Sentiment Analysis of Political News Coverage: Topic Modeling and a Comparative Study of Vectorization Methods and Machine Learning Classification Algorithms

[1]Mahak Gupta, [2]Melissa Harrup, [3]Luke Kraemer, [4]Romzy Safadi

Northwestern University – School of Professional Studies
[1]mhkgup@gmail.com, [2]melissa.harrup@gmail.com, [3]kraemerluke@gmail.com,
[4]safadi.romzy@gmail.com

Abstract

The overwhelming volume of digital news published online every hour of the day poses a significant and practical challenge to political campaign managers who need to monitor how political interest topics are being framed by news outlets. This is particularly important leading up to a Presidential campaign when it is critical for a campaign team to respond quickly and strategically to editorial commentary and headlines. This problem can be addressed through topic modeling and implementing a combination of natural language processing (NLP) and machine learning (ML) classification algorithms to classify the sentiment of digital news coverage. In this paper, we conduct topic clustering and modeling with k-means and LDA, respectively. We also undertake a comparative analysis of three different NLP techniques, Bag-of-Words (BoW), TF-IDF and Doc2Vec, combined with three different ML classification algorithms, logistic regression, SVM, and naïve bayes, on a corpus of 44 news articles from both left-leaning and right-leaning news outlets. This paper reviews the relative accuracy of each combination of vectorization technique and ML algorithm. The best combinations, namely Naïve Bayes and SVM utilizing BoW techniques, delivered an accuracy of 86%.

Key words - NLP, SVM, Naïve Bayes, BoW, TF-IDF, Doc2Vec, Logistic Regression, Sentiment Analysis, LDA, k-means, Topic Clustering, Hyperparameter Tuning, Topic Modelling, Political News.

I. Introduction

Digital media and the democratization of digital journalism has replaced the once steady daily cycle of print news media with an internet filled with untold numbers of news articles, opinion pieces, and reports. The ability to glean real-time insights from the chaotic crescendo of digital news coverage leading up to a Presidential election is of critical importance to political campaign managers who need to respond quickly and strategically to editorial commentary and headlines (Kinga 2021). Campaign managers need to know what topics are trending in the news; if the news coverage is positive or negative; and whether the political leaning of the news outlet impacts sentiment. Natural Language Processing (NLP) offers computational models and techniques that can address these needs through topic clustering, topic modeling and sentiment analysis.

In this paper, we worked with a corpus of real-world digital news articles that covered political topics identified by the Biden-Harris Administration as being priority areas (The White House 2022). The corpus of articles was comprised of an equal number of articles sourced from left-leaning and right-leaning digital publications, respectively. We undertook a comparative analysis of how different combinations of vectorization methods and machine learning techniques may be utilized to summarize the sentiment of news coverage. Sentiment analysis (also known as opinion mining) is a branch of NLP that refers to computational models and techniques that distinguish whether text contains objective or subjective language and, if the latter, whether such "information is expressed in a positive, neutral or negative way" (Alonso et al. 2021). NLP techniques of Bag of Words (BoW), Term Frequency-Inverse Document Frequency (TF-IDF) and Doc2Vec were applied to pre-process and vectorize the data. K-means and Latent Dirichlet Allocation (LDA) machine learning algorithms were used to undertake

document clustering and topic modelling. Thereafter, logistic regression, support vector machine (SVM) and Naïve Bayes were used to classify the polarity of the textual data. The paper also explores the challenges inherent in sentiment analysis and identifies potential avenues for future research.

II. Literature Review

There is significant research interest in the utilization of artificial intelligence and deep learning techniques for text mining and sentiment analysis of written communications, including tweets and digital news. The literature demonstrates the effectiveness of vectorization models, such as BoW, in pre-processing data for use in sentiment analysis and LDA for topic modeling.

Blei, Ng and Jordan (2003) introduced the world to the application of LDA, a generative probabilistic model, to text corpora. The authors observed that LDA was an effective method for dimensionality reduction of a dataset where individual words were treated as discrete features. LDA continues to be a popular method for topic modeling and has been used in a variety of fields including in the analysis of political speeches for insight mining of political attention and priorities (Jelodar et al. 2019).

Sabbagh and Ameri (2020) utilized a combination of k-means clustering and LDA topic modeling to analyze unstructured manufacturing capability data. The combination of the two methods resulted in supplier clusters with high accuracy.

Mikolov et al. (2013) observed that simple model architectures such as continuous BoW (CBoW) and continuous skip-gram could be used to train high quality word vectors. The authors foreshadowed that the quality of word vectors would become an important component of NLP applications in the future. In a CBoW model the surrounding words are used to predict the word

in the middle. By contrast, in a continuous skip-gram model, the input word is used to predict the context.

Hossain et al. (2021) demonstrated how text mining and sentiment analysis of newspaper headlines could be used to efficiently characterize the social and political climate in Bangladesh during both election season and the extent of the country's obsession with cricket. The authors utilized three different techniques to visualize their data, namely dendrogram clustering, bar chart of emotions and word count for sentiment analysis.

Tusar and Islam (2021) conducted a comparative study of sentiment analysis using NLP and various machine learning techniques (including SVM, logistic regression, multinomial naïve bayes) on US Airline Twitter data. The authors concluded that SVM and Logistic regression used with a BoW technique delivered the best results with 77% accuracy.

Liang, Liu & Zhang (2020) explored the use of Doc2Vec combined with SVM for sentiment classification of Mandarin language character micro-blogs with an accuracy of 92.87%. In addition to standard pre-processing techniques (including removal of stop words), the authors also removed certain parts of speech that were considered "noisy", as part of the data purification step.

Ligthart, Catal & Tekinerdogan (2021) argue that it is insufficient to approach sentiment analysis as a process of categorization or classification, but rather a holistic approach should be adopted that considers syntactics, semantics and pragmatics. The syntactics layer includes tasks such as microtext normalization and part of speech (POS) tagging, while the semantics layer involves tasks such as word sense disambiguation and concept extraction. The pragmatics layer includes tasks such as sarcasm detection and aspect extraction.

Rozado, Hughes & Halberstadt (2022) utilized transformer language models to conduct a longitudinal analysis of sentiment and emotion in news media headlines. The authors found that over the research interval (2000-2019) that on average right-leaning outlets tended to be consistently more negative than left-leaning news. By contrast, Alonso et al. (2021) utilized sentiment analysis techniques that focused on "the polarity and strength of sentiments expressed in a text" as a method of classifying whether news was "fake news." The ability to detect fake news is important to upholding democracy and unfortunately the prevalence of "fake news" has been shown to have a very real and tangible impact on voter perception and consequently elections (Grossman & Helpman 2023).

