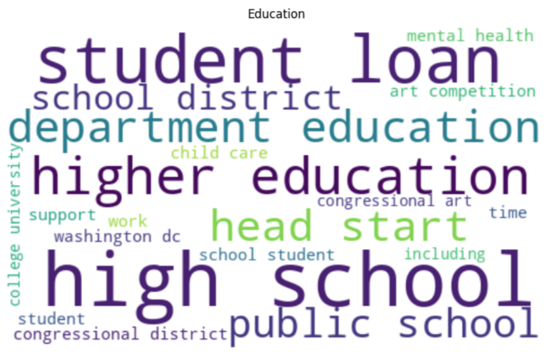
Keeping in mind that the frequency will strongly correlate with the number of records (represented by the semi-transparent histogram), it's clear that while military or veterans always are present in the congress debates, some events may result in significant increases of interest in some areas. The latter can be clearly seen by considering frequencies of Donald Trump's and Joe Biden's surnames being mentioned after they became presidents. And, of course, with no surprise the most rapid growth in interest belongs to Covid-19, as it was a rather short(dataset-wise), but very significant event.

6 Features extraction

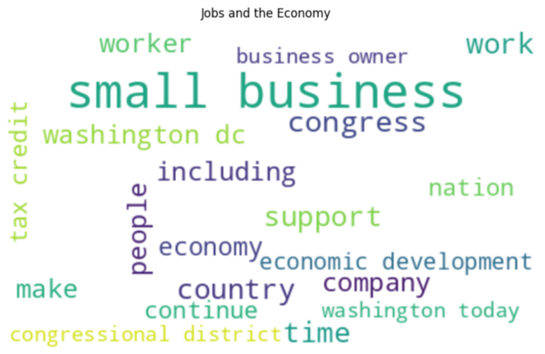
To gain a better understanding of the thematic content within different topics, we generated word clouds that visualize the most frequent phrases associated with six key issues: Education, Jobs and the Economy, Energy and Environment, Veterans and Military, Health Care and Social Security, and National Security, Defense, and Foreign Affairs. Word clouds are a simple yet effective way to highlight prominent phrases, as the size of each word reflects its frequency within the respective topic.

During the initial version of these visualizations, we observed that certain generic words—such as "bill," "state," "act," "community," "year," "American," "rep," "house," "program," "congressman," and "member"—were present across all categories. These words, while frequently used, did not provide meaningful differentiation among topics. To enhance the interpretability of the visualizations, we decided to exclude these common words, focusing instead on topic-specific key phrases.

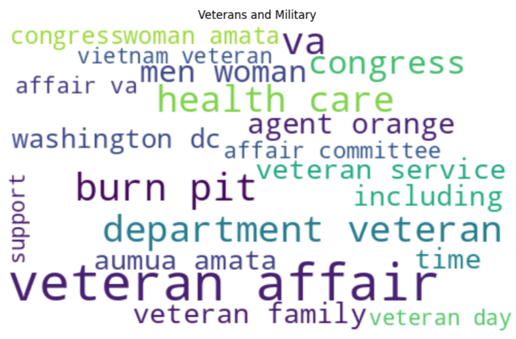
The refined word clouds, as shown in Figure 11, reveal unique and meaningful phrases for each category.



