Classifying Partisan Bias in News Articles: Leveraging an Understanding of Political Language and Article Structure

Stanford CS224N Custom Project

Emily Jin
Department of Mathematics
Stanford University
emilyjin@stanford.edu

Edoardo Yin
Department of Computer Science
Stanford University
edoyin@stanford.edu

Abstract

As a first step toward mitigating the harmful effects of political bias in the media, we classify news articles into 5 partisan bias categories (left, left-center, neutral, right-center, and right) and analyze important words for bias detection. We explore the effectiveness of fine-tuning BERT embeddings and Hierarchical Attention Transformers (HAT) on political text to predict partisan bias, both separately and together. By combining both, we expected our model to better classify text by leveraging a strong understanding of political language, and the underlying structure of news articles. We achieve an impressive acccuracy of 94.7% with our model, which is a significant improvement over baselines but slightly lower than the individual BERT or HAT classifiers. Our visualizations of the finetuned BERT embeddings demonstrate a strong understanding of political language though, which shows promise for using contextualized word embeddings to analyze partisan language. Furthermore, by achieving promising results on partisan bias detection, we hope to inspire others to address growing concerns about bias in the media and its detrimental effects to the future of society.

1 Key Information to include

- Mentor: Hans Hanley
- External Collaborators (if you have any): N/A
- Sharing project: Yes, CS324

2 Introduction

Background and Motivation The media has long played a critical role in democracy, by providing accurate news on current events and helping people stay informed. However, in recent years, many people have lost trust in it due to the growing prevalence of political bias.

Political bias in the media takes many forms – including the way that stories are reported, the selection of the stories covered, and the spread of fake news and misinformation. It is especially widespread today in the United States, where partisan division has intensified greatly. News sources are increasingly reporting from a particular point of view on the political spectrum, and political language is becoming increasingly divided. People in different parties use different, carefully chosen words to discuss the same issues (Dahlgren, 2020). For example, Democrats describe immigrants without valid documentation as "undocumented immigrants" while Republicans describe them as "illegal aliens." The chosen words serve as a reflection of key differences in the values and goals of the two parties, which poses a significant problem as media and language become increasing partisan

(Thompson, 2016). Many people consume news from a few select sources, and as a result, they are heavily influenced by media biases and exposed to a narrower range of beliefs. This exacerbates political polarization – ultimately harming society as a whole.

Existing Work In order to promote greater transparency and restore trust in the media, it is important to detect and analyze the presence of partisan bias in the news. With the advancement of natural language processing (NLP), many people have started to apply text classification models for this task. Many previous approaches focus on simpler classification tasks such as identifying the presence of bias in the news, classifying text into left or right-leaning categories, and classifying news from a limited number of sources (Gentzkow et al., 2019; Naredla and Adedoyin, 2022). Early approaches used traditional machine learning methods using constructed features, and more recently, people have had more success with modern deep learning techniques such as CNNs, LSTMs, and Transformer-based models. However, these methods do not perform as well on more complex tasks such as classifying more precise degrees of bias or classifying longer news articles from a wide range of sources.

Project Overview In our project, we aim to address these shortcomings by classifying a wide variety of news articles into 5 categories based on the presence of political bias in their content: left, left-center, neutral, right-center, or right. We employ text classification techniques that have shown success in other application domains, yet remain largely unexplored in partisan bias detection. Given the key role of word choice in political discourse, we fine-tune a BERT encoder on political news articles to learn better contextualized word embeddings. We also use a Hierarchical Attention Transformer (HAT) to learn underlying hierarchical structure in news articles. By combining these two approaches, we expect our model to have a better understanding of both bias in political language and underlying document structure and be able to outperform pre-existing methods in political bias classification. We obtain impressive accuracy results and show that our fine-tuned embeddings are able to successfully distinguish between the word choices of the left and right ecosystems, allowing us to better understand the growing divide in partisan language.

3 Related Work

3.1 Text Classification

Classifying partisan bias in news articles is fundamentally a text classification task, and people have applied a wide range of text classification approaches, ranging from traditional machine learning methods to more modern neural network and Transformer-based approaches.