III. Data

A. Data Selection

The corpus of data studied in this paper is comprised of forty-four political coverage news articles from online sources. Given Rozado, Hughes & Halberstadt's (2022) longitudinal study observations, we sought a balanced corpus of data by sourcing news articles from both left-leaning and right-leaning publications, such that 50% of the articles were sourced from left-leaning news outlets (e.g. The Guardian) with the remaining 50% sourced from right-leaning news outlets (e.g. Fox News). The 2023 AllSides Media Bias Ratings Chart, which categorizes bias of news outlets based on online U.S. political content, was used to identify left-leaning and right-leaning news outlets for this study (see figure 1).

Fig. 1: AllSides Media Bias Chart. Source: Com Library 2023

The topics for the forty-four articles used in this study were selected to cover the seven priority areas identified by the Biden-Harris Administration, namely: (i) covid-19; (ii) climate; (iii) racial equity; (iv) economy; (v) health care; (vi) immigration; and (vii) restoring America's global standing (The White House 2022). Each of the articles in the corpus was published in either 2022 or 2023. Table 1 sets out the corpus topics and corresponding sub-topics selected for this study. While racial equity is not specifically noted in the table below, racial equity is covered in several sub-topics including improving access to health care and education.

Topic	Sub-topic
(i) Economy	job creation taxes improving economic growth anti-trust
(ii) National Security	foreign policy national defense anti-terrorism
(iii) Health Care	improving access affordability WHO Pandemic Treaty Medicare
(iv) Social Issues	abortion LGBTQ+ rights gun control immigration
(v) Climate	Paris climate agreement 2050 net-zero emissions clean energy and infrastructure virtual climate summit
(vi) Education	improving access reducing student debt increasing funding for schools

Table 1: Corpus topics and sub-topics

B. Data Labelling

Each article was read, assessed and manually tagged with the appropriate sentiment. These tags were used as the target variable labels during the train and test phase of the model build. This supervised learning method relies on pre-labelled data with the correct labels. This informed the model to yield accurate labeling results when presented with never-before-seen data.

C. Data Pre-processing

The corpus of news articles was pre-processed and converted to vectors of numbers as machine learning algorithms cannot process text data. Figure 2 below outlines the pre-processing and vectorization steps we used to prepare the data before applying a machine

learning classification algorithm. The pre-processing and vectorization steps are collectively referred to as NLP. Further details on the pre-processing steps are set out in Appendix 1.

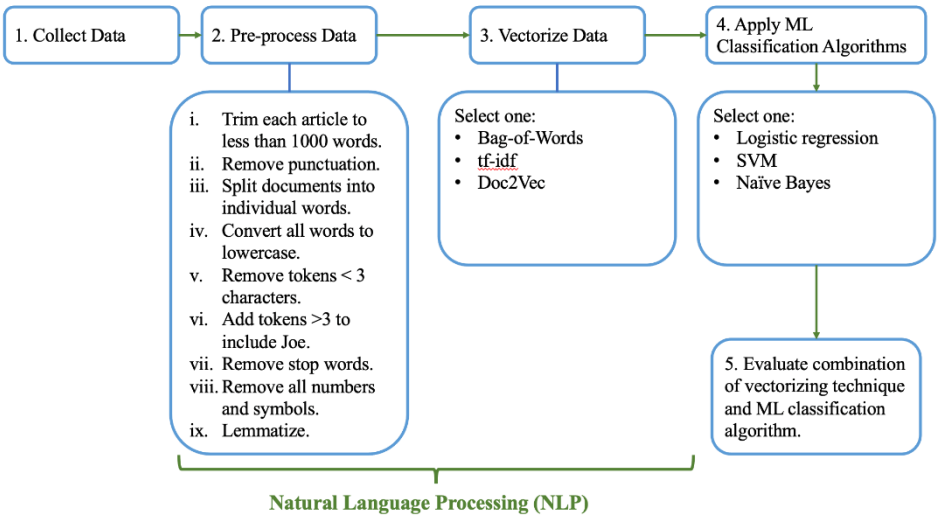


Fig 2. Research process from data collection to model evaluation

An exploratory data analysis (EDA) was also undertaken to better understand the corpus of data. Figure 2 illustrates the number of articles with negative or positive sentiment (as labelled by the authors) by political leaning. The finding was that ~70% of all articles were negative vs ~30% positive. Notably, there were no positive articles from the right-leaning news outlets which is understandable given that the coverage was focused on the Biden Administration priorities.

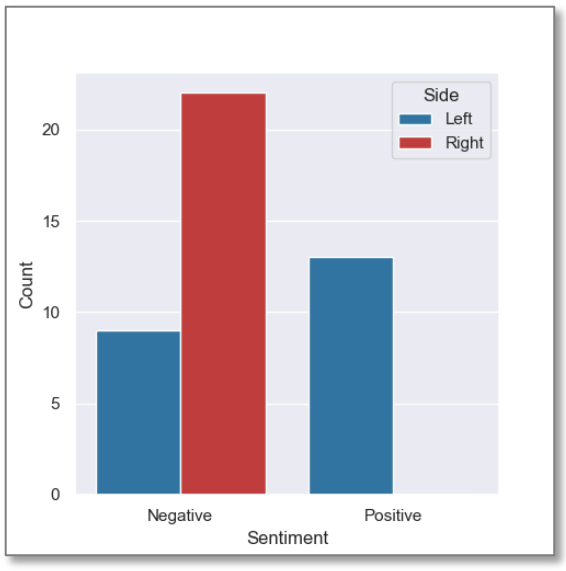


Fig 3. Number of articles with negative or positive sentiment by political leaning.

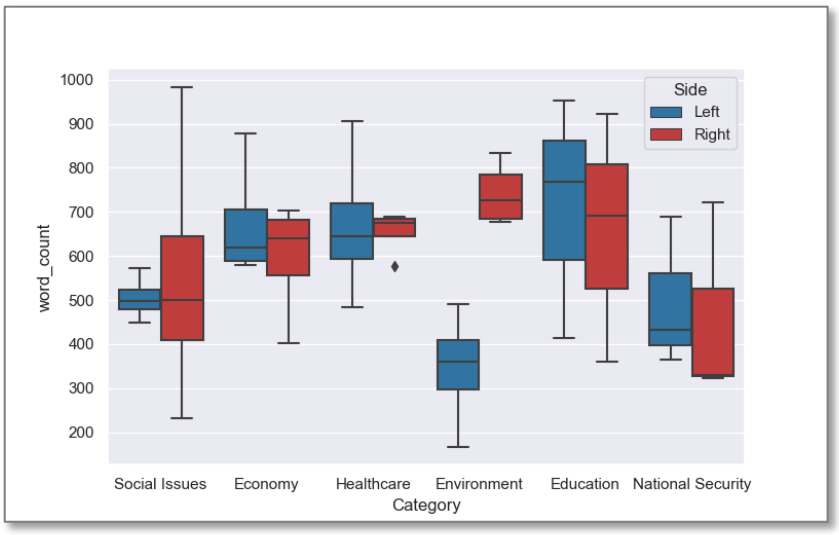


Fig 4. Corpus word count boxplot by topic and political leaning.