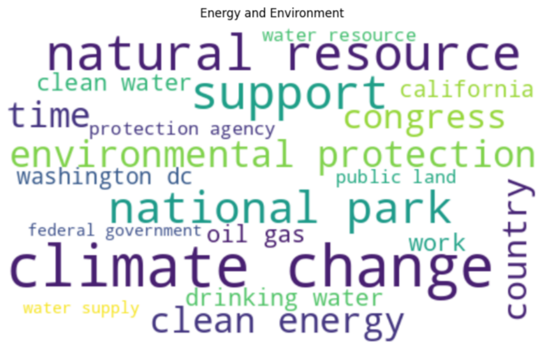
(a) "Education" topic



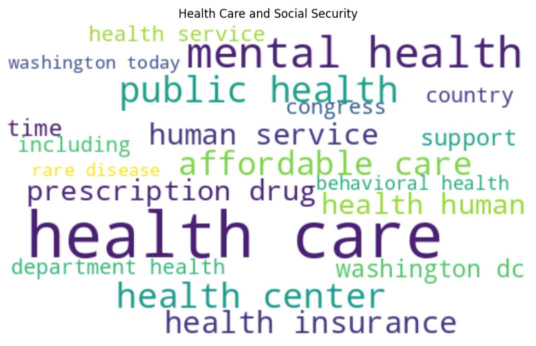
(b) "Jobs and the Economy" topic



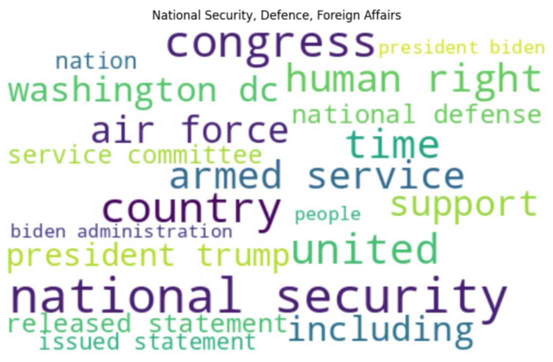
(c) "Veterans and Military" topic



(d) "Energy and Environment" topic



(e) "Health Care and Social Security" topic



(f) "National Security, Defence, Foreign Affairs" topic

Figure 11: The most frequent phrases for selected topics

7 Topic classification

Each article has assigned at least one topic from following topics:

Table 1: Final topics

Topic	Count
National Security, Defence, Foreign Affairs	3673
Health Care and Social Security	2930
Energy and Environment	2380
Jobs and the Economy	2333
Education	1411
Veterans and Military	1376
Government and Law	1015
Infrastructure and Transportation	823
Agriculture	758
Local issues	750
Federal Budget and Taxes	667
Equality and Civil Rights	511
Science, Technology, & Telecommunications	337
2nd Amendment and Gun Violence	306
Supporting Seniors	146
Pro-Life/Abortion and Family Values	141
Housing	115
Disaster Relief & Preparedness	67
Constitution	26

For the classification task, we have selected the major groups with the most data:

- National Security, Defence, Foreign Affairs (3496)
- Jobs and the Economy (2333)
- Energy and Environment (2380)
- Health Care and Social Security (2930)
- Education (1411)
- Veterans and Military (1376)

7.1 Preprocessing

In the preprocessing stage we have employed following methods:

1. Text Cleaning:
   - Hyperlinks (URLs) were removed from the text using regular expressions. This is because URLs often contain irrelevant information for the analysis and can introduce noise.
   - All characters except alphabets ([A-Za-z]) were replaced with spaces. This removes punctuation, symbols, and extraneous numbers that may not contribute significantly to the content analysis.
2. Word Tokenization using the NLTK library's function.
3. For stop word eemoval we used Smart Stop List available here.
4. Each remaining word was lemmatized using WordNetLemmatizer. This step allows for better identification of similar words with different morphological variations.

7.2 Articles Embedding

Word embedding is a vast field, so several approaches were considered to select the optimal solution.

7.2.1 Using pre-trained models

Numerous pre-trained models are readily available, ranging from lightweight options to massive models encompassing millions of records and requiring gigabytes of storage. The following table compares selected models based on their accuracy scores and neural network training times. For consistency, all results were obtained using the same neural network architecture and training/test datasets.

Table 2: Results of classification with different pre-trained embedding models.

Model	word2vec-google-news-300	fasttext-wiki-news-subwords-300	glove-twitter-200	glove-twitter-100	glove-twitter-50
Accuracy	85.68%	84.69%	84.65%	82.28%	77.35%
Training time	$225s$	$223s$	$160s$	$80s$	$49s$

The results indicate a clear inverse correlation between model complexity and training time. This relationship appears linear, with a proportional decrease in embedding vector length corresponding to a reduction in training time. While the accuracy differences are not substantial, models embedding words into vectors with more than 100 dimensions exhibit a clear advantage. It's important to note that, beyond size, the embedding methods employed by each model differ, with the exception of the 'glove-twitter-' family, which represents the same model.

7.2.2 Using word2vec

Given the library's support for the simple word2vec embedding method, we evaluated pre-trained models alongside custom models trained exclusively on our data. The results are quite promising, with nearly every custom model outperforming Google's previously best model. Custom models are denoted with the "Custom-" prefix.

Table 3: Comparison of Google's pre-trained model and locally trained models.