Traditional methods typically represent the original text at either the word-level or sentence-level by extracting features such as bag of words (BoW) and n-grams. These are then passed in as input into a linear classifier such as logistic regression or support vector machine (SVM) (Li et al., 2022). The performance of these linear models depend greatly on the quality of the extracted features, which are difficult to engineer manually.

Compared to traditional methods, deep learning approaches such as CNNs, RNNs, and LSTMs have been shown to better understand semantic relationships individually at the character-level, word-level, and sentence-level. For longer text, such as news articles, people have explored various approaches to take advantage of the underlying hierarchical structure of longer text. Hierarchical attention networks (HAN) were the first to learn this structure, by using self-attention to build sentence representations that capture the relationships between words and a document representation from the sentences (Yang et al., 2016).

Since then, additional hierarchical architectures have been proposed that take advantage of more recent, pre-trained Transformer-based models (Liu et al., 2022; Wu et al., 2021). For example, Hierarchical Attention Transformers (HATs) have outperformed both previous hierarchical models and efficient Transformer-based models, such as Longformer and Big Bird, in select long classification tasks, e.g. classifying biomedical summaries and legal contracts (Dai et al., 2022; Beltagy et al., 2020). However, they remain largely unexplored and have not been used to detect partisan bias in news articles, which have a strong underlying hierarchical structure that would be useful for a classifier to understand.

3.2 Contextualized Word Embeddings

In addition, despite the key importance of specific word choice in political conversations, the use of contextualized word embeddings remains largely unexplored in partisan bias detection (Hamborg, 2020). Most previous approaches use frequency counts or static embeddings such as word2vec and GloVe, which compactly map words to vectors but do not adapt to the context surrounding the words (Hitesh et al., 2019; Kishwar and Zafar, 2021).

On the other hand, contextualized word embeddings such as BERT, ELMo, and GPT-2 embed positional context and a general understanding of human language into the word representations (Peters et al., 2018; Devlin et al., 2018; Cohen and Gokaslan, 2020). They can easily be fine-tuned to significantly improve performance on a variety of downstream tasks such as movie review sentiment classification and question answering. By fine-tuning these embeddings on downstream tasks, models are able to leverage specific, domain knowledge on top of its broad knowledge about language to outperform previous approaches.

Within political text, word choice is particularly dependent on context and a key indicator of political leaning (Thompson, 2016). As a result, contextualized word embeddings have the potential to be extremely effective for partisan bias detection (Sezerer and Tekir, 2021).

4 Approach

Given a news article represented as a sequence of tokens, our model converts it to a corresponding sequence of BERT embeddings that have been fine-tuned to understand political language. The embeddings are used as input into a Hierarchical Attention Transformer (HAT) to produce an article encoding, which is then passed into a final classification layer to predict the article's bias (Figure 1). We describe our embeddings and encoder in greater detail in the following sections.

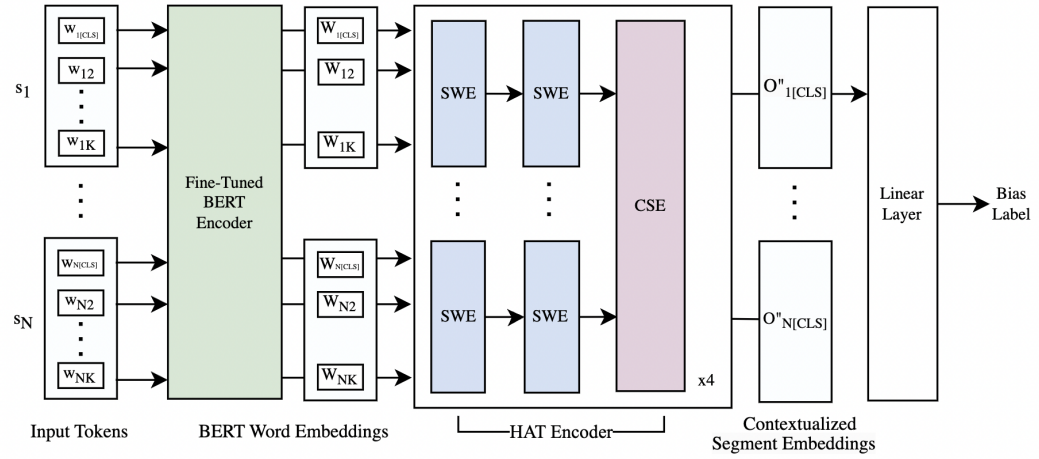

Figure 1: Our model first converts an input sequence of tokens, representing a news article, into a corresponding sequence of BERT embeddings fine-tuned on political text. These embeddings are then passed into a Hierarchical Attention Transformer (HAT) encoder, followed by a linear layer to predict the bias.