IV. Methods

A. Topic Clustering: K-means

K-means, an unsupervised learning algorithm, was applied to the total number of news articles (n=44) and organized into k clusters (k=11). K-means clustering was used to help group similar news articles together based on their content. Using n-grams (e.g., "job creation" or "worse than expected") enhanced the ability to identify like-phrases to cluster similar documents together. The method of optimizing a logical number of clustered articles was undertaken through trial and error, informed by the ontology, and selecting different k values to arrive at k=11.

B. Topic Modeling: LDA

LDA is an unsupervised method of topic modeling that can assist with discovering hidden themes in a collection of documents or, in this case, a collection of digital news articles on political topics. These "hidden" insights can be particularly important to a Presidential election campaign team looking to ensure it stays ahead of trending issues.

In this study we built an LDA model using Gensim, optimized the LDA hyperparameters through calculating the parameter grid coherence values and then leveraged pyLDAvis to visualize the output. Table 3 sets out the top ten results of our LDA hyperparameter tuning. Interestingly, the highest coherence value (0.40) recommended that our model be built with six topics with $\alpha = 0.01$ and $\beta = 0.01$. These results made intuitive sense, particularly when cross-referenced against our manually created ontology of six discrete topics (see figure 6).

1 to 10 of 300 entries Filter

Validation_Set	Topics	Alpha	Beta	Coherence
75% Corpus	6	0.01	0.01	0.4004482788079467
75% Corpus	6	0.01	0.31	0.380615187835458
75% Corpus	6	0.01	0.61	0.3849022943643719
75% Corpus	6	0.01	0.9099999999999999	0.3881191044988483
75% Corpus	6	0.01	symmetric	0.38244847692606615
75% Corpus	6	0.31	0.01	0.38341764050976623
75% Corpus	6	0.31	0.31	0.3762931115776853
75% Corpus	6	0.31	0.61	0.3726826956455958
75% Corpus	6	0.31	0.9099999999999999	0.3926139653178052
75% Corpus	6	0.31	symmetric	0.36580665147838126

Table 3. Gensim coherence table

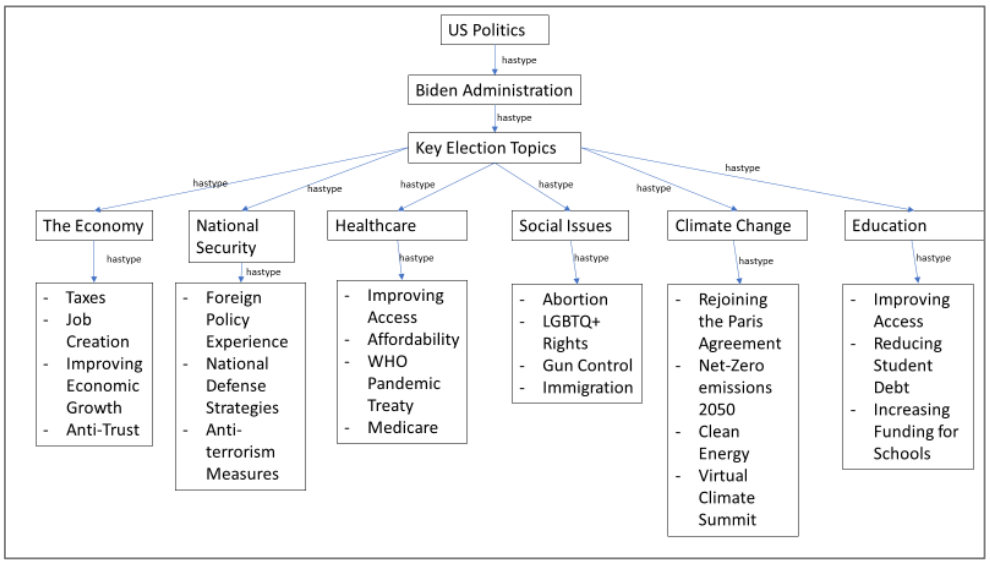


Fig 6. Manually Created Ontology

C. Semantic Analysis

(i) Vectorization

Text data cannot be analyzed via machine learning algorithms until such data is converted into numerical data through the process of vectorization. Converting to numerical data helps derive distinct features out of the text that the model can train on. In this study, the effectiveness of three different vectorization techniques were studied, namely BoW, TF-IDF and Doc2Vec.

Both TF-IDF and Bag of Words (BoW) are both effective methods of extracting features and determining frequency of terms. Although both algorithms do have their drawbacks, BoW is simple to use and creates vectors containing the counts of words, while TF-IDF determines how frequently the word appears in text and the importance. Each of these methods were chosen to compare and contrast the differences in effectiveness of each technique.

(a) BoW

BoW is a relatively straightforward vectorization technique that creates finite length vectors by counting the total occurrences of the most frequently used words (D'Souza 2018). The top 10 words by number of times the word appeared in our corpus were as follows: 'include' (53); 'health' (56); 'policy' (57); 'climate' (59); 'state' (61); 'make' (62); 'also' (69); 'year' (73); 'plan' (88) and 'say' (96).

(b) TF-IDF

TF-IDF is a vectorization approach that allows us to rescale the frequency of words in a document against their frequency across the entire corpus. This method assigns higher weights to words that appear less frequently in the corpus and lower weights to words that appear frequently. The TF component of the TF-IDF formula calculates the frequency of a word within a document, while the IDF component determines how rare a word is across the entire corpus. Multiplying these two values together provides a measure of the importance of a word in a document relative to its frequency in the corpus. Common words are penalized while unique words are rewarded. The mathematical formula for TF-IDF is as follows (Wikipedia 2023a):

Term Frequency (TF)

$$\mathrm{tf}(t,d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}}$$

Inverse Document Frequency (IDF)

$$\text{idf}(t, D) = \log \frac{N}{|\{d \in D : t \in d\}|}$$

This makes intuitive sense as it conveys the importance of a word with how often it is used within a document and how often it is not used across the entire body of the corpus. This relationship provides the basis for the importance or relevancy of a particular word or set of words in a document.

