Political Leaning in News with BERT

Janet Chen, Ilseop Lee, Jahnavi Maddhuri, Alejandro Paredes

Abstract

This paper explores the application of BERT-based architectures for document classification of political leanings, focusing on DistilBERT for its versatility and lightweight design while maintaining performance comparable to larger models. The study compares three methods: Naive Bayes, refined fully connected (FC) layers, and Low-Rank Adaptation (LoRA). Naive Bayes serves as a baseline, highlighting the superior performance of transformer-based approaches. Enhancements to FC layers, incorporating cosine similarity and neural networks, provided incremental improvements and benchmarks. LoRA, integrated into the DistilBERT architecture, delivered the best results by optimizing attention mechanisms with trainable query layers and improved FC layers.

Experimental results demonstrate the effectiveness of these approaches. On the 2017 dataset, DistilBERT with cosine similarity achieved an accuracy of 0.45, while the method using a neural network fully connected layer model improved it to 0.63. However, LoRA significantly outperformed both, achieving 0.91 accuracy with precision, recall, and F1-scores of 0.9083, 0.9074, and 0.9066, respectively. On the 2019 dataset, while distilBERT with neural networks FC layer model's accuracy dropped to 0.4162, distilBERT+LoRA maintained robust performance with 0.7942 accuracy.

These results highlight the adaptability of BERT-based models to varying data conditions and emphasize LoRA's role in optimizing classification tasks, even with data drift across years. Future work will explore headline-specific performance challenges, alternative architectures, and advanced fine-tuning techniques to further improve classification accuracy.

1. Introduction

Media today covers issues in such a subtle manner that special attention is required to fully understand nuanced word choices when reading the news. While objectivity and unbiased reporting are a main goal of journalism, scholars argue that media tends to display clear ideological biases. This is evident in the recent coverage of the presidential election, reflecting more and more the importance of analyzing the political ideology of the media, as a framing of topics can significantly influence policy as noted by Dardis et al. (2008) [3].

While manual classification may be the most accurate procedure to tackle this problem, it poses a challenge when scaled up, therefore, extensive research work has been done on developing advanced computational models for automatically detecting political ideology, ultimately classifying news based on their political leanings (Iyyer M., et al, 2014) [4].

There have been prior research efforts to detect ideological biases in news articles and documents (Gerrish and Blei, 2011) by applying various algorithms to understand the subtleties and tone of language. Many of these papers propose structures that try to capture the semantic analysis

contained in the body of news (Iyyer M., et al, 2014) [4], optimizing the analysis within the tone of certain documents and using specific keywords to demarcate political leaning – this differs from the traditional document classification task based on the author (Klebanov, B. B. et al. 2010).

Unlike purely political texts, narratives contained in news introduce bias in very subtle ways, emphasizing other aspects of the report. The task of capturing complex sequential and semantic relationships in sentences and paragraphs has led to research of different neural network architectures. Work related to word representation such as those introduced by word2vec (Mikolov et al., 2013) [1], played a foundational role in capturing semantics in a vector space for detecting political leaning (Kulkarni et al., 2018) [2].

Furthermore, the development of large neural networks that can capture complex relationships within text has expanded the scope of analysis of these matters. A breakthrough in the field of NLP was the concept of attention mechanisms, which led to the development of transformer-based models such as Bidirectional Encoders Representations for Transformers (BERT) (Devlin et al., 2018). By leveraging self-attention, these models enable the capture of long-range dependencies and contextual relationships within text, significantly improving performance in text classification tasks.

Building on this insight we propose an implementation of BERT for document classification based on political leaning. By utilizing the powerful architecture and available large pre-trained models, our contributions aim to provide a new application for this task through:

1. Using naive Bayes as a base-level model to compare the performance of BERT across different configurations.
2. Proposing changes in the structure of the fully connected layer, using cosine similarity and neural networks to evaluate the embeddings of the model directly.
3. Analyzing the performance of the document classification task using Efficient Low-Rank Adaptation of Large Models (LoRA+) with BERT, analyzing the clear advantages of adding query and key layers in the initial configuration of BERT.
4. Comparing the performance of the model in different configurations of the data given that the model has been proven efficient classifying even small sentences for sentiment analysis.
5. Evaluating the data drift present in news (as for different years it is expected that news has more clear political leaning, especially in years prior elections) and evaluating metrics of the model in distinct snapshots of time.