4.1 Word Embeddings: Fine-Tuned BERT

To convert input tokens to word embeddings, we use a BERT encoder that has been fine-tuned to classify political bias in news articles. Given a sequence of 512 tokens or less, BERT outputs a sequence of word embeddings by using bidirectional pretraining in masked language modeling and next-sentence prediction to learn contextual word embeddings (Devlin et al., 2018). To fine-tune BERT, we add a classification layer on top of the pre-trained BERT encoder, which takes in the final hidden state corresponding to the `[CLS]` token and predicts the bias label. We train the classifier on a dataset of news articles and use only the resulting embedding in our model. Through fine-tuning,

the BERT embeddings develop a better understanding of political language, in addition to general language.

4.2 Article Encoder: Hierarchical Attention Transformer (HAT)

To create an article representation from the fine-tuned BERT embeddings, we use a Hierarchical Attention Transformer (HAT). HAT encoders take advantage of the hierarchical structure of text documents by using self-attention at the word-level and sentence-level to create contextualized sentence representations from words and a contextualized article representation from the sentences (Yang et al., 2016).

Given a document as a sequence of word embeddings, HAT splits it into equally-sized sub-sequencesl Each subsequence is treated as a 'sentence', and each one is prepended with a special [CLS] token that serves as a segment-level representation This segmentation strategy is used over the original sentences, as it effectively preserves text structure while minimizing padding.

The HAT encoder itself consists of two types of encoder modules: segment-wise encoders (SWEs) and cross-segment encoders (CSEs). A SWE is a shared Transformer block that processes each segment individually to produce segment-level representations, while a CSE is a single Transformer block that processes all segments to produce a document representation.

In particular, for each sequence of word embeddings, SWE produces a corresponding sequence of contextualized word embeddings and a position embedding. The resulting contextualized $[\texttt{CLS}]'$ tokens are segment-level representations that capture local context within the segments. A CSE takes in all contextualized segment-level representations $[\texttt{CLS}]'$ and produces contextualized segment representations $[\texttt{CLS}]''$ and a single cross-segment position embedding (Figure 2). The first contextualized segment representation $[\texttt{CLS}]''$ is treated as the document vector, which is the final output of the HAT encoder.

These SWE and CSE blocks can be layered in any order, but the highest performing pattern interleaves the two by repeatedly stacking 2 SWE encoders followed by 1 CSE encoder (Figure 2). In our model, we use this pattern for our HAT encoder, and then we add a final linear layer on top of the HAT encoder to predicts the bias label given the article representation output from HAT.

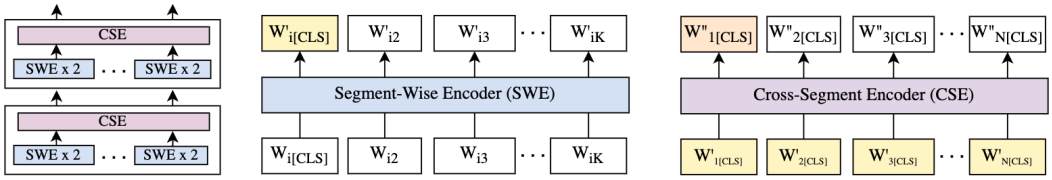

Figure 2: (left) Interleaving pattern used in our model of 2 SWE encoders followed by a CSE encoder. (middle and right) Visualization of the inputs and outputs of the SWE and CSE.