(c) Doc2Vec

Doc2Vec was implemented using Python and Gensim packages. Doc2Vec creates a vectorized representation of a group of words irrespective of length. The Doc2Vec output was visualized using the t-sne algorithm which focuses on local clusters. The vector length was set to 100 for each document. This provided 100 variables (dimensions) to cross-compare documents. An epoch size of 256, representing the number of training passes over the corpus, allowed for Doc2Vec to adequately learn on the small corpus that was used.

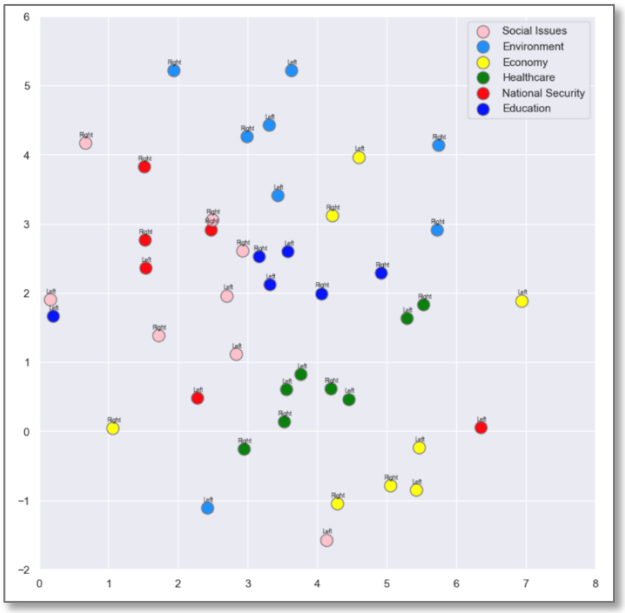


Fig 5. Doc2Vec visualization by topic.

(ii) Machine Learning Algorithms

Logistic regression, SVM and naïve bayes machine learning algorithms were respectively applied in combination with each of the different methods of vectorization, namely BoW, TF-IDF and Doc2Vec, to create 9 different models.

Logistic Regression is a machine learning model that is used for classification and predictive analytics. Logistic regression estimates the probability of an event occurring, using regression algorithms. SVM is a powerful machine learning model, capable of performing linear or nonlinear classification, regression, and even outlier detection. SVM operates by finding a hyperplane that aids in distinctively classifying various data points. Naïve Bayes in a machine learning model classification model based on Bayes Theorem; a statistical method using a priori information to inform probability.

(iii) Applying Sentiment Analysis to a Single Topic

Having observed the accuracy of the application of BoW combined with SVM to the entire corpus, we also explored whether it would be possible to run similar experiments against a subset of the corpus by topic. Unfortunately, the results were unreliable due to the small size of the corpus within a given topic. This would approach be interesting to explore with a larger corpus.

V. Results

A. Topic Clustering: K-Means

The results for the K-means clustering are set out in Table 4. While clusters tended to combine articles with similar topics, there were times where the article content was truly unique and was clustered by itself (e.g. cluster 4). This presented a challenge while trying to optimize the clustering method as there were articles that did not fit easily into another cluster. Using our

manually created ontology (figure 6), as a guide to how we ideally wanted the articles to cluster, the clustering method was refined through hyperparameter tuning resulting in k=11. This provided clusters that tended to combine like articles. Typically, when tuning for values of *k* being less than eleven, there were one or two main clusters that included most articles; providing little value for topic clustering. In addition to this, when tuning for values of *k* being greater than eleven, most clusters were composed of no more than one article.

It is also important to note that a lot of these clusters tended to combine topics which represent the nature of presidential issues. Issues are often interrelated and the contrast between them can be ambiguous (e.g. health care and social issues).

Cluster 1: Category: Social Issues, Side: Right, Sentiment: Negative Category: National Security, Side: Left, Sentiment: Negative Category: Healthcare, Side: Right, Sentiment: Negative	**Cluster 2:** Category: Economy, Side: Left, Sentiment: Positive Category: Economy, Side: Left, Sentiment: Positive Category: Economy, Side: Right, Sentiment: Negative Category: Education, Side: Left, Sentiment: Negative Category: Education, Side: Left, Sentiment: Positive Category: Education, Side: Right, Sentiment: Negative Category: Education, Side: Right, Sentiment: Negative Category: Education, Side: Right, Sentiment: Negative Category: Social Issues, Side: Right, Sentiment: Negative Category: Social Issues, Side: Right, Sentiment: Negative
Cluster 3: Category: Healthcare, Side: Right, Sentiment: Negative Category: Healthcare, Side: Left, Sentiment: Positive Category: Environment, Side: Left, Sentiment: Positive	**Cluster 4:** Category: Healthcare, Side: Right, Sentiment: Negative
Cluster 5: Category: Environment, Side: Right, Sentiment: Negative	**Cluster 6:** Category: Social Issues, Side: Left, Sentiment: Negative Category: Social Issues, Side: Left, Sentiment: Negative Category: Social Issues, Side: Left, Sentiment: Positive Category: Healthcare, Side: Right, Sentiment: Negative Category: Social Issues, Side: Left, Sentiment: Positive
Cluster 7: Category: Environment, Side: Right, Sentiment: Negative Category: Environment, Side: Right, Sentiment: Negative Category: Environment, Side: Left, Sentiment: Positive Category: Environment, Side: Left, Sentiment: Positive Category: Environment, Side: Right, Sentiment: Negative	**Cluster 8:** Category: Healthcare, Side: Left, Sentiment: Negative
Cluster 9: Category: Education, Side: Left, Sentiment: Negative Category: National Security, Side: Right, Sentiment: Negative Category: Economy, Side: Left, Sentiment: Negative Category: National Security, Side: Left, Sentiment: Positive Category: National Security, Side: Left, Sentiment: Positive Category: National Security, Side: Right, Sentiment: Negative	**Cluster 10:** Category: Economy, Side: Right, Sentiment: Negative Category: Economy, Side: Left, Sentiment: Positive Category: Healthcare, Side: Left, Sentiment: Negative Category: Healthcare, Side: Left, Sentiment: Positive Category: National Security, Side: Right, Sentiment: Negative Category: Economy, Side: Right, Sentiment: Negative Category: Social Issues, Side: Right, Sentiment: Negative Category: Environment, Side: Left, Sentiment: Negative
Cluster 11: Category: Economy, Side: Right, Sentiment: Negative	

Table 4. Articles Clustered by Topic (k=11).

B. Topic Modeling: LDA

LDA results were visualized using pyLDAvis. Figure 7 depicts the pyLDAvis topic modeling visualization for the corpus of pre-processed digital news articles and displays the most relevant terms for topic (one of six) in the bar chart.