Model	word2vec-google-news-300	Custom-300, Window = 5	Custom-150, Window = 5	Custom-150, Window = 7	Custom-150, Window = 9
Accuracy	85.68%	85.22%	85.71%	86.21%	86.18%
Training time	$225s$	$228s$	$119s$	$119s$	$119s$

Interestingly, the custom model with a 300-dimensional embedding underperformed both in accuracy and training time compared to the 150-dimensional model. This discrepancy may be attributed to the simplicity of the neural network architecture, which likely struggled to effectively utilize the larger embedding space. A more complex network, longer training duration, and a lower learning rate could potentially yield better results with larger embeddings.

Those results, however, do contrast with the trend that pre-trained models have shown, where bigger embeddings did lead to better accuracy scores. It probably is the nature of the model that explain this phenomena. While US congress dataset certainly has broad range of topics tackled and, therefore, phrases used within, it is nowhere near the diversity of words that appear in news websites and social media that pre-trained models were trained with. That means that models used for transfer learning are not case-oriented, being as general as possible, so there's more different relations to describe that need more data. Smaller models are simply too general to allow very accurate word embedding. Custom model, on the other hand, is strictly case-oriented, meaning it can afford smaller embeddings, as it is exposed to more specific data. It is very likely that, when compared with some other type of problem, pre-trained models would outperform custom one.

While vector size parameter has undeniable impact on embedding quality, another parameter was tested, specific to word2vec method. "Window" parameter stands for the neighbourhood based on which the word is embedded. As it turned out, larger windows tend to perform slightly better, although it has to be acknowledged that it is not always the case. If the text would be, for instance, a list of short bullet points not related to each other, large neighbourhood would cause unrelated sentences to overlap each other, making the embedding quite messy. That is, however, not the case in this situation, as articles long texts that are are consistent with their meaning.

7.2.3 Using Bag of Words

We also tested a Bag-of-Words (BoW) approach with ngram ranges of (1,4). To mitigate the impact of common words, we removed those appearing in all articles. Additionally, we filtered the corpus, retaining only phrases that occurred at least 100 times. This filtering was necessary to address memory constraints, as including all phrases resulted in an out-of-memory error: *Unable to allocate 404. GiB for an array with shape (11282, 4805714) and data type int64.*

This approach yielded impressive results, achieving 89% accuracy on the top 6 most significant topics and 81% accuracy across all topics.

7.2.4 Using BERT

While multiple approaches were taken with many models tested, it turned out that they were not performing as well as BERT. We used BERT solely as a feature extractor and then feed those embeddings into a separate neural network for classification. This approach significantly reduced computational cost compared to fine-tuning the entire BERT model.

1. Data Splitting: We began by splitting the preprocessed data into training and validation sets using scikit-learn library. We specified a 20% validation split size and ensure stratified sampling to maintain the label distribution across both sets.

2. Label Encoding: Next, the issue column containing category labels underwent label encoding with LabelEncoder. This converts categorical labels into numerical representations for efficient processing by the BERT model.

3. Embedding Generation: We leveraged a pre-trained Sentence Transformer model, specifically all-MiniLM-L6-v2, for generating article embeddings. The chosen model is a smaller and faster variant of the full BERT model, offering a balance between performance and computational efficiency.

7.3 Training Custom Deep Neural Network

We implemented couple of Deep Neural Networks with various architectures. Here is the architecture of the DNN which gave the best results:

- Input Layer: The network receives the pre-generated article embeddings as input.
- Hidden Layers: The network utilizes 3 fully-connected hidden layers with decreasing neuron counts (256, 128, 64).
- Activation Functions: Each hidden layer employs the ReLU (Rectified Linear Unit) activation function for non-linearity.
- Dropout Layers: Dropout layers are inserted after each hidden layer with varying dropout rates (0.4, 0.3, 0.2, 0.1). Dropout helps prevent overfitting by randomly dropping a certain percentage of neurons during training, encouraging the network to learn more robust features.
- Output Layer: The final layer has a number of neurons equal to the number of unique issue categories in the dataset (6).

We used the Cross Entropy Loss as the loss function and the Adam optimizer.

7.4 Results

The best results were obtained using BERT embeddings, and the evaluation metrics for the top six topics are summarized in Table 4. For these six topics, our model achieved an accuracy of 91% on unseen test data, despite the inherent imbalance in topic distribution. The classification metrics—precision, recall, and F1-score—are consistently high across all six topics, indicating robust performance.