4.3 Baselines and Ablations

We compare our model against various baselines, including linear baselines, SVM, and random forest. For our linear baselines, we implemented and trained multiclass logistic regression models from scratch, with bag-of-words or n-grams features:

- Bag-of-Words: We use the 5,000 most frequent word tokens in the training corpus to construct a vocabulary, excluding stop words. For each article, we create a BOW-representation by counting the occurrences of each token, which is passed in as input to the linear model.
- n-grams: For a given n, we used the 5,000 most frequent n-grams as features.

In total, we had 5 linear baselines: BOW, 2-grams, 3-grams, 4-grams, and 5-grams. We also fit a non-linear SVM with a radial basis function kernel and a random forest classifier to perform this task.

In addition to these baselines, we also ran a fine-tuned BERT classifier and a fine-tuned HAT classifier to use as ablation studies for the evaluation of our final model.

5 Experiments

5.1 Dataset

We use the SemEval-2019 Hyperpartisan News Detection Dataset, which is publicly available on HuggingFace (Kiesel et al., 2019). The dataset contains $1,200,000$ news articles collected from 158 media outlets, and each article has a headline and partisan bias label (left, left-center, neutral, right-center, right).

To pre-process our data, we removed the HTML tags from all of the news articles and then concatenated each article's headline before its content. Since BERT takes in at most 512 tokens, we limited each sample down to 512 tokens using the BERT WordPiece tokenizer and saved the list of tokens. We then split our data using a 70-15-15 train/val/test split.

5.2 Evaluation Metrics

To evaluate our model, we use accuracy and macro-weighted precision, recall and F1-score. These were the metrics used by the SemEval-2019 Challenge, so using these metrics allows us to compare our models against their results.

5.3 Experimental details

Model Implementation To implement our model with the fine-tuned BERT embeddings, we used the official HAT implementation available on HuggingFace and replaced the pre-trained HAT embedding block with our fine-tuned BERT encoder.

We created two model variants (*MiniPartisanHAT* and *PartisanHAT*) using different versions of HAT, which have the same model architecture but different numbers of Transformer blocks. *MiniPartisanHAT* uses their miniature version (*MiniHAT-I3*), which has 12 Transformer blocks with 256 hidden units and 4 attention heads each, for a total of 17.7M parameters. On the other hand, *PartisanHAT* uses the full version (*HAT*), which is a much larger architecture with 16 Transformer blocks and 152M parameters.

Given the significant difference in the number of parameters, we chose to create two model variants to better explore the effectiveness of our fine-tuned encoder in either speeding up training or improving the model's understanding of political language.

Fine-tuning BERT We used the pre-trained BERT available on HuggingFace with an additional classification head and then fine-tuned it on our training set to classify partisan bias. We performed hyperparameter tuning on the learning rate and weight decay and used the default HuggingFace values for the other hyper-parameters. Ultimately, we found that weight decay did not correlate with validation loss, so we trained our classifier with a lr=5e-5 and weight_decay=0. We used 2 epochs as we found it sufficient for model convergence and batch_size=16 due to memory limitations.

Training the Model To train our full model variants, we froze the BERT encoder so that the embeddings would remain fixed, and then trained the remaining HAT encoder and classification layer. We tuned the learning rate and weight decay, and for the remaining hyper-parameters, we used the same values that were used to pretrain HAT initially.

After hyper-parameter tuning, we trained both model variants for 2 epochs with batch_size=16, lr=1.2e-4, and an Adam optimizer with $\beta = (0.9, 0.999), \epsilon$=1e-8. For *MiniPartisanHAT*, we used weight_decay=0.01 and gradient_accumulation_steps=4, while for *PartisanHAT*, we used weight_decay=0 and gradient_accumulation_steps=8.

5.4 Results

Our final models combining finetuned BERT embeddings with a HAT encoder achieved impressive accuracies of 0.947 for *MiniPartisanHAT* and 0.937 *PartisanHAT* (Table 1). These results were a significant improvement over our baseline models as we expected, and it even outperformed the top accuracy result of 0.706 from the official SemEval-2019 Hyperpartisan News Detection Challenge.