There are two components to the visualization: the topic bubbles and the horizontal bar graph. The topic bubbles represent the distribution of the various topics in 2-dimensional space. The larger the topic bubble, the more frequently that topic appears in the documents. The distance between the bubbles is an indication of the semantic relationship between the bubbles. On the right-side of figure 7, the bar chart illustrates the frequency distribution of the words in the documents in blue and the frequency of each word in a particular topic in red (Kohli 2019).

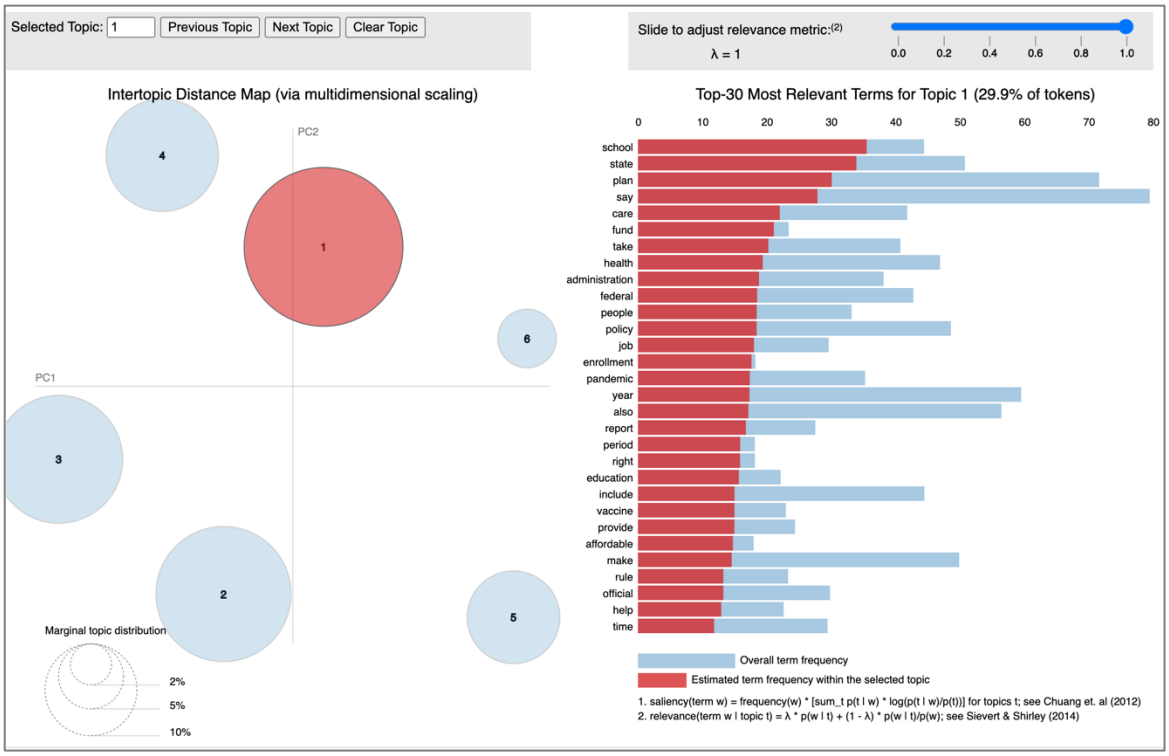


Fig 7. Topic modeling with pyLDAvis.

C. Models: Vectorization and Machine Learning Classification Algorithms

This study is a comparative review of the effectiveness of different combinations of vectorizing and machine learning algorithms for semantic classification. Table 5 sets out the results for each combination. As demonstrated by Mikolov et al. (2013), the results illustrate that simple model architectures, such as BoW, can be used to train high quality word vectors and consequently have a direct impact on the effectiveness of the machine learning algorithm.

	BoW				**TF-IDF**				**Doc2Vec**			
Algorithm	**Accuracy**	**Precision**	**Recall**	**F1 Score**	**Accuracy**	**Precision**	**Recall**	**F1 Score**	**Accuracy**	**Precision**	**Recall**	**F1 Score**
Logistic Regression	0.79	0.84	0.79	0.74	**0.71**	0.51	0.71	0.60	0.43	0.55	0.43	0.45
Naïve Bayes	**0.86**	0.86	0.86	0.86	0.60	0.36	0.60	0.45	**0.80**	0.85	0.80	0.78
Support Vector Machine (SVM)	**0.86**	0.88	0.86	0.84	**0.71**	0.51	0.71	0.60	0.57	0.70	0.59	0.60

Table 5: Results: Vectorizing Models and Machine Learning Algorithms.

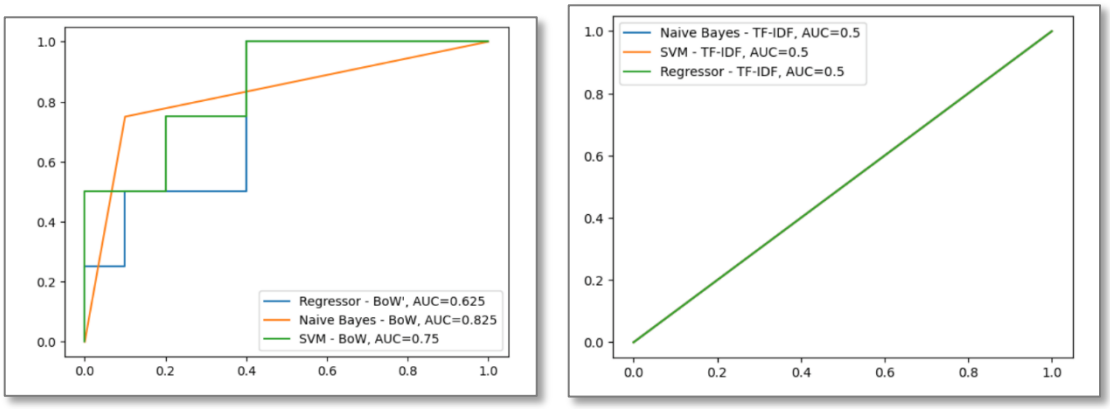
Naïve Bayes and SVM worked very well with BoW, while logistic regression did not. Overall, BoW based models were our best performing models. BoW classifies documents by converting words into tokens by giving each word an integer id, separated by white space and/or punctuation, and then counting the occurrence of each token in each document. The higher quantity of unique tokens per topic may have assisted BoW to perform better than the rest of the methods. Furthermore, the BoW method is relatively simple and not computationally demanding. The small size and simplicity of our corpus may have been why BoW performed better than the rest of the methods. BoW is dependent on the unique number of tokens and typically does not do well with larger corpuses and larger vocabulary size.

