Balancing Transparency and Accuracy: A Comparative Analysis of Rule-Based and Deep Learning Models in Political Bias Classification

Manuel Nunez Martinez, Sonja Schmer-Galunder, Zoey Liu,
Sangpil Youm, Chathuri Jayaweera, Bonnie J. Dorr
University of Florida, FL,USA
{manuel.nunez, s.schmergalunder, liu.ying, youms,
chathuri.jayawee, bonniejdorr}@ufl.edu

Abstract

The unchecked spread of digital information, combined with increasing political polarization and the tendency of individuals to isolate themselves from opposing political viewpoints, has driven researchers to develop systems for automatically detecting political bias in media. This trend has been further fueled by discussions on social media. We explore methods for categorizing bias in US news articles, comparing rule-based and deep learning approaches. The study highlights the sensitivity of modern self-learning systems to unconstrained data ingestion, while reconsidering the strengths of traditional rule-based systems. Applying both models to left-leaning (CNN) and right-leaning (FOX) news articles, we assess their effectiveness on data beyond the original training and test sets. This analysis highlights each model's accuracy, offers a framework for exploring deep-learning explainability, and sheds light on political bias in US news media. We contrast the opaque architecture of a deep learning model with the transparency of a linguistically informed rule-based model, showing that the rule-based model performs consistently across different data conditions and offers greater transparency, whereas the deep learning model is dependent on the training set and struggles with unseen data.

1 Introduction

The current political climate in the United States is characterized by intense polarization and an unprecedented ease of publishing and disseminating information, where partisan hostility and negative perceptions of opposing party members are at an all-time high (Doherty et al., 2023). This dynamic is further exacerbated by social media platforms, where users curate their news feeds in a way that reinforces existing biases and isolates them from diverse perspectives, stifling constructive dialogue and creating what researchers term "epistemic bubbles" (Kelly, 2021).

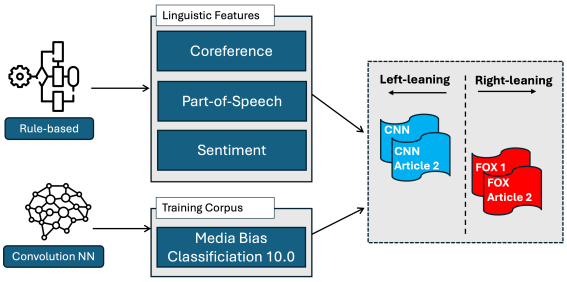

Figure 1: Comparison of Rule-based and Convolutional NN models: CNN and FOX news articles serve as external, unseen datasets for the Convolutional NN model. The rule-based model determines political bias using three linguistic features.

To address this, Natural Language Processing (NLP) researchers have developed models intended to automatically and objectively detect the presence and direction of bias. Examples include model architectures ranging from rule-based designs (Hube and Fetahu, 2018) to State of the Art (SoA) transformer architectures (Raza et al., 2024). While SoA architectures have been shown to distinguish biased narratives from neutral ones, they struggle to learn the nuanced nature of bias expression without a sufficiently large and comprehensive dataset.

Our contributions include an investigation of both a rule-based and a deep learning model for political bias classification as depicted in Figure 1, with the goal of promoting a more informed discussion on bias detection methodologies. To overcome data demands of SoA architectures, we adopt a convolutional neural network model.[1] Our contrasting approach is a simpler, more transparent rule-based model for bias classification using sentiment detection and linguistic features. This model does not rely on preexisting bias lexicons, *"black box"* machine learning models, or large training datasets. Moreover, its simplicity allows for easy correction, with a few, clearly delineated, components.

A second contribution is the use of linguistic in-

[1]For brevity, we use "convolutional NN model" henceforth, as the abbreviation CNN is employed to refer to a news outlet.

Proceedings of the Second Workshop on Social Influence in Conversations (SICon 2024), pages 102–115

formation for detecting an article's stance towards entities.[2] Our rule-based approach includes a novel part-of-speech driven "reference resolution" (e.g., associating adjectives with a corresponding noun), for a more focused stance assignment. We emphasize that it is not our goal to achieve SoA performance for political bias classification through the rule-based model, but rather to explore the extent to which straightforward linguistic features (parts of speech, coreference, and sentiment) can be leveraged to classify political bias.

A third contribution involves exploring methods to enhance **explainability** of deep learning models. By testing a convolutional NN model on various datasets and correlating its performance disparities with differences in the data, we identify the features prioritized by the model.

Our findings show that the rule-based model maintains consistent performance across various data conditions, presenting a clear right-leaning bias for FOX. By contrast, the convolutional NN model relies heavily on its training set, struggling with data not directly related to the political bias data on which it is trained. The rule-based approach performs comparably to deep learning in these situations, making it more applicable to real-world scenarios and offering greater transparency.

The next section reviews bias detection methodologies in news media. Section 3 covers data collection, preprocessing, and experimental setup. Section 4 details the implementation of rule-based and convolutional NN models. Section 5 evaluates model performance and their application to external data, with concluding remarks in Section 6.

2 Related Work

Following Mullainathan and Shleifer (2002), we view bias in news articles not as as a distortion or selective presentation of information to convey a belief, potentially impacting readers' opinions. Media bias is categorized into coverage bias, gatekeeping bias, and statement bias (Saez-Trumper et al., 2013). Our study focuses on statement bias, i.e., the use of rhetoric describing entities (Hamborg et al., 2019) identified by our rule-based sentiment analysis model through identification of words conveying sentiment toward entities.

Entity Level Semantic Analysis (ELSA) (Rønningstad et al., 2022), is exemplified by the work of Luo and Mu (2022), where sentiment toward an entity is computed across sentences, iteratively lowering the sentiment scores for entities appearing in negative contexts. Our current study adopts a form of ELSA that eliminates the need for "Negative Smoothing" by using part-of-speech (POS) resolution to identify sentiment towards a given entity, thus filtering out "noise" introduced by incidental occurrences of nearby negative terms. Deep-learning ELSA models often suffer from an opaque architecture and overly broad feature selection. Fu et al. (2022) address this with a *transparency layer* in a convolutional NN, that adjusts feature selection using an integrated gradient technique, aligning with the POS resolution method described here.

Bias detection in media is typically handled as binary or multi-class classification, mapping to political leanings using e.g., Support-Vector Machines, Logistic Regression, and Random Forest techniques (Rodrigo-Ginés et al., 2024) with hand-crafted feature extraction. Hube and Fetahu (2018) adopt a rule-based strategy, defining a list of inflammatory terms and expanding it with Word2Vec (Mikolov et al., 2013) from Conservapedia articles[3] to create a lexicon of politically charged words.

Our rule-based model differs by not relying on lists of predefined terms; instead, it assumes that differing stances towards an entity across articles indicate bias. This simpler approach hinges on stances towards notable entities, differing from the single-sentence approach of Hube and Fetahu (2018). Our model's theoretical foundation suggests that differences in stance expression between media outlets signal statement bias.

Bias detection research favors Transformers over Recurrent Neural Networks (RNNs), due to their self-attention mechanism for modeling sequential structures (Vaswani et al., 2017). However, their reliance on low-level lexical information (Rodrigo-Ginés et al., 2024), is often insufficient for political bias detection.[4] Chen et al. (2020) attempt to overcome hand-crafted feature extraction limitations, while avoiding deep learning's pitfalls, by analyzing second-order information, like the frequency and order of biased statements, and employing machine-learning methods for bias detection. Our hybrid approach aligns with this, but focuses

[2]We define *stance* as the overall attitude of a news article toward an entity, whereas *sentiment* refers to a sentence-level (pos/neg) label.