Despite this improvement, our results were surprising in two ways. First, we expected *PartisanHAT* to outperform *MiniPartisanHAT* due to its larger size and computational complexity. This was the case during training, as our models using the full HAT encoders had slightly higher training accuracies than the miniHAT encoders. However, given the higher performance of *MiniPartisanHAT* over *PartisanHAT* and *MiniHAT* over *HAT* on the test set, we believe that the models using the full HAT encoder were subject to some degree of overfitting. The original HAT encoder was pre-trained on text up to 4096 tokens, while MiniHAT was on text up to 1024 tokens. Since each sample in our dataset was limited to 512 tokens, we suspect that MiniHAT was more suitable for our data, and our data was not complex enough to utilize the full capacity of the full HAT architecture.

Second, our models did not perform as well as we expected in comparison to the fine-tuned BERT classifier and *MiniHAT* and *HAT* classifiers, which had extremely high accuracies of 0.972, 0.967, and 0.952 already. This may be due to the fact that we froze the BERT embeddings when feeding them to the HAT encoder. We chose to do this because we wanted to preserve the political context learned by our BERT embeddings, and it is also common practice to do so given our computational resource and time limitations. However, we suspect that by preventing the embeddings from updating during training, this limited our final model's ability to jointly leverage a combination of knowledge on political language and article structure.

Table 1: Evaluation Results on Test Set – This table shows the accuracy, precision, recall, and F1 of all the models we ran. We include the accuracies of the two models with the top accuracy and top F1 scores from the SemEval challenge, as an additional reference. Note that we only include the results of our highest performing linear baseline, BoW, and excluded the remaining n-gram results.

	Model	Accuracy	Precision	Recall	F1
SemEval	Top Accuracy	0.706	0.742	0.632	0.683
Baselines	BoW	0.464	0.446	0.392	0.418
	SVM	0.504	0.726	0.371	0.348
	Random Forest	0.592	0.673	0.457	0.455
Ablations	BERT	0.972	0.962	0.965	0.965
	MiniHAT	0.967	0.956	0.9580	0.957
	HAT	0.952	0.942	0.938	0.940
Models	**MiniPartisanHAT**	**0.947**	**0.935**	**0.933**	**0.934**
	PartisanHAT	**0.937**	**0.919**	**0.924**	**0.921**

6 Analysis

We visualize our fine-tuned BERT embeddings using principal component analysis (PCA) to gain additional insight into their effectiveness at understanding political language. For our analysis, we choose the most 500 most commonly used words from each bias class and exclude the ones that appear in multiple classes, ending up with 314 unique words.

As can be seen in Figure 3, which shows the PCA projections of these top words, our fine-tuned BERT does a better job at clustering together words that appear within each bias class than the original pre-trained BERT. The points within the fine-tuned BERT clusters are more dense, and the clusters themselves are also clearly distinct. This is an improvement over the clusters for the pretrained BERT, which overlap especially for the left-center, neutral, and right-center classes. This indicates that our model is in fact succeeding at learning which words are commonly used by left- and right-wing media as the corresponding embeddings of each bias class are close together.

Notice that although the fine-tuned BERT clusters in Figure 3 (left) are well-defined, the embeddings don't make it clear that left and right fall on opposite sides of a spectrum. The right, right-center, and left clusters are closer to each other in embedding space, with the neutral and left-center clusters far away from the others with no particular pattern. This may be one limitation of our final model.

In Figure 4, we do a more detailed analysis of our word embeddings by plotting a list of polarizing language to see whether they lie in the expected clusters. We obtain a list of highly partisan words used by left and right parties to discuss key political issues such as BLM, immigration, and abortion,