Table 4: Classification Report for top 6 topic classification using BERT embedding.

	precision	recall	f1-score	support
Education	0.91	0.88	0.90	282
Energy and Environment	0.90	0.95	0.92	476
Health Care and Social Security	0.91	0.94	0.93	586
Jobs and the Economy	0.88	0.85	0.87	467
National Security, Defence, Foreign Affairs	0.93	0.94	0.94	735
Veterans and Military	0.92	0.85	0.88	275
accuracy			0.91	2821
macro avg	0.91	0.90	0.91	2821
weighted avg	0.91	0.91	0.91	2821

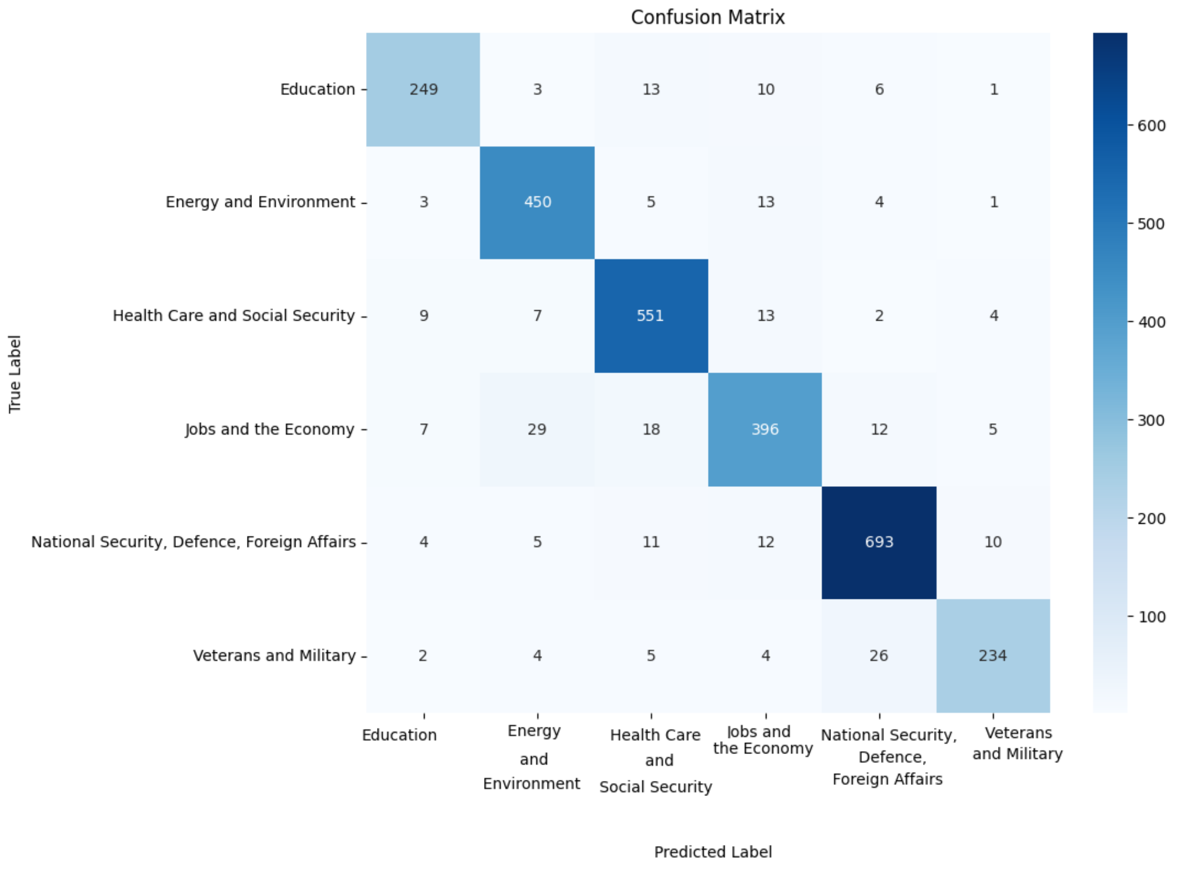
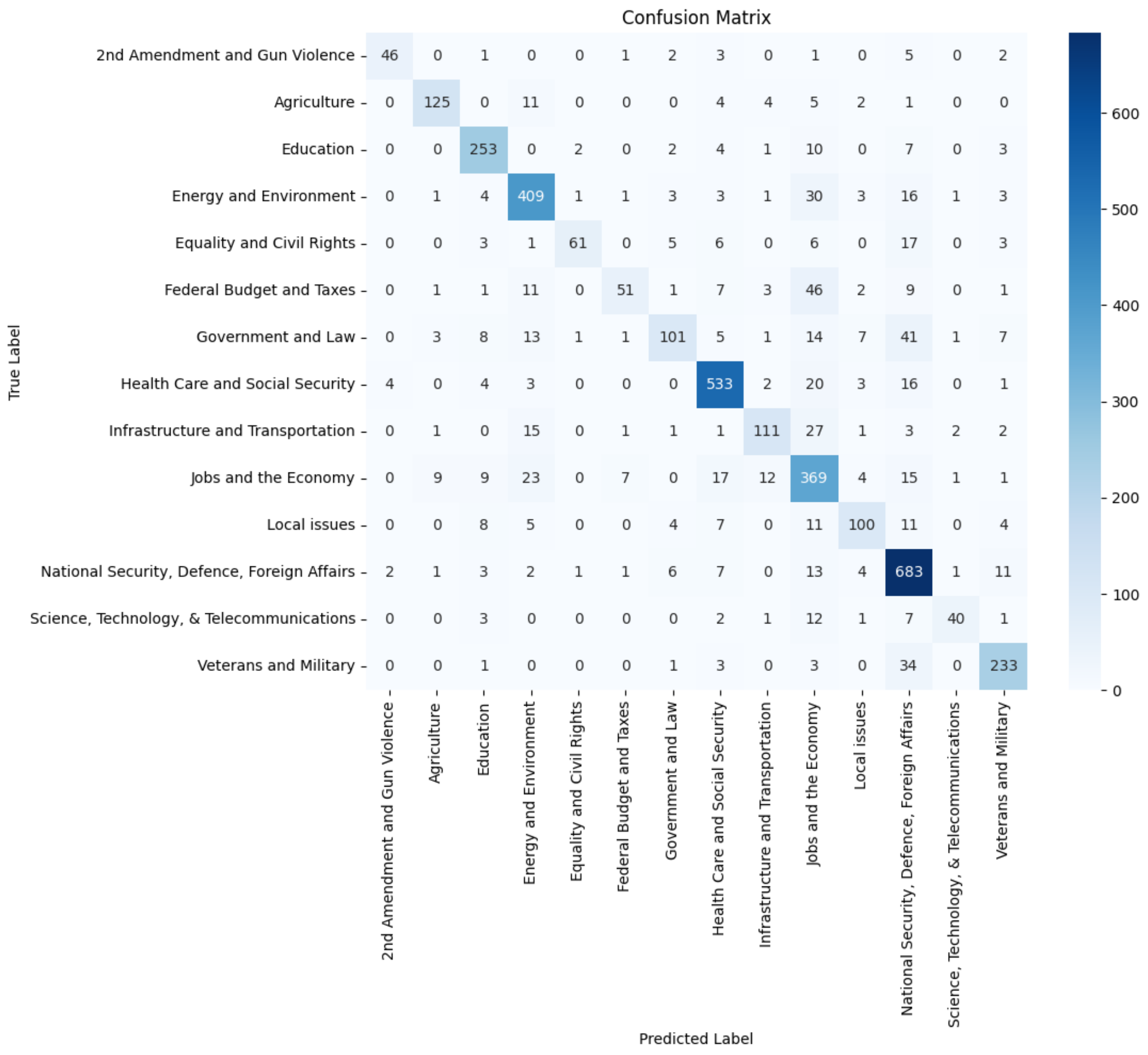