In contrast, TF-IDF quantifies words by how uniquely important they are in a specific document within a corpus/collection of documents. This is done by finding the term frequency

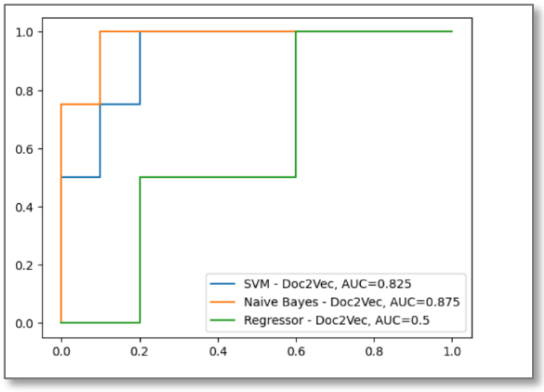
in a document divided by how many other documents that word is present in. This means that words that do not appear in document 1 but appear in multiple other documents will receive a low value of importance in document 1. This also means that words that appear in multiple documents will receive a low importance score. For instance, if the word "education" appears many times in one document, and then appears a very few times in other documents, and none in other documents, the importance score can be very low. This would make it particularly difficult to utilize TF-IDF efficiently with our corpus, as we have six major topics and three to four subtopics per major topic, resulting in many key words receiving far lower importance scores than they should have.

Doc2Vec performed the worst out of all three models. This is likely due to Doc2Vec being very dependent on the training phase of the model. During testing Doc2Vec aims to predict what a document's contents are composed of based on the title of the article in test and the results of what similarly titled documents produced during training. Since our corpus was very small, this was not an ideal dataset for Doc2Vec. The nature of the small sized corpus made it difficult to provide sufficient data for the Doc2Vec model to train on and then predict what articles were composed of during the testing phase.

Figures 8(a)-(c) illustrate the Receiver Operating Characteristic curve (ROC curve) for each of the different model combinations. The ROC curve is an important visualization that illustrates "the diagnostic ability of a binary classification system as its discrimination threshold is varied" (Wikipedia, 2023b).



(a) (b)



(c)

Fig 8: ROC curves: (a) Bag of Words; (b) TF-IDF; (c) Doc2Vec

VI. Analysis and Interpretation

One of the challenges raised by this corpus of data, and any dataset of news articles that address contemporary political commentary, is the way themes and topics interconnect or overlap. In particular, the LDA visualization provided insights into topic groupings that were not necessarily evident simply from reading the articles. By way of example, figure 7 illustrates the thirty most relevant terms for Topic 1. Topic modeling provides interesting insight into non-

obvious connections that a Presidential election campaign team may want to dig into deeper. For example, while one of the seven priority areas identified by the Biden-Harris Administration is Covid-19, it may pay to delve deeper into the sentiment of news coverage at the intersection of the pandemic and education.

The second key observation is the importance of selecting the right combination of vectorization method and ML algorithm to optimize accuracy of the model. Of the nine experiments that were run, Naïve Bayes and SVM utilizing BoW technique delivered the highest accuracy at 86%. The highest precision model for BoW was SVM at 88%. SVM handles high-dimensional spaces and non-linear boundaries, which makes it useful for NLP when the features, such as the number of words, are large. SVM tends to be more accurate than Naïve Bayes but can be slower on larger datasets. By contrast, Naive Bayes can be used for NLP applications on smaller datasets if the assumptions of independence hold true. The model essentially assumes that the presence of one feature doesn't affect the presence of another. By contrast, the combination with the lowest accuracy was Doc2Vec with logistic regression with 43%. Logistic regression performed better in combination with BoW, delivering a higher accuracy of 79%.

VII. Conclusions

Given our results one can deduce that the BoW NLP method with either a Naive Bayes and/or a Support Vector Machine (SVM) classification algorithm would be best when conducting sentimental analysis on political news. That said, given our small corpus, these results can potentially be skewed. Although BoW has largely outperformed TF-IDF and Doc2Vec for our corpus, we may find the results may be different over a larger corpus. To give an absolute conclusion it will be best to carry out this project with some of the methodologies discussed in our future work, such as web scraping, VADER, and transformers. This will better

test our theory and methodologies on a larger scale and help give us a more definitive answer on what would be the most optimal and highest performing NLP methodology and classification algorithm model combination.

VIII. Directions for Future Work

The most significant issue we encountered throughout our research was the size of our corpus. With a total of forty-four articles, it made it more difficult to train models and obtain higher accuracy and precision. While a smaller corpus worked for simpler methods such as BoW, we obtained poorer results with models such as Doc2Vec.

One method of tackling our corpus size problem is to enlarge the corpus via web scraping news outlet websites by topic. A few tools that may be particularly useful for expanding our corpus are the beautifulsoup and NewsCatcher packages in python. Both packages can help expand our existing corpus from a few dozen articles to a few thousand articles in no time.

Another way to tackle our issue of not having enough data points in our small corpus is through transformers. Transformers can be used to augment a small dataset into a much larger dataset by creating proxy data through paraphrasing and synonym matching.

In addition to increasing our dataset size through augmenting our existing corpus, we could also look at different approaches in the type of emotions we are trying to draw from our data in our NLP models. Sentiment analysis only looks at pleasant and unpleasant emotions. While emotion detection covers the whole range of emotions in the Circumplex Model; pleasant, unpleasant, activation, and deactivation (see figure 9 for Circumplex Model). This covers a much wider range of human emotions and perhaps can assist our models yield better results than when just looking for positive or negative results in sentiment analysis.

An alternative solution would be to divert from a machine learning model and move towards a Lexicon-based method. Lexicon methods are simple rule-based approaches that tokenize data. This is a simpler model that already defines keywords to positive, neutral, and negative emotions. One specific Lexicon method is Valence Aware Dictionary and Sentiment Reasoner, or VADER for short. VADER was specifically created for sentiment analysis on social media. Arslan et al. (2017) demonstrated promising results with their Lexicon-based sentiment analysis on a small corpus compiled of real-time data from twitter. Although news outlets are not exactly comparable to social media content, it would be worth exploring the application of that method to news coverage.

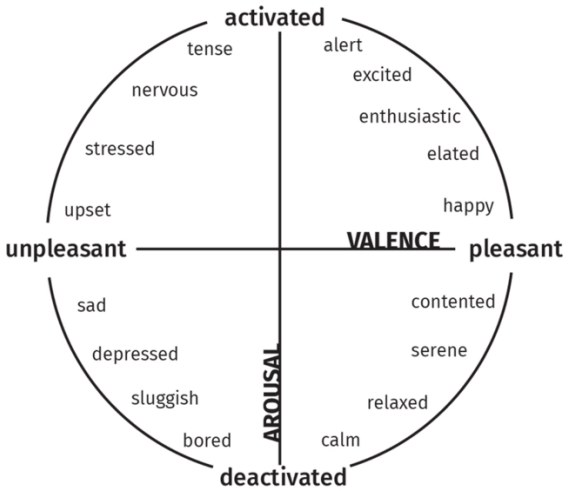


Fig 9: Circumplex Model. Source: Wan, Xiaoqing. 2021. "NLP Emotion Detection in Python: Compare Lexicon-Based and Deep Learning Methods with Tutorial." *Decision Analytics, YouTube.* September 11, 2021.