[3]Conservapedia is a wiki-based resource shaped by right-conservative ideas (Hube and Fetahu, 2018)

[4]Our convolutional NN model implementation is also affected by this constraint.

on human interpretable bias features (e.g. stance), and favors deep learning over classical machine learning methods for a more flexible interpretation of these features.

To develop explainability, techniques such as sensitivity analysis and layer-wise relevance propagation (LRP) address the black-box nature of deep learning models (Samek et al., 2017). This explores the limitations of deep convolutional architectures by assessing model performance against training and external articles, and identifying differences in the data that correlate with performance variations.

3 Data Querying and Setup

Both models require first acquiring article data and correcting imbalances to prevent model bias.

3.1 Data Sourcing

The news feed used to implement the models in our study is obtained through The Newscatcher API. [5] This API provides flexible methods for querying data, allowing users to specify attributes such as news sources, keywords, topics, etc. Both models are premised on the idea that, by exploring outlets with extreme or centrist political biases, three distinct categories of bias can be identified, establishing ground truth. This allows for assigning far-right, center or far-left political leanings to each group of queried articles.

We first query the available news sources provided by the API and then research political bias charts to identify trustworthy sources and select an eclectic group of news outlets. We adopt a well-known academic media bias classification 10.0 (University of Central Oklahoma Library, 2022), which is based on political bias and reliability.[6] Focusing solely on the political bias dimension, we select outlets situated within the colored circles in our simplified rendering of the news outlet spectrum shown in Figure 2. Specifically, **PBS, AP News, and News Nation Now** are chosen as center outlets, **Palmer Report and Bipartisan News** as far-left outlets, and **VDare, News Max, and Ricochet** as far-right outlets.

Although it would be ideal to consider a greater number of sources for each political category, access to outlets is limited by the available number of API calls and outlets accessible to the API. To ensure an evenly distributed news feed, the articles examined are restricted to a three-year range from January 1st, 2021 to December 31st, 2023.

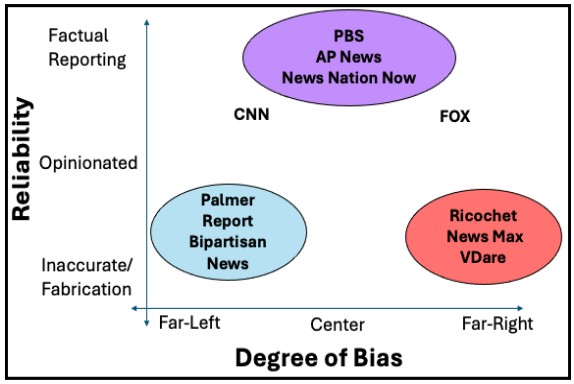

Figure 2: News outlet spectrum selected from Media Bias Chart 10.0 (University of Central Oklahoma Library, 2022): left, center, right

3.2 Exploratory Data Analysis

While the query loop aims to collect an equal amount of data from each outlet, it inadvertently results in an uneven distribution across outlets.[7] To mitigate potential model bias, each category is truncated to only contain ten thousand articles. Additionally, due to the depletion of API calls, this work prioritizes a balanced distribution across all articles relevant to each class, rather than striving for an equal distribution for each outlet. By systematically removing article entries within specific time intervals for different outlets, the resulting distributions for each category, although not perfect, are substantially improved.[8]

4 Model Implementations

The rule-based sentiment analysis model isolates sentiment expressed towards both common and proper nouns, leveraging adjectives and verbs that describe them. This approach aligns with findings from recent research, which focus on descriptive language used in relation to specific entities (Alam et al., 2022) to detect bias through sentiment and stance in news articles. The model employs coreference resolution to ensure direct reference of verbs and adjectives with correct name entities. Locating the nouns referenced by verbs and adjectives is accomplished through the aforementioned POS

[5] We are granted an educational license intended for research of non-commercial use.

[6] See https://adfontesmedia.com/static-mbc/ for a full rendering of news outlets from 2018 through 2024.

[7] Appendix A.1 reveals this discrepancy, showing the distribution by outlet for each three-month period, with certain outlets having significantly more queried articles than others.

[8] Appendix A.2 displays the final state of training data, accomplishing a relatively even distribution across time periods and outlet groupings.

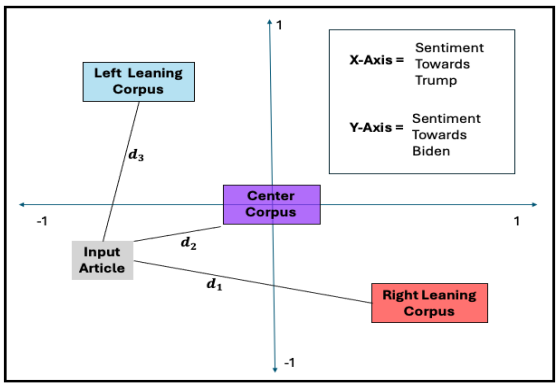

Figure 3: Theoretical mapping of left-vs-right space where an input articles positioned.

reference algorithm, which achieves acceptable performance based on precision, recall and F1 score.

Leveraging these rule-based outputs, the model creates sentiment vectors embedding the sentiment towards all nouns in articles by political leaning, where each dimension is defined by a unique noun. Sentiment is quantified using the valence scores of all verbs and adjectives considered. It classifies bias by comparing the cosine distance between an article's vector and sentiment vectors for each political leaning. Then the political leaning closest to the article's vector is predicted as its bias.

Figure 3 shows a theoretical mapping of the three corpora projected onto a 2D plane, with each dimension representing sentiment toward a corresponding entity based on all the adjectives and verbs referencing it within each corpus. As expected, the right-leaning corpus shows negative sentiment towards Biden and positive sentiment towards Trump, while the left-leaning corpus expresses the opposite. The center-leaning corpus displays a neutral sentiment towards both. An input article is then positioned on the plane based on the sentiment it expresses towards both entities. The shortest cosine distance is found between the input article and the center corpus, indicating that the article's stance is most aligned with the center corpus. This suggests that a highly negative stance towards Trump, with a moderately negative stance towards Biden, indicates a politically centered standpoint.

By contrast, the deep learning model processes raw text directly, without segmenting or extracting stance-specific meaning. The convolutional NN model captures dependencies and recurring structures in text through multiple deep learning layers. This model achieves strong performance (see Section 5) in classifying articles across three outlets.

Both the rule-based and deep learning models are applied to the preprocessed dataset. The implementation of the rule-based model (Section 4.1) is more involved than that of the deep-learning convolutional NN model (Section 4.2) in that the latter uses a standard architecture, whereas the former proposes a novel design. Our models are powered by 2 AMD EPYC 75F3 CPU cores complemented by 2 NVIDIA A100 GPU cores. We use 80% of the data for training and the remaining 20% for testing the models. Although the training suite encompasses the full extent of the time period explored, temporal leakage is not an issue, as the models are not devoted to the prediction of bias in future news articles. The idea of a temporal dataset is solely meant to provide a more comprehensive span of biased text.