2. Related work

Document classification, a foundational task in NLP, has seen remarkable advancements with the application of BERT-based models, such as DocBERT (Adhikari et al., 2020) [22], which was the first to successfully apply BERT for document classification. Despite the typical challenges of longer documents and syntactic variability, DocBERT demonstrated that BERT's contextual understanding could be effectively applied to documents with multiple labels. The authors addressed the computational challenges by distilling knowledge from larger BERT models into smaller, more efficient models, achieving BERT-base performance with a significant reduction in

model size. This work set a new baseline for document classification, showing that BERT could be used in more diverse contexts than originally thought.

Another notable application of BERT is in the domain of political leaning detection, where Retweet-BERT (Jiang et al., 2020) was introduced to estimate the political leanings of social media users based on both linguistic features and network structures. The Retweet-BERT model integrates information from Twitter's retweet network, exploiting the homophily effect among users with similar ideologies. This work illustrates how language models, in conjunction with network data, can be used to analyze political biases in social media platforms.

Similarly, the problem of political leaning detection has been explored on YouTube with a BERT-based model proposed by AlDahoul and Rahwan (2021) [8]. The paper addresses a classifier that detects the political leaning of YouTube videos based solely on their titles, a task that had not been adequately addressed prior to their work. Their classifier, trained on a large dataset of 10 million YouTube video titles, achieved high performance, with an accuracy of 75% and an F1 score of 77%. This research demonstrates the versatility of BERT-based models in identifying political leanings not only in textual data but also in multimedia contexts such as social media platforms.

As demonstrated by both Retweet-BERT and DocBERT, the ability to fine-tune models such as distilBERT for specific tasks—whether document classification or political bias detection—has significantly advanced the field of NLP. The ongoing research in this area underscores the growing importance of developing lighter, more efficient models capable of operating within resource-constrained environments, while still achieving state-of-the-art performance.

3. Dataset

The data we used for our implementations was the POLUSA Dataset: 0.9M Political News Articles Balanced by Time and Outlet Popularity provided by Zonodo [17]. This dataset contains information about nearly 0.9 million political news articles. Our features of interest included "body," containing the news article's entire text, "political_leaning," which classifies the news article's political leniency and "headline," which contains the headline of the article. The articles ranged from various sources such as CNN, ABC, and Reuters. The political leniency has four distinct classifications: "LEFT," "RIGHT," "CENTER," and "UNDEFINED." Below are tables depicting the distribution of the different classes and news sources in the dataset:

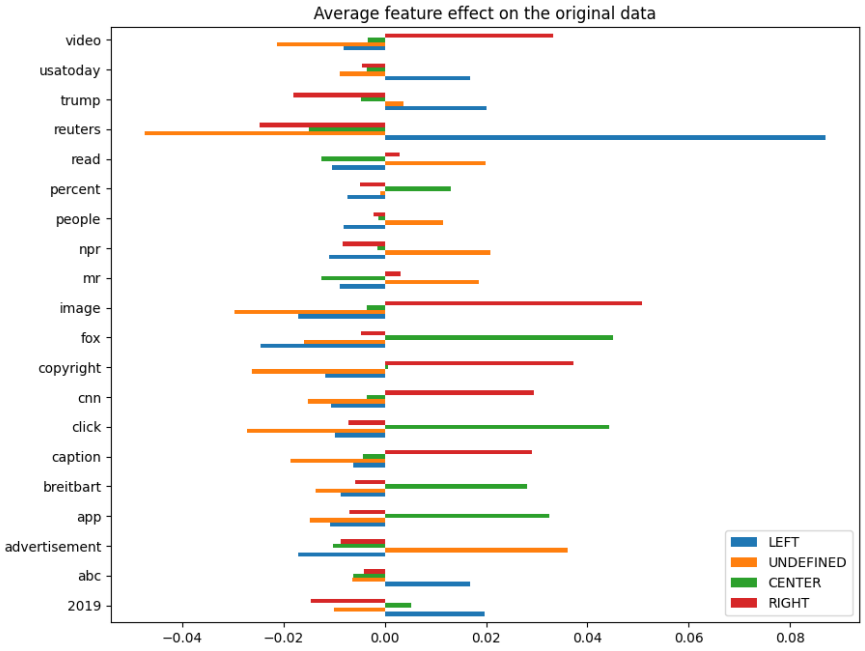
political_leaning	count
CENTER	47343
LEFT	42599
RIGHT	17028
UNDEFINED	39748