Acknowledgements: We would like to express our sincere gratitude and appreciation to Dr A.J Maren for her support and constructive feedback on the initial draft of this manuscript. Further, we would like to thank Paul Huynh for the pre-processing corpus code and initial clustering code, as well as our peers who undertook the Natural Language Processing course at Northwestern University and provided interesting commentary and insights which we incorporated into the manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

Alonso, Miguel A., David Vilares, Carlos Gómez-Rodríguez, and Jesús Vilares. 2021. "Sentiment Analysis for Fake News Detection." *Electronics (Basel)* 10, no. 11 (2021), 1348: 1-32. https://doi.org/10.3390/elcectronics10111348.

Arslan, Yusuf, Aysenur Birturk, Bekjan Djumabaev and Dilek Küçük. 2017. "Real-time Lexicon-based sentiment analysis experiments on Twitter with a mild (more information, less data) approach," *2017 IEEE International Conference on Big Data (Big Data)*, Boston, MA, USA, 2017: 1892-1897. https://doi.org/10.1109/bigdata.2017.8258134.

Blei, DM, AY Ng, and MI Jordan. 2003. "Latent Dirichlet Allocation." *Journal of Machine Learning Research* 3, no. 4-5 (2003): 993–1022. https://doi.org/10.1162/jmlr.2003.3.4-5.993.

Brownlee, J. 2017. *Deep Learning for Natural Language Processing: Develop Deep Learning Models for Natural Language in Python.* Self-published. Machine Learning Mastery, 2017.

Church, Kenneth Ward. 2017. "Word2Vec." *Natural Language Engineering* 23, no. 1 (2017): 155–62. https://doi.org/10.1017/S1351324916000334.

Com Library. 2023. "Media Bias: Which Way Does Your News Lean?" *Com Library.* Last modified May 17, 2023 2:30pm. https://libguides.com.edu/c.php?g=649909&p=4556556.

D'Souza, Jocelyn. 2018. "An Introduction to Bag-of-Words in NLP." *Medium,* April 3, 2018. https://medium.com/greyatom/an-introduction-to-bag-of-words-in-nlp-ac967d43b428.

Edwards, Kinga. 2021. "How To Do Social Media Sentiment Analysis in Politics." *Determ.* Accessed on April 22, 2023. https://www.determ.com/blog/social-media-sentiment-analysis-in-politics/.

Edwards, King. 2021. "Media Monitoring: The Ultimate Guide." *Determ.* Accessed on April 22, 2023. https://www.determ.com/blog/media-monitoring-ultimate-guide/.

Farkhod, A.; Abdusalomov, A.; Makhmudov, F.; Cho, Y.I. 2021. "LDA-Based Topic Modeling Sentiment Analysis Using Topic/Document/Sentence (TDS) Mol." *Applied Sciences.* 2021, 11, 11091. https://doi.org/10.3390/app112311091.

Galgoczy, Michael C, Atharva Phatak, Danielle Vinson, Vijay K Mago, and Philippe J Giabbanelli. 2022. "(Re)shaping Online Narratives: When Bots Promote the Message of President Trump During His First Impeachment." *PeerJ. Computer Science* 8 (2022): e947–e947. https://doi.org/10.7717/peerj-cs.947.

Grossman, Gene M., and Elhanan Helpman. 2023. "Electoral Competition with Fake News." *European Journal of Political Economy* 77 (2023), 102315: 1-12. https://doi.org/10.1016/j.ejpoleco.2022.102315.

Havrlant, Lukáš, and Vladik Kreinovich. 2017 "A Simple Probabilistic Explanation of Term Frequency-Inverse Document Frequency (tf-Idf) Heuristic (and Variations Motivated by This Explanation)." *International Journal of General Systems* 46, no. 1 (2017): 27–36. https://doi.org/10.1080/03081079.2017.1291635.

Hossain, Arafat, Md Karimuzzaman, Md. Moyazzem Hossain, and Azizur Rahman. 2021. "Text Mining and Sentiment Analysis of Newspaper Headlines." *Information (Basel)* 12, no. 10 (2021): 414–. https://doi.org/10.3390/info12100414.

Huang, C. C., Yang, C. C., & Lee, C. C. 2013. "Mining opinion polarity in political news coverage: A multi-level approach to analyzing news articles". *Government Information Quarterly*, 30(4), 373-382.

Hutto, C., and Eric Gilbert. 2014. "VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text." *Proceedings of the International AAAI Conference on Web and Social Media* 8, no. 1 (2014): 216–25. https://doi.org/10.1609/icwsm.v8i1.14550.

Huynh, Paul. 2019. "Text Processing v7_2019-03-03.Py," March 3, 2019.

Jelodar, Hamed, Yongli Wang, Chi Yuan, Xia Feng, Xiahui Jiang, Yanchao Li, and Liang Zhao. 2019. "Latent Dirichlet Allocation (LDA) and Topic Modeling: Models, Applications, a Survey." *Multimedia Tools and Applications* 78, no. 11 (2019): 15169–211. https://doi.org/10.1007/s11042-018-6894-4.

Khder, Moaiad. 2021. "Web Scraping or Web Crawling: State of Art, Techniques, Approaches and Application." *International Journal of Advances in Soft Computing and its Applications* 13, no. 3 (November 28, 2021): 145–68. https://doi.org/10.15849/ijasca.211128.11.

Kohli, Pahul Preet Singh. 2019. "Interpreting Topic Model Visualization – LDAvis Package." *Pahul Preet Singh Kohli Blog,* August 13, 2019. https://pahulpreet86.github.io/interpreting-topic-model-visualization-ldavis-package/.

Lekhtman, Alon. 2021. "When Logistic Regression Simply Doesn't Work." *Towards Data Science.* Accessed on June 4, 2023. https://towardsdatascience.com/when-logistic-regression-simply-doesnt-work-8cd8f2f9d997#:~:text=The%20reason%20is%20that%20the,even%20on%20the%20training%20data).

Liang, Yinghong, Haitao Liu, and Su Zhang. 2020. "Micro-Blog Sentiment Classification Using Doc2vec + SVM Model with Data Purification." *Journal of Engineering (Stevenage, England)* 2020, no. 13 (2020): 407–10. https://doi.org/10.1049/joe.2019.1159.