Figure 12: Final Confusion Matrix for top 6 topic classification using BERT embedding.

For the complete set of topics, the results are also encouraging. The confusion matrix for all topics counting over 150 articles (14 topics) suggests that the model can differentiate well between the categories. However, due to the increased complexity and further class imbalances, the overall accuracy drops to 82%. Despite this decrease, the performance remains strong and demonstrates the capability of our approach to handle multi-topic classification effectively.



True Label \ Predicted Label	2nd Amendment and Gun Violence	Agriculture	Education	Energy and Environment	Equality and Civil Rights	Federal Budget and Taxes	Government and Law	Health Care and Social Security	Infrastructure and Transportation	Jobs and the Economy	Local issues	National Security, Defence, Foreign Affairs	Science, Technology, & Telecommunications	Veterans and Military
2nd Amendment and Gun Violence	46	0	1	0	0	1	2	3	0	1	0	5	0	2
Agriculture	0	125	0	11	0	0	0	4	4	5	2	1	0	0
Education	0	0	253	0	2	0	2	4	1	10	0	7	0	3
Energy and Environment	0	1	4	409	1	1	3	3	1	30	3	16	1	3
Equality and Civil Rights	0	0	3	1	61	0	5	6	0	6	0	17	0	3
Federal Budget and Taxes	0	1	1	11	0	51	1	7	3	46	2	9	0	1
Government and Law	0	3	8	13	1	1	101	5	1	14	7	41	1	7
Health Care and Social Security	4	0	4	3	0	0	0	533	2	20	3	16	0	1
Infrastructure and Transportation	0	1	0	15	0	1	1	1	111	27	1	3	2	2
Jobs and the Economy	0	9	9	23	0	7	0	17	12	369	4	15	1	1
Local issues	0	0	8	5	0	0	4	7	0	11	100	11	0	4
National Security, Defence, Foreign Affairs	2	1	3	2	1	1	6	7	0	13	4	683	1	11
Science, Technology, & Telecommunications	0	0	3	0	0	0	0	2	1	12	1	7	40	1
Veterans and Military	0	0	1	0	0	0	1	3	0	3	0	34	0	233

Figure 13: Final Confusion Matrix for all topics classification using BERT embedding.

8 Key phrase extraction

Key phrases are concise, multi-word expressions that encapsulate the core thematic content of a document. They serve as condensed summaries, enhancing or even substituting titles and abstracts to facilitate efficient information retrieval.

Contrary to common assumptions, key phrases are not merely the most frequent words in a document but rather the terms that hold significant contextual and semantic importance. To achieve this, we developed a novel unsupervised algorithm for key phrase extraction and evaluated its performance on three benchmark datasets—three widely-adopted scientific publication datasets for key phrase extraction. The proposed method outperformed all existing approaches, achieving the best performance across all evaluation metrics and datasets.

Our method does not require any modifications to the architecture of pretrained language models nor does it introduce additional parameters, making it a simple yet powerful approach for key phrase extraction.

Using this method, we extracted $N = 15$ key phrases from all articles published between 2015 and 2024. We then analyzed the frequency of their usage across different articles to identify and measure changes and patterns over time.

8.1 Proposed algorithm for keyphrase extraction

The algorithm for extracting keyphrases from the provided document d can be divided into the following steps:

1. Candidate Keyphrases Generation

Given a single document d, a set of candidate keyphrases $C = c_1, c_2, ..., c_k$ is generated by splitting d based on a stop-word pattern. This is motivated by the observation that keyphrases often consist of multiple words and typically lack punctuation or stop words [1].

2. Document and Candidate Embeddings Creation

The algorithm creates embedding representations for both the document and each candidate keyphrase. A pre-trained BERT model [2] is employed for this purpose. These embeddings capture the semantic meaning of the text without requiring further fine-tuning of the model.

3. Candidates Probability Calculation

For each candidate keyphrase $c \in C$ a probability score, $prob$, is calculated to indicate the likelihood of it being a true keyphrase. This score is based on the number of words it contains, denoted by $word\ count(c)$. The function $prob(w)$ is a pre-defined function that captures the observed frequency distribution of phrase lengths within expert-annotated keyphrases from relevant datasets [3, 4, 5]. By leveraging this information, the function can assign higher probabilities to phrase lengths that are statistically more likely to represent keyphrases.

4. Scoring and Selection

A final score s is computed for each candidate $c \in C$. This score combines two factors: the cosine similarity between the candidate and document embeddings, and the previously calculated probability score. The cosine similarity metric reflects the semantic similarity between the candidate and the entire document. The final score is calculated using the following equation:

$$score(c) = cos\,similarity(embed(c), embed(d)) \times prob(word\,count(c)) \quad (1)$$

The top N candidates with the highest scores are selected as the keyphrases.

8.2 Evaluation

To evaluate the proposed method for keyphrase extraction more comprehensively, three widely-adopted scientific publication datasets were used: Inspec [3], SemEval-2010 [4] and SemEval-2017 [5], with a combined total of 2,729 abstracts. The datasets were transformed to consist of abstracts and corresponding keyphrases that appear in the abstracts. Keyphrases whose stemmed forms were absent in the stemmed versions of the abstracts were removed to ensure the reliability of keyphrase extraction results comparison. Table 5 presents a summary of these datasets, including the number of words per abstract, the number of keyphrases per abstract, and the total number of abstracts.

Table 5: Statistics of the datasets used for evaluation.