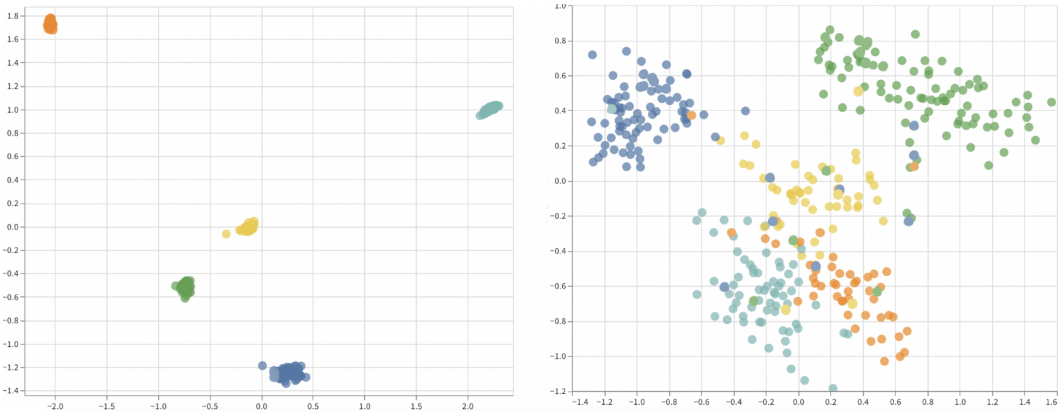

Figure 3: Top word embeddings for each bias class using our fine-tuned BERT (left) and original pre-trained BERT (right). The different colors represent different partisan bias classes, with blue=left, orange = left-center, light-blue = neutral, yellow = right-center, green = right.

from D'Alonzo and Tegmark (2021). This includes words such as "rally", "protesters", "asylum", and "undocumented" as words that commonly appear in media that leans politically left. We compare this against words like "riots", "mob", "infanticide", "illegal", and "aliens", which are more commonly used in right-wing media. These words embeddings all appeared in their respective bias clusters.

Furthermore, we observe these clusters more closely and see that words that are used to discuss a specific topic within each bias class are located more closely together. On the left, 'protestors' and 'rally' are close together in the embedding space. On the right, 'mob' and 'riots' are close, and 'illegal' and 'alien' are close as well. Given these observations, we see that the model was able to learn associations between word choice and political party.

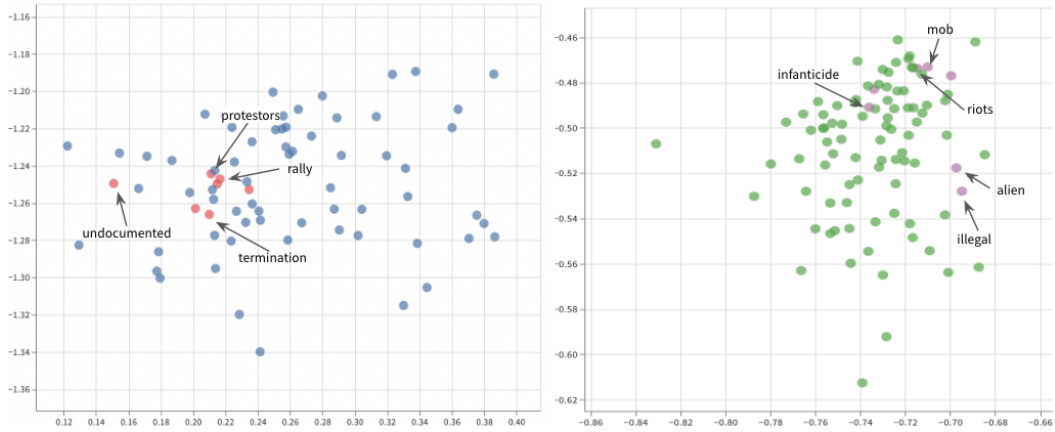

Figure 4: Finetuned BERT embeddings for select polarizing language used to discuss important political topics such as immigration, Black Lives Matter, and abortion. (left) A close-up of the left-leaning (blue) cluster with select words in red. (right) A close-up of the right-leaning (green) cluster with select words in purple.

7 Conclusion

In our project, we propose a model for partisan bias detection that uses a fine-tuned BERT encoder to leverage an understanding of political language on top of general language and a Hierarchical Attention Transformer (HAT) to embed knowledge of the underlying hierarchical structure of text documents. Individually, these NLP techniques have demonstrated success in improving text classifi-

cation results in other domains. Yet, prior to this work, they have remained largely unexplored in partisan bias detection.

In combining these two techniques, we expected our model to significantly outperform previous approaches in the task of classifying news articles into 5 political leaning categories. Although our model achieved high accuracies and outperformed traditional machine learning baselines, they did not outperform the individual BERT and HAT classifiers.