4.1 Rule-Based Sentiment Analysis Model

Our rule-based sentiment analysis implementation aims to identify political bias by extracting and quantifying the sentiment expressed towards nouns through the verbs and adjectives that refer to them. This involves coreference resolution, dependency parsing, POS reference resolution, sentiment vectorization, and cosine distance as the ultimate classification metric. Each step is detailed below.

4.1.1 Coreference Resolution

Our study resolves coreference to prevent the aggregation of sentiment for lexically equivalent nouns that represent different entities. Two common examples in the dataset include the use of pronouns and common nouns to reference named entities. For example, the text "John is gifted. *He* was always good at math." becomes "John is gifted. *John* was always good at math." This allows us to attribute both the adjectives "good" and "gifted" to John rather than associating "gifted" with "he".

Without coreference resolution, the resulting sentiment dictionaries would inaccurately average sentiments expressed toward entities referred to by the same pronouns, significantly undermining the model's effectiveness. We employ the spaCy coreference resolution model (Kádár et al., 2022), an end-to-end neural system applicable to various entity coreference tasks.

4.1.2 Dependency Parsing and Part of Speech Reference Resolution

With coreferences resolved, the model associates verbs and adjectives with their corresponding nouns using spaCy dependency trees (Honnibal and

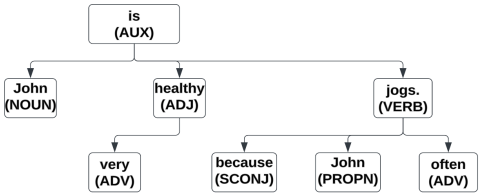

Figure 4: Dep Tree: Algorithm Walk-through

Montani, 2017). The complexity of dependency paths makes rule-based identification of all noun-adjective-verb relations challenging. To maintain sentiment accuracy without high computational costs, we balance relation accuracy with the number of identified relations.

We observe that, while nouns are not always associated with a modifying verb or adjective, verbs and adjectives almost always imply the presence of a noun. Accordingly, instead of finding relations from nouns to verbs and adjectives, the method identifies relations from verbs and adjectives to the nearest noun, regardless of its position. This noun is then considered the one being referenced by the verbs and adjectives it stems from.

The algorithm uses bottom-up dynamic programming to reduce complexity. It progressively updates (int, string) pairs corresponding to each token in a sentence with the distance from the closest noun to that token and the noun itself with a complexity of $O(N^2)$.

For clarity, consider the sentence "John is very healthy because *he* often jogs" after coreference resolution: "John is very healthy because *John* often jogs". Figure 4 shows the dependency tree for this enriched sentence. A memoization array of length eight is initialized (Table 1). For example, the entry at index 1 contains a distance of 1 because the noun "John" is a child of the auxiliary "is".

Index	0	1	2	3	4	5	6	7
Dist.	0	1	-1	-1	-1	0	-1	1
Noun	John	John				John		John

Table 1: Memoization: Algorithm Walk-through

Starting with verbs, the algorithm examines index 7 corresponding to the verb "jogs" with entry (1, "John") and traverses upward $D - 1$ times to find a closer distance. Since $D = 1$, no parents are considered and "jogs" is associated with "John". A similar procedure is followed for adjectives, e.g., index 3 corresponds to "healthy" with entry (-1, ""). Given that there is no noun successor to "healthy," the algorithm recursively traverses parent nodes to find the shortest distance. It considers the auxiliary "is", corresponding to entry (1, "John"). Since "is" is one edge above "healthy," and "John" is one edge above "is," "John" is determined to be two edges away from "healthy." Further traversal is unnecessary as no shorter distance exists, confirming that "healthy" corresponds to "John".

In order to verify the algorithm's viability for POS reference resolution, we use a dataset containing 100 random sentences, comparing the algorithm's identified relations with human-annotated relations. The sentences are generated using Chat GPT 4o (OpenAI, 2024), and varied in their use of verbs, adjectives, and nouns. The POS relations used as a ground truth are verified by a human. The tests yield F1-Scores of 0.80 for adjective-noun relations and 0.71 for verb-noun relations.

4.1.3 Sentiment Vectorization and Cosine Distance Computation

Sentiment vectorization defines an N-dimensional space with a vector of length N, where each dimension pertains to a unique noun from the corpus of articles. Each group of news outlets is mapped onto this space. An article is mapped by extracting the sentiment expressed towards nouns in it, considering only those nouns present in the original training corpus. The closest vector indicates the primary political leaning expressed in the article.

Using dependency parsing and the POS Reference Resolution algorithm to identify adjective and verb references to nouns, we create a mapping that associates each noun in an article with a list of referencing adjectives and verbs. The valence score of each verb and adjective is extracted using the TextBlob library (Loria, 2018), where valence indicates the degree of positive or negative sentiment. The sentiment towards a noun is defined by the average valence score of all referring adjectives and verbs. This process is repeated for each article within a political leaning. By merging these mappings, we obtain final mapping with keys representing all unique nouns in the corpus, each pointing to the average valence score of all its mentions.

Applying this process to each of the three article groups produces three distinct mappings, each containing the nouns found in their respective corpus. To compare these groups, the mappings are consolidated to hold references to the same group of nouns. A separate mapping assigns an index to each noun across all three corpora. If N distinct nouns are identified, a vector of length N is defined for each article group. The sentiment expressed towards Kth noun is assigned to index K.

For indices where the relevant noun is absent in a group's corpus, a sentiment score of 0 is assigned.

This process produces three equal-length vectors in an N-dimensional space, representing sentiment toward all nouns in the corpus. To classify an article, valence scores for all nouns are computed and added onto an N-length vector, ignoring nouns not in the training corpus. The cosine distance between this vector and each of the three original vectors is calculated, assigning the article the political leaning of the closest vector.

Consider the simplified example in Table 2, where three sub-tables show the stance of each article group towards their respective nouns. Each unique noun is assigned an identifier (Table 3). Using this mapping, the initial stance tables are converted into vectors of length 6 (Table 4), with absent nouns assigned a score of 0.

Noun	Trump	IRA	Israel	Immigrant
Left Stan.	-0.7	0.5	0.1	0.3

Noun	Trump	IRA	Israel	Vaccine
Right Stan.	0.8	-0.1	0.8	-0.5

Noun	Trump	IRA	Israel	China
Center Stan.	-0.2	0.1	0.3	-0.1

Table 2: Noun Stance by Corpus

Trump	IRA	Israel	Immigrant	Vaccine	China
0	1	2	3	4	5

Table 3: Noun Identifier Mapping

Identifier	0	1	2	3	4	5
Left	-0.7	0.5	0.1	0.3	0.0	0.0
Right	0.8	-0.1	0.8	0.0	-0.5	0.0
Center	-0.2	0.1	0.3	0.0	0.0	-0.1

Table 4: Vectorization

The first sub-table of Table 5 shows a stance dictionary for an article to be classified, listing all nouns in the article and their associated stance scores. The second sub-table appends this dictionary to the end of Table 4 using the aforementioned index mapping. Note that "Canada", a noun not in the training corpus, is absent from the classification vector. The third sub-table shows classification by calculating the cosine distance between the article's vector and each of the three vectors representing political leanings.