	Words per abstract			Keyphrases per abstract			
	Min	Mean	Max	Min	Mean	Max	Abstracts
Inspec	15	125	502	1	9	33	1991
Sem-Eval-2017	60	170	355	2	12	30	500
Sem-Eval-2010	53	412	12479	1	5	20	238

The performance of the keyphrase extraction model was assessed by comparing F1 scores of the top N predicted keyphrases, denoted as $F1@N$. Given that our approach attempts to match a fixed number of outputs against a variable number of ground-truth keyphrases, the results may not accurately reflect the quality of extraction. Therefore, the additional metric $F1@\mathcal{O}$, introduced by Yuan et al. in [6], was employed. It compares ground-truth phrases with the top k predicted keyphrases, where k is the total number of ground-truth keyphrases. Before determining if two phrases match, the Porter stemmer [7] was applied to each word in the phrase. Subsequently, macro-averaging was utilized to aggregate the evaluation scores for all testing samples.

As baseline models, popular keyphrase extraction methods were selected: statistics-based methods such as RAKE [1] and YAKE [8], as well as graph-based methods such as TextRank [9] and PositionRank [10]. Table 6 presents the results of F1 scores for @10, @15, and @$\mathcal{O}$ of the proposed solution (KeyExtract) and the baseline models on the three benchmark datasets.

Table 6: Comparison of F1 scores for baseline models and the proposed model.

	Inspec			Sem-Eval-2010			Sem-Eval-2017		
	@10	@15	@$\mathcal{O}$	@10	@15	@$\mathcal{O}$	@10	@15	@$\mathcal{O}$
YAKE	0.17	0.2	0.19	0.2	0.15	0.12	0.16	0.19	0.17
RAKE	0.29	0.34	0.29	0.13	0.15	0.05	0.2	0.24	0.23
TextRank	0.26	0.3	0.3	0.21	0.24	0.24	0.23	0.27	0.29
PositionRank	0.23	0.26	0.27	0.18	0.24	0.24	0.2	0.24	0.27
KeyExtract	**0.38**	**0.43**	**0.38**	**0.33**	**0.33**	**0.31**	**0.28**	**0.31**	**0.35**

The results show that the proposed solution achieves the best performance on all evaluation metrics across all the datasets. We acknowledge that the scores of the baseline models presented in other publications might not be completely comparable with the approach presented in this work due to the use of additional post-processing and filtering methods.

8.3 Time-Based Distribution of Articles Containing Selected Key Phrases

In the Exploratory Data Analysis section, we conducted a straightforward word frequency analysis, focusing on the occurrences of individual words. In contrast, key phrases can consist of multiple words, offering a more specific and nuanced representation of thematic content. For this analysis, we selected phrases that appeared as key phrases in at least 100 articles. We then calculated the total number of articles in which these key phrases were present, providing a more accurate depiction of trends over time.