	outlet	count
0	The Guardian	19464
1	Reuters	19072
2	ABC News	17120
3	CNN	15878
4	Breitbart	10710
5	Chicago Tribune	10124
6	Los Angeles Times	9399
7	NPR	7097
8	Fox News	6318
9	NBC News	6236
10	The Wall Street Journal	5781
11	Politico	5083
12	USA Today	3826
13	Slate	2931
14	BBC	2882
15	HuffPost	2868
16	CBS News	1089
17	The New York Times	840

The body and classification features were the main inputs to train our base models. Further, data ranged from 2017 to 2019, but utilizing the entire dataset would have required high computational power to train on. Hence, we used a subset of the 2017 data when training our models. During later iterations, we also tested and trained on limited classifications, and various years of data, and used headlines only to train a new model.

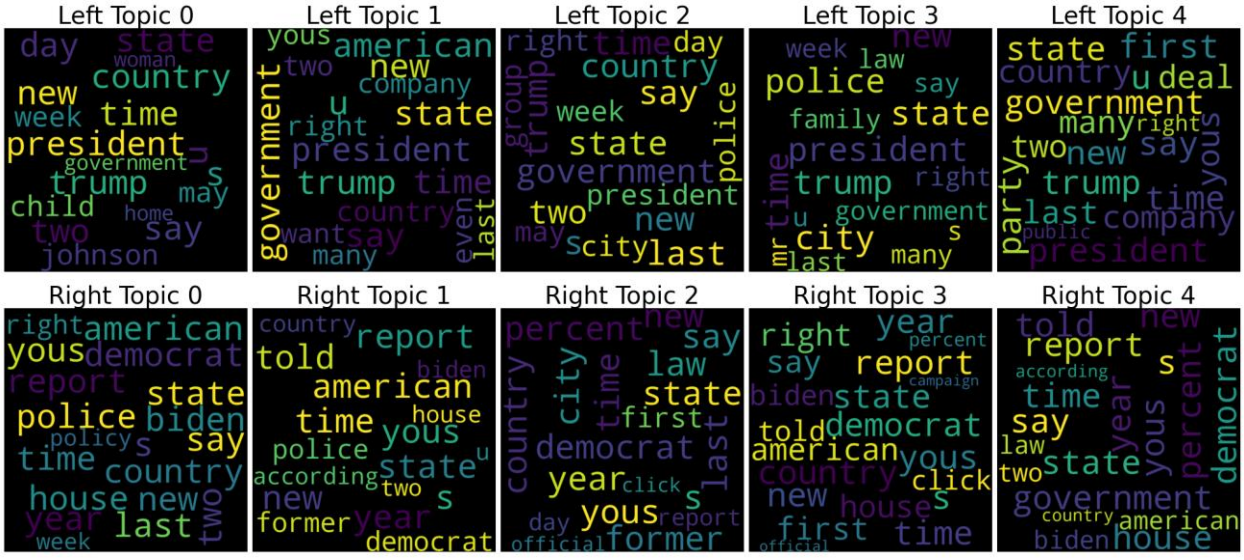
3.1. Topic Modeling for Exploratory Data Analysis

Leveraging a Term Frequency – Inverse Document Frequency (TF-IDF) analysis to perform classification for the political leaning classes, we saw the relevance of headers and footers within the body of the dataset. Words such as "trump" did not have a clear distinction between the classes of political leaning, while Reuters, Fox, and Click had a clear contrast between political leaning. Moreover, classifying political leaning using the author of the document as a reference could lead to the oversimplification of the analysis, and could impose a challenge when the document did not have a clear political leaning.