4.2 Convolutional NN Model

We choose to use a convolutional NN model to classify bias since convolutional models employ a highly unconstrained assessment of features through their convolutional and pooling layers,

Noun	Trump	Immigrant	Canada
Article Stan.	-0.3	0.10	0.05

Identifier	0	1	2	3	4	5
Left	-0.7	0.5	0.1	0.3	0.0	0.0
Right	0.8	-0.1	0.8	0.0	-0.5	0.0
Center	-0.2	0.1	0.3	0.0	0.0	-0.1
Article	-0.3	0.0	0.0	0.1	0.0	0.0

X	Left	Right	Center
Cosine Dist.(Article, X)	0.17	1.51	0.51

Table 5: Evaluation Process

which can capture complex patterns in text data. A Convolutional NN is chosen over common models applied to textual analysis (e.g. transformers) for their ability to apply a uniform focus on features across the input, maintaining a more liberal and direct interpretation of data. This unconstrained feature assessment contributes to a lack of explainability, as the internal logic of the convolutional model remains opaque. In the sections below, we discuss opaqueness as a limitation that challenges complete reliance on deep learning methodologies for complex classification tasks. Instead, we argue that rule-based or hybrid approaches would provide greater transparency.

Inspired by the work of Prosise (2023), we combine datasets representing three political leanings. After removing stop words from each article, a Keras tokenizer assigns an index to each unique word enabling the neural network to interpret input and identify patterns for political classification. The input embeddings are of 32 dimensions and the model consists of two convolution layers with a max pooling layer in between and a global max pooling layer at the end. The model is trained over five epochs using the Adam optimizer and categorical cross-entropy loss to improve accuracy using a validation dataset.[9] We use the tensorflow library under the Apache License 2.0. (For more details, see Appendix A.4.)

5 Results

We evaluate both models' performance by examining precision, recall, and F1 across the three classes. ,using a dataset comprising 20% of the original data. Results reflect the models' classification accuracy, not their ability to recognize political bias. We revisit this distinction in our evaluation of model performance on external news outlets.

[9]Details regarding training and validation process are provided in Appendix A.5

5.1 Sentiment Analysis Model: Evaluation

Each tested article undergoes coreference resolution, and the sentiment towards all nouns is quantified into a vector. This vector is then compared to the vectors representing the three political leanings. The article is assigned the political leaning corresponding to the closest vector.

Table 6 shows the performance of the rule-based model for each classification group. The *Left* class has the highest F1-Score of 0.57, with the *Center* and *Right* classes having slightly lower F1-Scores of 0.51 and 0.52, respectively.

	Precision	Recall	F1-Score
Left	0.78	0.45	0.57
Center	0.42	0.66	0.51
Right	0.48	0.56	0.52

Table 6: Rule-based Model: Metrics

5.2 Convolutional NN Model: Evaluation

Table 7 shows the convolutional NN model's performance for each classification group. The *Left* class has the highest F1-Score of 0.98, indicating excellent performance, with *Center* and *Right* classes having F1-Scores of 0.93 and 0.91, respectively.

	Precision	Recall	F1-Score
Left	0.98	0.98	0.98
Center	0.91	0.96	0.93
Right	0.94	0.88	0.91

Table 7: Convolutional NN Model: Metrics

5.3 Application and Insights: CNN and FOX

The applicability of both models is explored by classifying articles from news outlets not included in the training corpus. This approach distinguishes a model's ability to recognize bias from simply differentiating between training outlets. The rule-based approach aims to target and extract text features that express stance, ignoring non-political rhetoric or features. Conversely, the convolutional NN model is allowed complete freedom to differentiate between corpora by any means available. This makes the convolutional NN model sensitive to corpora that show distinguishing features past their expression of political bias.

Although the convolutional NN model accurately categorizes articles in the training corpus, this does not necessarily translate to accurate interpretation of bias. By the same token, the rule-based model's lower accuracy in classifying articles does not mean it is worse at recognizing bias than the covolutional NN model. To focus solely on political bias detection, we exclude test outlets from the training corpus, preventing the models from leveraging similarities in prose, structure, and other lexical features within each group of outlets.

CNN and FOX News are used to test the models beyond the outlets in the training data. These outlets are chosen because they are among the country's largest news media corporations and are widely acknowledged for representing opposite ends of the political spectrum. While their opinions are expected to align closer to with left-leaning and right-leaning classes, they are also widely read and resemble center-leaning articles in style and structure. We consider 1,500 articles for each outlet over the three-year period of the training corpus.[10] To incorporate a temporal analysis and evaluate the models' predictions across different periods of political tension, batches for each three-month period within the three years are classified separately. Tables 8 and 9 show the distribution of predictions across political leanings throughout each time period for each model applied on both FOX and CNN articles. A darker shade for a given entry indicates a higher percentage of articles classified as pertaining toward that political leaning for that time period.

	21Q2	21Q3	21Q4	22Q1	22Q2	22Q3	22Q4	23Q1	23Q2	23Q3
Left	0.09	0.09	0.02	0.01	0.04	0.06	0.07	0.01	0.07	0.07
Center	0.62	0.67	0.79	0.81	0.83	0.7	0.75	0.73	0.62	0.61
Right	0.29	0.24	0.19	0.18	0.13	0.24	0.18	0.25	0.31	0.31

	21Q2	21Q3	21Q4	22Q1	22Q2	22Q3	22Q4	23Q1	23Q2	23Q3
Left	0.15	0.21	0.18	0.2	0.27	0.21	0.21	0.24	0.22	0.26
Center	0.57	0.6	0.55	0.26	0.38	0.5	0.36	0.41	0.47	0.46
Right	0.28	0.19	0.27	0.54	0.36	0.29	0.43	0.35	0.31	0.27

Table 8: Convolutional Model. Bias Classification of FOX (top) and CNN (bottom) articles Over Time.

	21Q2	21Q3	21Q4	22Q1	22Q2	22Q3	22Q4	23Q1	23Q2	23Q3
Left	0.1	0.11	0.04	0.11	0.08	0.13	0.11	0.15	0.07	0.09
Center	0.49	0.55	0.62	0.49	0.55	0.61	0.57	0.6	0.55	0.67
Right	0.4	0.33	0.33	0.39	0.37	0.26	0.32	0.25	0.38	0.25

	21Q2	21Q3	21Q4	22Q1	22Q2	22Q3	22Q4	23Q1	23Q2	23Q3
Left	0.04	0.05	0.09	0.16	0.09	0.1	0.06	0.09	0.09	0.08
Center	0.73	0.7	0.61	0.53	0.66	0.66	0.72	0.69	0.65	0.71
Right	0.24	0.24	0.03	0.32	0.25	0.25	0.23	0.22	0.26	0.21

Table 9: Rule-Based Model. Bias Classification of FOX (top) and CNN (bottom) articles Over Time.

The Convolutional and Rule-Based Model results on FOX articles show that the models often classify most articles in each period as Center-leaning, with the majority of the remaining portion classified as Right-leaning. The Convolutional

[10] Appendix A.3 illustrates the distribution of CNN and Fox articles, with roughly 1500 articles classified for each outlet

Model is less likely to predict the Left class compared to the Rule-Based Model, favoring more confident Center predictions. Meanwhile, the Rule-Based Model assigns more articles to the Right class than the Convolutional Model. Both models' predictions lean more toward the Center than public perception of FOX, though they still align with its center-right reputation.