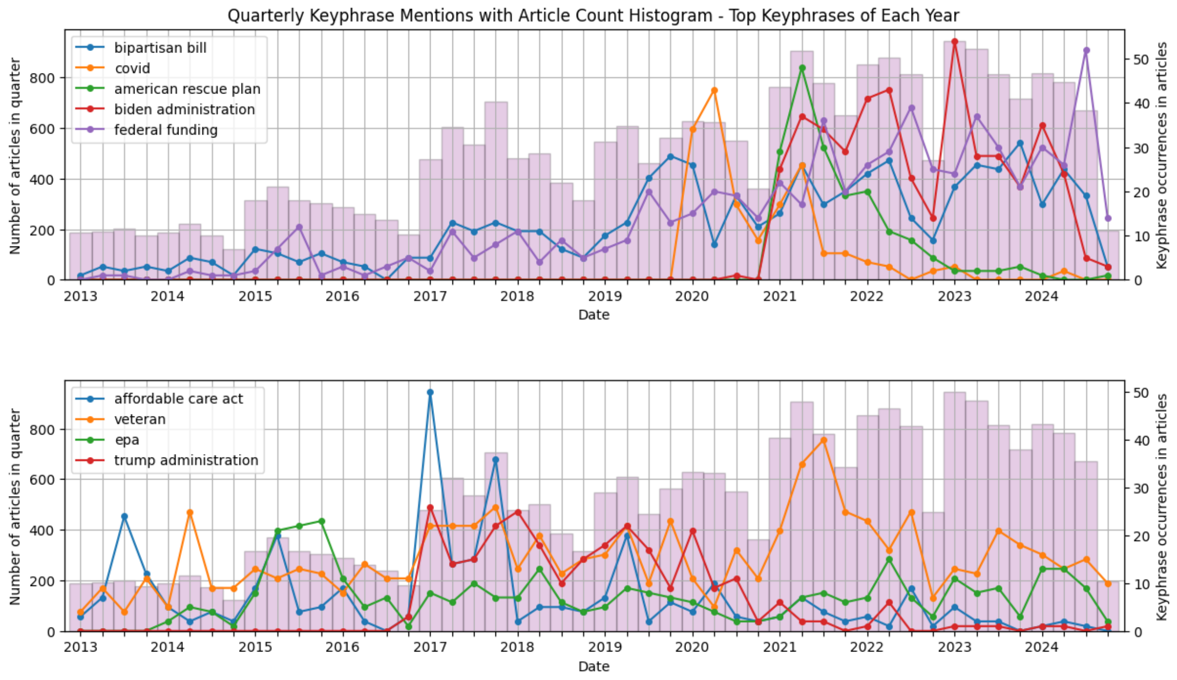


Figure 14: Frequency of selected key phrases since 2013.

Figure 14 showcases the most frequently occurring key phrases across the analyzed years. Due to the overlap in popular terms across multiple years, the total number of unique key phrases is lower than the number of years considered. Notably, the phrases "veteran" (2014, 2016), "affordable care act" (2013, 2017), and "Biden administration" (2022, 2023) appear in multiple years.

When examining the most common phrases across the entire timeline, the top 10 words closely resemble those identified as the most popular in specific years.

Table 7: 10 most frequently appearing key phrases between 2013 and 2024

Key phrase	Count
veteran	747
federal funding	660
bill	622
bipartisan bill	590
funding	569
national security	453
legislation	447
biden administration	445
representative	408
bipartisan legislation	403

References

[1] S. Rose, D. Engel, N. Cramer, and W. Cowley, *Automatic Keyword Extraction from Individual Documents*, pp. 1 – 20. 03 2010.

[2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)* (J. Burstein, C. Doran, and T. Solorio, eds.), (Minneapolis, Minnesota), pp. 4171–4186, Association for Computational Linguistics, June 2019.

[3] A. Hulth, "Improved automatic keyword extraction given more linguistic knowledge," in *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, pp. 216–223, 2003.

[4] S. N. Kim, O. Medelyan, M.-Y. Kan, and T. Baldwin, "SemEval-2010 task 5 : Automatic keyphrase extraction from scientific articles," in *Proceedings of the 5th International Workshop on Semantic Evaluation* (K. Erk and C. Strapparava, eds.), (Uppsala, Sweden), pp. 21–26, Association for Computational Linguistics, July 2010.

[5] I. Augenstein, M. Das, S. Riedel, L. Vikraman, and A. McCallum, "SemEval 2017 task 10: ScienceIE - extracting keyphrases and relations from scientific publications," in *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)* (S. Bethard, M. Carpuat, M. Apidianaki, S. M. Mohammad, D. Cer, and D. Jurgens, eds.), (Vancouver, Canada), pp. 546–555, Association for Computational Linguistics, Aug. 2017.

[6] X. Yuan, T. Wang, R. Meng, K. Thaker, P. Brusilovsky, D. He, and A. Trischler, "One size does not fit all: Generating and evaluating variable number of keyphrases," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, eds.), (Online), pp. 7961–7975, Association for Computational Linguistics, July 2020.

[7] M. F. Porter, "An algorithm for suffix stripping," *Program*, vol. 14, pp. 130–137, 1980.

[8] R. Campos, V. Mangaravite, A. Pasquali, A. Jorge, C. Nunes, and A. Jatowt, "Yake! keyword extraction from single documents using multiple local features," *Information Sciences*, vol. 509, pp. 257–289, 2020.

[9] R. Mihalcea and P. Tarau, "TextRank: Bringing order into text," in *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing* (D. Lin and D. Wu, eds.), (Barcelona, Spain), pp. 404–411, Association for Computational Linguistics, July 2004.

[10] C. Florescu and C. Caragea, "PositionRank: An unsupervised approach to keyphrase extraction from scholarly documents," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (R. Barzilay and M.-Y. Kan, eds.), (Vancouver, Canada), pp. 1105–1115, Association for Computational Linguistics, July 2017.