Ligthart, Alexander, Cagatay Catal, and Bedir Tekinerdogan. "Systematic Reviews in Sentiment Analysis: a Tertiary Study." *The Artificial Intelligence Review* 54, no. 7 (2021): 4997–5053. https://doi.org/10.1007/s10462-021-09973-3.

Mahar, Klara. 2021. "Improve Your Political Campaign With Media Monitoring." *Determ*. Accessed on April 22, 2023. https://www.determ.com/blog/improve-your-political-campaign-with-media-monitoring/.

Mazzoni, Claudio. 2021. "Augment Your Small Dataset Using Transformers and Synonym Replacement for Sentiment Analysis- Part..." *Medium*, May 22, 2021. Accessed June 3, 2023. https://towardsdatascience.com/augment-your-small-dataset-using-transformers-synonym-replacement-for-sentiment-analysis-part-1-87a838cd0baa.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. 2013. "Efficient Estimation of Word Representations in Vector Space" *ArXiv*. https://arxiv.org/pdf/1301.3781.pdf

Nandwani, Pansy, and Rupali Verma. 2021. "A Review on Sentiment Analysis and Emotion Detection from Text." *Social Network Analysis and Mining* 11, no. 1, November 28, 2021: 1–19. https://doi.org/10.1007/s13278-021-00776-6.

Nayak, Manish. 2019. "An Intuitive Introduction to Document Vector(Doc2vec)." *Medium*. June 24, 2019. Accessed June 4, 2023. https://pub.towardsai.net/an-intuitive-introduction-of-document-vector-doc2vec-42c6205ca5a2.

Pang, B., & Lee, L. 2008. "Opinion mining and sentiment analysis". *Foundations and Trends in Information Retrieval*, 2(1-2): 1-135.

Prabhakaran, Selva. 2018. "LDA in Python – How to grid search best topic models?" *Machine Learning Plus.* Accessed June 4, 2023. https://www.machinelearningplus.com/nlp/topic-modeling-python-sklearn-examples/.

Python Package Index. 2022. *newscatcherapi* V. 0.7.2. https://pypi.org/project/newscatcherapi/

Roul, Abhinandan. 2021. "Sentiment Analysis- Lexicon Models vs Machine Learning." *Medium,* February 18, 2021. Accessed June 4, 2023. https://medium.com/nerd-for-tech/sentiment-analysis-lexicon-models-vs-machine-learning-b6e3af8fe746.

Rozado, David, Ruth Hughes, and Jamin Halberstadt. 2022. "Longitudinal Analysis of Sentiment and Emotion in News Media Headlines Using Automated Labelling with Transformer Language Models." *PloS One* 17, no. 10 (2022): e0276367–e0276367. https://doi.org/10.1371/journal.pone.0276367.

Sabbagh, Ramin, and Farhad Ameri. 2020. "A Framework Based on K-Means Clustering and Topic Modeling for Analyzing Unstructured Manufacturing Capability Data." *Journal of*

Computing and Information Science in Engineering 20, no. 1 (2020). https://doi.org/10.1115/1.4044506.

Saeed, Mehreen. 2022. "A Guide to Text Preprocessing Techniques for NLP - Blog." *Scale Virtual Events*, June 28, 2022. Accessed on April 21, 2023. https://exchange.scale.com/public/blogs/preprocessing-techniques-in-nlp-a-guide.

Scott, Jake. 2021. "What's in a Word?" *Medium*, March 29, 2021. Accessed, June 3 2023. https://towardsdatascience.com/whats-in-a-word-da7373a8ccb.

The White House. 2022. "The Biden-Harris Administration Immediate Priorities" https://www.whitehouse.gov/priorities/.

Tusar, Md. Taufiqul Haque Khan, and Md. Touhidul Islam. 2021. "A Comparative Study of Sentiment Analysis Using NLP and Different Machine Learning Techniques on US Airline Twitter Data." *Presented at the Proceeding of the International Conference on Electronics, Communications and Information Technology (ICECIT)* https://doi.org/10.48550/arxiv.2110.00859.

Uzila, Albers. 2022. "All You Need to Know about Bag of Words and Word2vec - Text Feature Extraction." *Medium*, November 26, 2022. Accessed June 4, 2023. https://towardsdatascience.com/all-you-need-to-know-about-bag-of-words-and-word2vec-text-feature-extraction-e386d9ed84aa#0e83.

Verma, Y. 2022. "A complete tutorial on zero-shot text classification". *Analytics India Magazine*. Accessed on April 20, 2023, from https://analyticsindiamag.com/a-complete-tutorial-on-zero-shot-text-classification/.

Wan, Xiaoqing. 2021. "NLP Emotion Detection in Python: Compare Lexicon-Based and Deep Learning Methods with Tutorial." *Decision Analytics.* YouTube. September 11, 2021.

Wikipedia. 2023a. "tf-idf". *Wikipedia Foundation*. Last modified March 6, 2023, 07:26 UTC. https://en.wikipedia.org/wiki/Tf–idf.

Wikipedia. 2023b. "Receiver Operating Characteristic". *Wikipedia Foundation*. Last modified May 20, 2023, 05:47 UTC. https://en.wikipedia.org/wiki/Receiver_operating_characteristic.

Zafra, Miguel Fernandez. 2020. "Web Scraping News Articles in Python." *Medium,* May 25, 2020. Accessed June 3, 2023. https://towardsdatascience.com/web-scraping-news-articles-in-python-9dd605799558.

Appendix

1. Data Pre-processing

Each of the 44 articles in the corpus was trimmed to a maximum of 1000 words and saved as individual files in docx format.

Given that the nature and extent of data pre-processing has a material impact on the efficacy of NLP models (Saeed 2022), the following pre-processing steps were utilized in this study:

(a) Tokenization of data

Tokenization of data is the process of converting sentences into tokens. These tokens become the building blocks for analysis. The tokens are created by splitting documents into individual words, removing all punctuation, removing all non-letters, removing all words shorter than three letters, and converting all words to lowercase such that the same words with different capitalizations will be read as one word by the algorithm.

(b) Removing stop words

Removing stop words is a standard step within preprocessing data in NLP. "Stop Words" are redundant words that are commonly used as filler words. Stop word examples include: are, and, the, is, are, and of. The NLTK stop words library was utilized to remove stop words.

(c) Data Stemming:

Stemming data is the process of converting a word into its stem word or base word. For example, the words stemming, stems, and stemmed all simply become just stem. Stemming is utilized to identify and create classifications of words within the dataset. The NLTK porter stemmer function was utilized to effect data stemming.