In contrast, the Convolutional Model's results on CNN articles differ from the general perception of the outlet. Over a third of the articles are classified as Center-leaning, with the rest slightly favoring the Right class. The Rule-Based Model classifies most CNN articles as Center, with the remaining majority leaning Right. Although neither model's predictions match CNN's center-left stance, the Rule-Based Model tends to classify articles further left than the Convolutional Model.

5.4 Model Explainability

The Rule-Based Model's transparency and strong theoretical foundation allow us to attribute shortcomings in both the corpus and external dataset results to specific components of the model architecture. A combination of factors prevents the model from accurately assigning sentiment to entities, resulting in instances where detected sentiments do not align with political biases.

Understanding the performance differences between the corpus and external dataset for the Convolutional Model is more challenging due to its black-box deep learning architecture. To investigate this gap, we employ LIME (Ribeiro et al., 2016) to identify the words that most influence article classification. A subset of the test suite is analyzed, revealing the 20 most important words in each article's classification. The frequency of the top 25 influential words for each political leaning is shown in Figures 5, 6, and 7. The LIME analysis reveals three types of words that the Convolutional Model relies on for classification.

The first influential word types are those that recur frequently due to the limited number of outlets comprising the corpus, e.g., "Palmer" and "Report" (Figure 6) and "AP" and "Associated" (Figure 7). The second type comprises words that lack political meaning on their own but are common in the rhetoric of certain outlets, e.g., "us" and "said" (Figure 5) and "apparently" (Figure 6). The third type includes nouns with inherent political connotations, e.g., "Trump", "Leftist", "aliens" "riot" and "GOP" (Figure 6) and "Republican", "Democratic" and "Capitol" (Figure 7).

From this analysis, we conclude that the Convolutional Model struggles to maintain its high performance when applied to external news outlets, primarily because CNN and FOX articles lack the first and second types of influential words that are specific to the outlets used for training. Without relying on rhetoric similarities or outlet-specific names, the model assesses politically charged terms—words it does not emphasize adequately during training to draw reliable conclusions.

Notably, Right leaning predictions are the least reliant on the first and second type of words, and are generally less reliant on any given word in the classification of articles. This explains why FOX article predictions by the Convolutional model aligns more with the outlet's political stance than CNN article predictions. The improved performance resulting from the Convolutional Models's focus on politically charged words supports the Rule-Based model's framework, which is primarily designed to detect sentiment towards such words.

Left leaning predictions rely heavily on the first type of influential terms, causing the Convolutional Model to perform poorly on external data when predicting CNN articles. We hypothesize that the superior performance of the Rule-Based Model in this task stems from its ability to focus on nouns associated with political entities, which the Convolutional Model does not sufficiently emphasize in its classification of Left leaning articles.

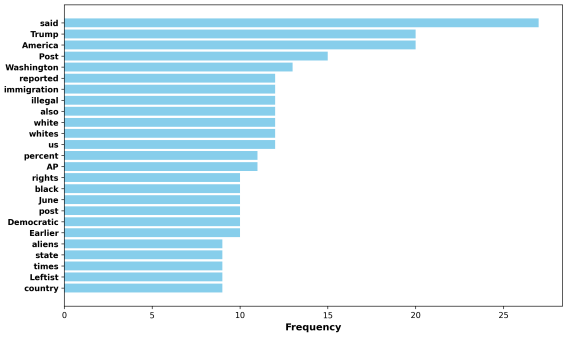

Figure 5: Top 25 Influential Terms in Right Class Classification

6 Conclusion and Future Work

This paper examines two models for classifying political bias in news media: a sentiment analysis rule-based model and a convolutional NN model. Given the complexity of politically biased text, a ground truth is established using the political spectrum placement of widely read news outlets by

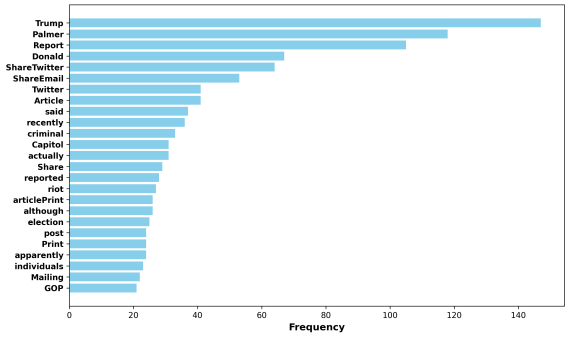

Figure 6: Top 25 Influential Terms in Left Class Classification

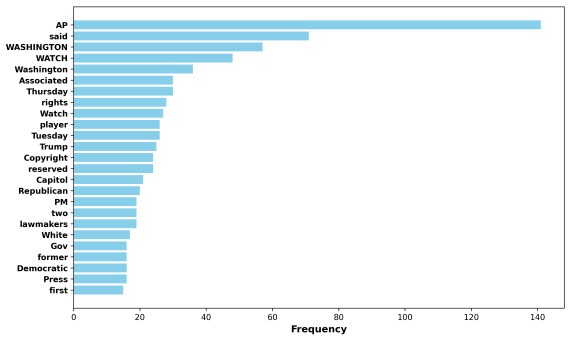

Figure 7: Top 25 Influential Terms in Center Class Classification

credible academic sources (University of Central Oklahoma Library, 2022).

The rule-based model applies coreference resolution and a POS reference algorithm to extract stance towards nouns, mapping them into an N-dimensional space for comparison with input articles. The convolutional NN model focuses on identifying distinctive patterns.

Results initially indicate that the convolutional NN model significantly outperforms the rule-based in accuracy. However, when models are tested on external data, using CNN and FOX articles, the limitations of the convolutional NN model are uncovered through its significant change in performance. The rule-based model, in contrast, retains its initial performance, thereby demonstrating its adaptability to different datasets.

Potential improvements to the rule-based model include incorporating machine learning techniques for feature extraction and input classification, such as using a decision tree instead of using a closest POS resolution algorithm to indentify noun relations. Alternatively, stance detection would benefit from leveraging more accurate pre-trained models in Aspect Based Sentiment Analysis (Hoang et al., 2019) to better quantify targeted sentiment. Additionally, understanding synonymy through tools like Word2Vec could help map nouns in input articles to similar counterparts in classification vectors, enabling more effective classification.

Improvements to the Convolutional NN model include prioritization of explainability and generalizability. Future work involves a thorough data selection process paired with an iterative analysis, using LIME or SHAP, to ensure the use of corpora that do not allow models to hinge predictions on terms unrelated to the classification task. In the classification of bias, this process translates to the prioritization of true bias indicators and disregard of irrelevant stylistic nuances.

Lastly, future work could incorporate large language models into the study by exploring their performance in the classification of bias and their potential improvement through hybridization techniques. In addition, expanding the study to include a training corpus from a diverse range of news outlets would help to prevent models from relying on stylistic differences in writing.

Limitations: A Case for Hybridization

Overall, this exploration examines the extremes of techniques used for media bias classification. It contrasts a clearly defined, rule-based model with a deep learning model that has an opaque internal methodology. The rule-based model, while theoretically sound, fails to beat the convolutional NN in testing, but shows a similar performance in external applications. Both approaches have shortcomings that could be mitigated through hybridization.

Both models are evaluated using ground truth for political bias in news articles, determined by the publishing outlets and academic sources that classify the outlet's political leaning. However, political bias is a highly dynamic, nuanced, and subjective expression that cannot be fully captured through the perspectives of various news outlets. While our research aims to investigate bias in text, the models we construct are ultimately designed to classify articles based on lexical and syntactic features of the three corpora considered. Thus, they classify text, rather than classify bias directly. Although the dataset facilitates temporal and diverse analysis of political news media, datasets with articles annotated specifically for bias would provide a more robust ground truth. Additionally, the dataset encompasses only US news outlets, which limits the broader international applicability of models trained using it.