In our quantitative analysis, we use PCA to compare visualizations of our fine-tuned BERT embeddings against original BERT embeddings, which revealed that the fine-tuned BERT embeddings were able to effectively learn political context.

Given this, we suspect that our proposed model did not perform as well as expected because the HAT encoder was not as effective in combination with the BERT embeddings. In the future, given more computational resources and time, we would try training our model without freezing BERT embeddings as to allow them to update during training with the HAT encoder. We expect that this will enable our model to learn relationships between political words and the structure of news articles jointly, leading to a stronger classifier for partisan bias. Furthermore, we hope to explore additional variants of hierarchical attention, in order to find one that best captures the underlying structure of news articles together with fine-tuned contextualized embeddings like BERT.

Our work is still promising in that our model achieves impressive quantitative results in partisan bias classification, and the fine-tuned BERT embeddings are able to capture political context. By showing that these NLP techniques are able to perform well on partisan bias detection, we hope that this serves as a major step toward identifying partisan bias and increasing awareness about its presence. Furthermore, we hope that our initial results inspire further work in this area and motivate others to use powerful NLP techniques to address growing concerns about bias in the media and its detrimental effects to the future of society.

References

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *CoRR*, abs/2004.05150.

Vanya Cohen and Aaron Gokaslan. 2020. Opengpt-2: Open language models and implications of generated text. *XRDS*, 27(1):26–30.

Peter Dahlgren. 2020. Media, knowledge and trust: The deepening epistemic crisis of democracy. *The Liquefaction of Publicness*, page 20–27.

Xiang Dai, Ilias Chalkidis, Sune Darkner, and Desmond Elliott. 2022. Revisiting transformer-based models for long document classification.

Samantha D'Alonzo and Max Tegmark. 2021. Machine-learning media bias. *CoRR*, abs/2109.00024.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding.

Matthew Gentzkow, Jesse M Shapiro, and Matt Taddy. 2019. Measuring group differences in high-dimensional choices: method and application to congressional speech. *Econometrica*, 87(4):1307–1340.

Felix Hamborg. 2020. Media bias, the social sciences, and nlp: Automating frame analyses to identify bias by word choice and labeling. In *Annual Meeting of the Association for Computational Linguistics*.

MSR Hitesh, Vedhosi Vaibhav, Y.J Abhishek Kalki, Suraj Harsha Kamtam, and Santoshi Kumari. 2019. Real-time sentiment analysis of 2019 election tweets using word2vec and random forest model. In *2019 2nd International Conference on Intelligent Communication and Computational Techniques (ICCT)*, pages 146–151.

Johannes Kiesel, Maria Mestre, Rishabh Shukla, Emmanuel Vincent, Payam Adineh, David Corney, Benno Stein, and Martin Potthast. 2019. SemEval-2019 task 4: Hyperpartisan news detection. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 829–839, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Azka Kishwar and Adeel Zafar. 2021. Predicting fake news using glove and bert embeddings. In *2021 6th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM)*, pages 1–6.

Qian Li, Hao Peng, Jianxin Li, Congying Xia, Renyu Yang, Lichao Sun, Philip S. Yu, and Lifang He. 2022. A survey on text classification: From traditional to deep learning. *ACM Trans. Intell. Syst. Technol.*, 13(2).

Yang Liu, Jiaxiang Liu, Li Chen, Yuxiang Lu, Shikun Feng, Zhida Feng, Yu Sun, Hao Tian, Hua Wu, and Haifeng Wang. 2022. Ernie-sparse: Learning hierarchical efficient transformer through regularized self-attention. *arXiv preprint arXiv:2203.12276*.

Navakanth Reddy Naredla and Festus Adedoyin. 2022. Detection of hyperpartisan news articles using natural language processing techniques. *International Journal of Information Management Data Insights*, 2.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations.

Erhan Sezerer and Selma Tekir. 2021. A survey on neural word embeddings. *CoRR*, abs/2110.01804.

Derek Thompson. 2016. Why democrats and republicans speak different languages. literally.

Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. 2021. Hi-transformer: Hierarchical interactive transformer for efficient and effective long document modeling.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489, San Diego, California. Association for Computational Linguistics.