The rule-based sentiment detection model focuses solely on the sentiment expressed toward

nouns, avoiding irrelevant textual features related to political leaning. While this approach offers insight into how political bias is conveyed, the model does not target other forms of bias (e.g. Framing bias) and additionally considers nouns of a non-political nature in its classification process, which may not necessarily indicate political leaning. Beyond its broad interpretation of bias, the model faces challenges regarding its practicality. Since the model only interprets one feature, sentiment expressed towards nouns, an article must contain mentions of nouns found in the corpus for its political leaning to be accurately classified. Furthermore, if the nouns within the article are apolitical or rare, the identified bias may lack substantial basis.

The POS reference method for the rule-based model sometimes misses correct relationships or incorrectly identifies them. This is because the algorithm assumes a one-to-many relationship between nouns and their referencing parts of speech (verbs or adjectives), even though many-to-many or many-to-one relationships are possible. For example, in the sentence "John is happy and excited", the one-to-many relationship between the noun "John" and the adjectives "happy" and "excited" is identified correctly. However, in the sentence "John and Peter are happy", which has a many-to-one relationship between the nouns "John" and "Peter" and the adjective "happy," the algorithm only links "happy" to the closest noun, "Peter".

Despite the convolutional NN model's impressive classification performance when tested on outlets found in the training corpus, its focus on political bias as a deciding factor is shown to be insufficient. The model accurately categorizes the three classes in the training corpus, but it identifies a strong moderate leaning for FOX and fails to converge on a general political leaning for CNN articles. Due to the inherent opacity of deep learning models, the specific textual features used for classification are unpredictable, leaving developers to speculate on the mix of features driving article classification and how much these features are influenced by the political bias of each outlet.

Ideally, classification of political bias in news media would combine the feature extraction of a rule-based model with the self-correction of a convolutional NN model. By examining additional text features that signal political bias and quantifying them similarly to sentiment expression, a suitable input vector for convolution could be generated. Although the internal processes of the convolutional NN would remain opaque to the developer, its predictions would focus solely on factors related to political bias. Allowing the developer to set the initial parameters of the neural network would enable the imposition of constraints while preserving its self-learning ability, thereby ensuring that only relevant resources are used for learning.

Ethics Statement

The data used for this study is obtained using the News Catcher API, but is otherwise publicly accessible. The API is employed to allow for fast and efficient sourcing of a large number of articles. The integrity of the data is maintained by verifying the reputability of the API used and by assessing the articles queried.

Maintaining objectivity is crucial in this study on automatic detection of political bias in text. Both implemented models use standardized datasets and transparent processes to ensure a fair analysis of results. It is important to emphasize that our models' evaluation of CNN/FOX bias is not intended as a definitive judgment of their political leaning. Rather, it serves as an exercise to demonstrate the capabilities and limitations of NLP techniques in analyzing political bias with respect to a well-known academic media bias classification (University of Central Oklahoma Library, 2022).

The use of AI in this study, seen primarily through the CNN model, is done responsibly. We acknowledge AI's limitations in assessing a highly subjective and sensitive subject as is political bias. In fact, this study argues for greater transparency to transcend opaque deep learning systems.

Acknowledgement

This work is supported, in part, by DARPA Contract No. HR001121C0186. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the US Government.

References

Mehwish Alam, Andreea Iana, Alexander Grote, Katharina Ludwig, Philipp Müller, and Heiko Paulheim. 2022. Towards analyzing the bias of news recommender systems using sentiment and stance detection. In *Companion Proceedings of the Web Conference 2022*, WWW '22, page 448–457, New York, NY, USA. Association for Computing Machinery.

Wei-Fan Chen, Khalid Al Khatib, Benno Stein, and Henning Wachsmuth. 2020. Detecting media bias in news articles using Gaussian bias distributions. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4290–4300, Online. Association for Computational Linguistics.

Carroll Doherty, Jocelyn Kiley, Nida Asheer, and Talia Price. 2023. Americans' dismal views of the nation's politics.

Xue-yong Fu, Cheng Chen, Md Tahmid Rahman Laskar, Shayna Gardiner, Pooja Hiranandani, and Shashi Bhushan Tn. 2022. Entity-level sentiment analysis in contact center telephone conversations. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 484–491, Abu Dhabi, UAE. Association for Computational Linguistics.

Felix Hamborg, Anastasia Zhukova, and Bela Gipp. 2019. Automated identification of media bias by word choice and labeling in news articles. In *2019 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, pages 196–205.

Mickel Hoang, Oskar Alija Bihorac, and Jacobo Rouces. 2019. Aspect-based sentiment analysis using BERT. In *Proceedings of the 22nd Nordic Conference on Computational Linguistics*, pages 187–196, Turku, Finland. Linköping University Electronic Press.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Christoph Hube and Besnik Fetahu. 2018. Detecting biased statements in wikipedia. In *Companion Proceedings of the The Web Conference 2018*, WWW '18, page 1779–1786, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

Morgan Kelly. 2021. Political polarization and its echo chambers: Surprising new, cross-disciplinary perspectives from princeton.

Ákos Kádár, Paul O'Leary McCannPaul, O'Leary McCann, Richard Hudson, Edward Schmuhl, Sofie Van Landeghem, Adriane Boyd, Madeesh Kannan, and Victoria Slocum. 2022. End-to-end neural coreference resolution in spacy. `https://explosion.ai/blog/coref`.

Steven Loria. 2018. textblob documentation. *Release 0.15*, 2.

Manman Luo and Xiangming Mu. 2022. Entity sentiment analysis in the news: A case study based on negative sentiment smoothing model (NSSM). *International Journal of Information Management Data Insights*, 2(1):100060.

Tomas Mikolov, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *International Conference on Learning Representations*.

Sendhil Mullainathan and Andrei Shleifer. 2002. Media Bias. NBER Working Papers 9295, National Bureau of Economic Research, Inc.

OpenAI. 2024. Chat gpt (version 4.0) [large language model].

Jeff Prosise. 2023. Text classification with neural networks. *Atmosera*. Accessed: 2024-06-13.

Shaina Raza, Oluwanifemi Bamgbose, Veronica Chatrath, Shardule Ghuge, Yan Sidyakin, and Abdullah Yahya Mohammed Muaad. 2024. Unlocking bias detection: Leveraging transformer-based models for content analysis. *IEEE Transactions on Computational Social Systems*.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why should i trust you?": Explaining the predictions of any classifier.

Francisco-Javier Rodrigo-Ginés, Jorge Carrillo de Albornoz, and Laura Plaza. 2024. A systematic review on media bias detection: What is media bias, how it is expressed, and how to detect it. *Expert Systems with Applications*, 237:121641.

Egil Rønningstad, Erik Velldal, and Lilja Øvrelid. 2022. Entity-level sentiment analysis (ELSA): An exploratory task survey. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 6773–6783, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Diego Saez-Trumper, Carlos Castillo, and Mounia Lalmas. 2013. Social media news communities: gatekeeping, coverage, and statement bias. In *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management*, CIKM '13, page 1679–1684, New York, NY, USA. Association for Computing Machinery.

Wojciech Samek, Thomas Wiegand, and Klaus-Robert Müller. 2017. Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models.

University of Central Oklahoma Library. 2022. Media bias chart - how to avoid misinformation. `https://library.uco.edu/misinformation/mediabias`.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

A Appendix

A.1 Initial Data Distribution Figures

Figures 8–15 show the initial distributions for each of the eight news outlets considered in the construction of both models. The distributions are separated into groups based on the three biases being explored.

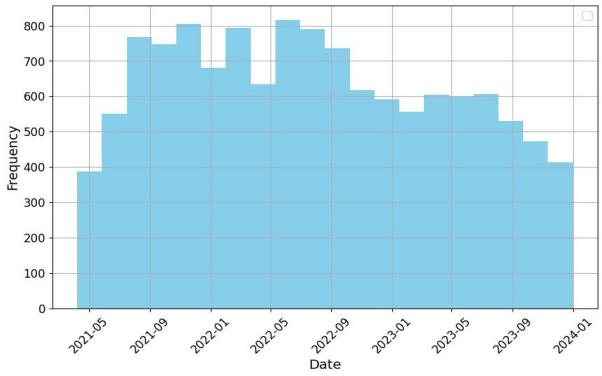

Figure 8: Left-Leaning Outlets: Bipartisan News

Figure 9: Left-Leaning Outlets: Palmer Report

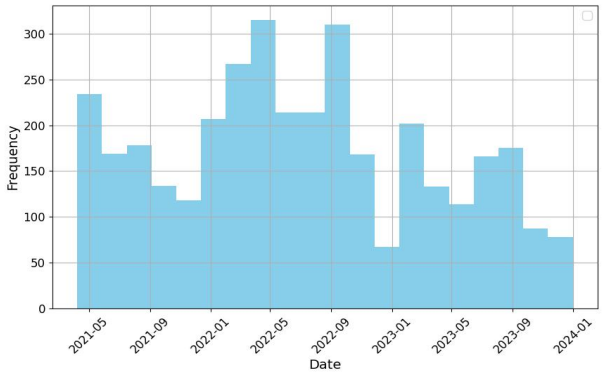

Figure 10: Right-Leaning Outlets: VDare

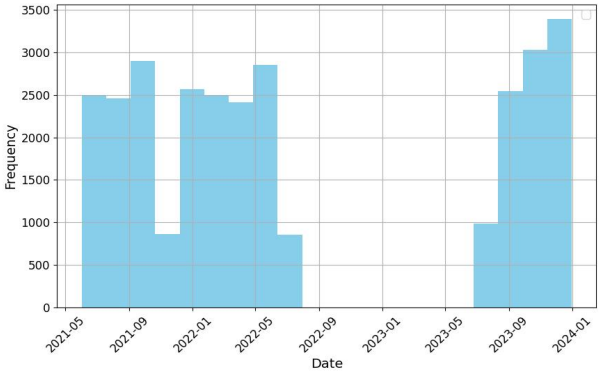

Figure 11: Right-Leaning Outlets: News Max

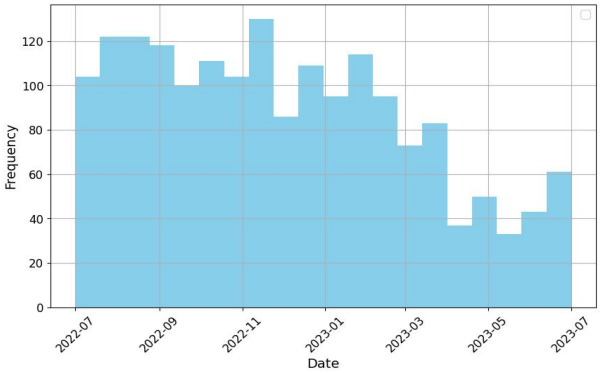

Figure 12: Right-Leaning Outlets: Ricochet

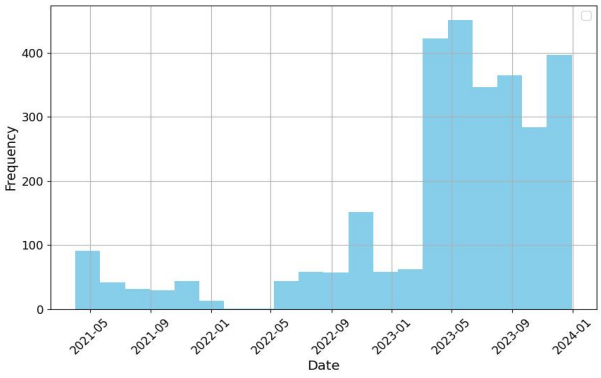

Figure 13: Center-Leaning Outlets: News Nation Now

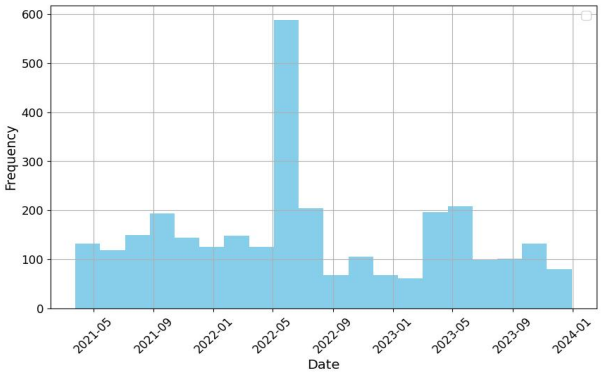

Figure 14: Center-Leaning Outlets: AP News

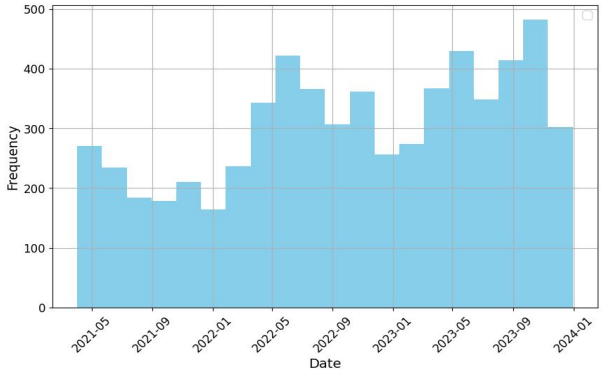

Figure 15: Center-Leaning Outlets: PBS

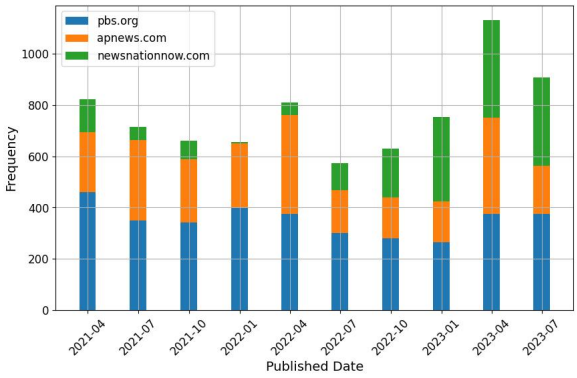

Figure 18: Center Outlets Aggregate Distribution

A.2 Resulting Distribution Across News Outlets for Each Political Grouping

Figure 16- 18 presents final state of training data (University of Central Oklahoma Library, 2022), demonstrating even distributions across different time periods and outlet grouping.

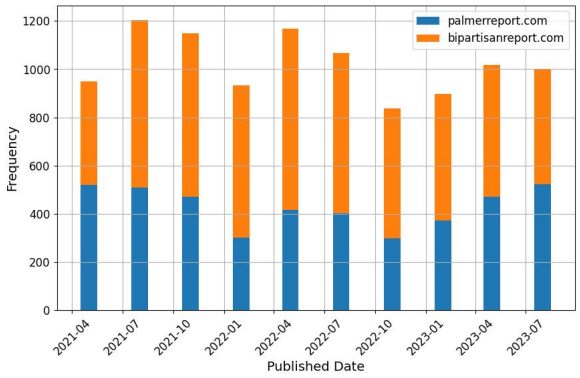

Figure 16: Left Outlets Aggregate Distribution

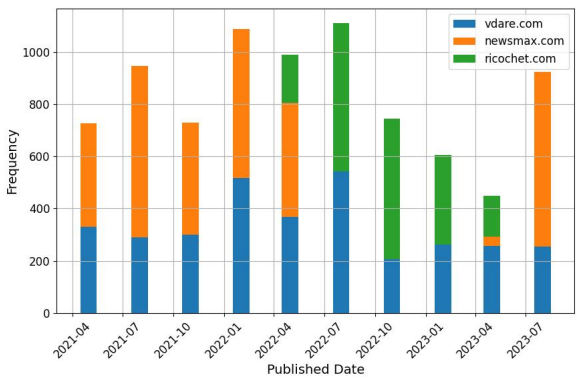

Figure 17: Right Outlets Aggregate Distribution

A.3 Application Data Distribution Figures

The histograms below show the articles queried from FOX and CNN. These articles are used to apply the models developed throughout the study to external news outlets. As can be seen in Figures 19 and 20, roughly 1500 articles are queried from each outlet to be evenly distributed throughout the 3 year interval explored.

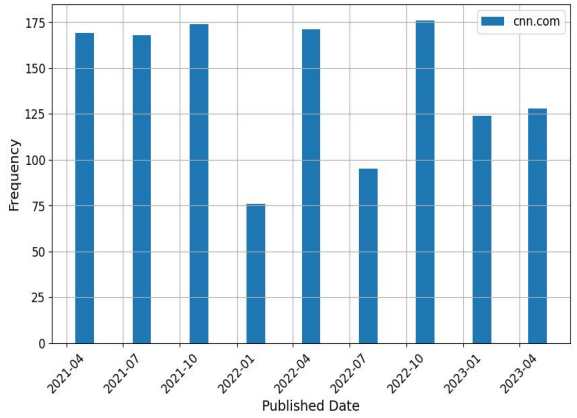

Figure 19: CNN Articles: Distribution by Period

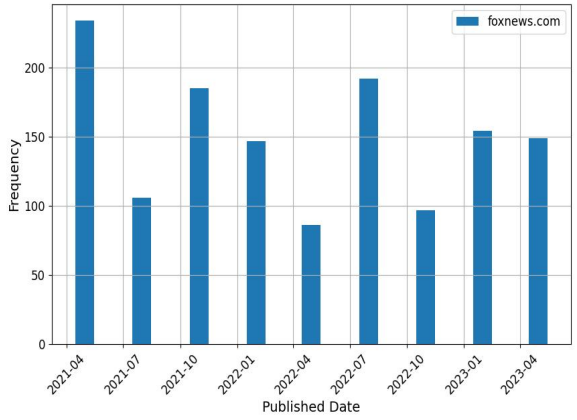

Figure 20: FOX Articles: Distribution by Period

A.4 Convolutional NN Model Architecture

The code shown in Figure 21 defines the architecture for the convolutional neural network model, which is largely inspired by the work of Prosise (2023). First an Embedding layer converts integer-encoded vocabulary into dense vectors of fixed size. This layer efficiently handles the vast vocabulary of text data, providing meaningful representations of words that capture semantic similarities based on their context within the corpus. This dense vector representation allows the model to interpret text input effectively, facilitating identification of patterns relevant to classification tasks.

```
from tensorflow.keras.models import
    Sequential

from tensorflow.keras.layers import
    Embedding, Flatten, Dense, Conv1D,
    MaxPooling1D, GlobalMaxPooling1D

num_classes = 3

model = Sequential()

# Embedding layer
model.add(Embedding(100000, 32))

# First convolutional layer
model.add(Conv1D(32, 7,
    activation='relu'))

# First pooling layer
model.add(MaxPooling1D(5))

# Second convolutional layer
model.add(Conv1D(32, 7,
    activation='relu'))

# Global max pooling layer
model.add(GlobalMaxPooling1D())

model.add(Dense(num_classes,
    activation='softmax'))
model.compile(loss=
    'sparse_categorical_crossentropy',
              optimizer='adam',
              metrics=['accuracy'])
```

Figure 21: CNN Architecture Code

Following the Embedding layer are two sets of one-dimensional convolution layers and Max Pooling layers. The convolution layers apply convolutional operations to the embedded word vectors, using filters to extract local patterns (such as the presence of specific n-grams) indicative of the text's class. The rectified linear unit (ReLU) activation function ensures that the model captures nonlinear relationships between these features. Each convolution layer is followed by a Max Pooling layer, which reduces the dimensionality of the data by retaining only the most prominent features, thus improving computational efficiency and helping to prevent overfitting.

After the second convolution and pooling sequence, a Global Max Pooling layer aggregates the most significant features from across the entire text, ensuring that the model's final predictions are informed by the most impactful elements of the input data. The architecture culminates in a Dense layer with a Soft Max activation function, which maps the extracted features to probabilities across the three classes, allowing the model to quantify and output the distinctions noted between classes explored. The model is trained through five separate epochs, making use of a validation dataset to progressively increase its accuracy.

A.5 Convolutional NN Model Training

The plot in Figure 22 shows the progression of the convolutional NN model's training and validation accuracy throughout the five training epochs. Training accuracy defines the model's ability to precisely classify articles it recurrently sees throughout each epoch, whereas validation accuracy refers to the model's ability to generalize to unseen data. It is common for validation accuracy and training accuracy to initially increase together. When validation accuracy plateaus, while training accuracy continues to increase, the model begins to overfit to its training data and looses its ability to generalize to external data (e.g. validation data). Five epochs are sufficient to train the model as the validation accuracy begins to plateau around the fourth epoch.

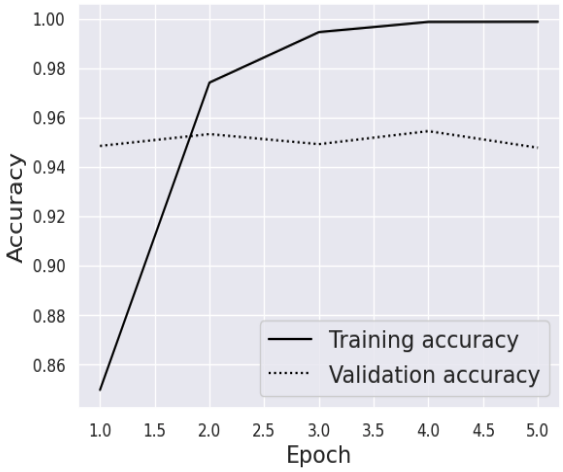

Figure 22: Convolutional NN Model Training Process: Training and Validation Accuracy