Furthermore, an initial exploration of the topics contained in the headline and body of news separated by class was performed using Latent Dirichlet Allocation (LDA). LDA represents each document as a finite mixture of latent topics, while each topic is characterized as a probability distribution over words. This approach provides an explicit and interpretable representation of documents in terms of their underlying topic distributions. Efficient inference methods, leveraging variational techniques and an Expectation-Maximization (EM) algorithm for empirical Bayes estimation, enable scalable application of LDA (Blei et al., 2003) [5].

For the analysis in hand, LDA is applied to the body of the news to determine the topics within each classified class, the distribution of those topics helped us to determine that some of the words present in the respective classes overlap.



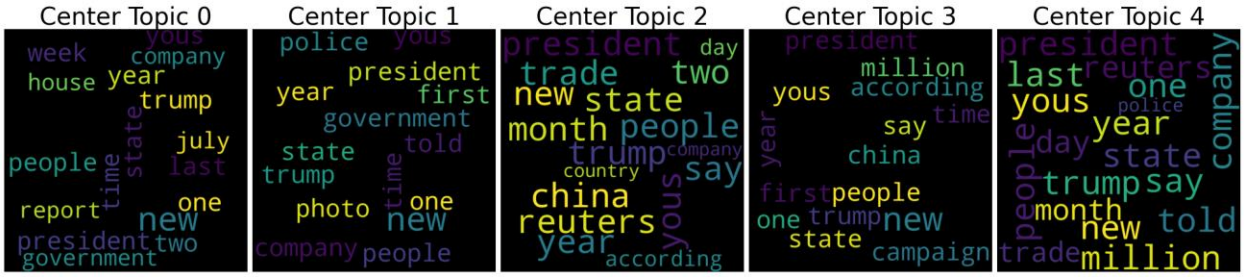


Figure 1.: Distribution of word's topics for documents classified by their political leaning.

Figure 1 reveals an overlap in certain words shared by left- and right-leaning documents, with examples including "country," "state," and "American." However, some topics are more distinctly differentiated. The center-leaning corpus is characterized by general themes such as "company", "million" and "trade". In contrast, the right-leaning corpus emphasizes partisan figures and policies, frequently referencing terms like "Biden" and "Democrat" suggesting a focus on ideological framing and political narratives. Meanwhile, the left-leaning corpus highlights terms related to societal structures and government actions, such as "government" and "Trump" indicating a critique-oriented perspective.

Thus, this analysis was used as a first insight, showing that the differentiation between topics is not clearly separated. This supports our approach of structuring a model capable of capturing the nuances used in the language in news.

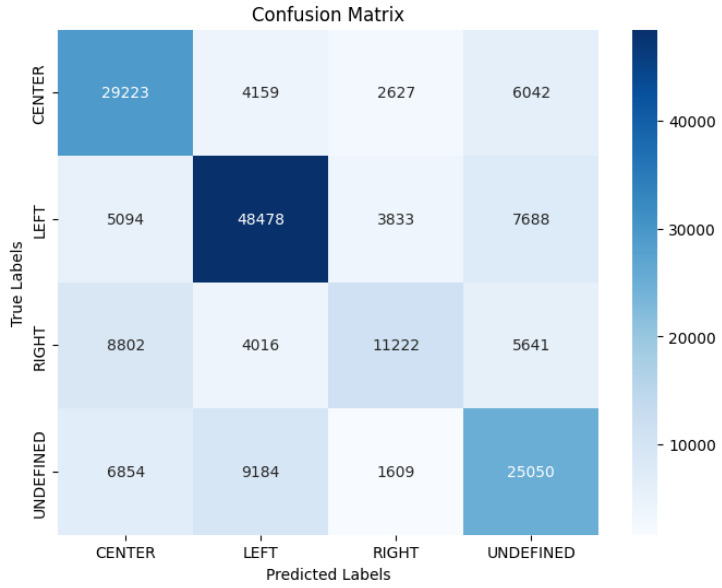
4. Methodology & Implementation

4.1. Naive Bayes Classifier

Naive Bayes for classification is the first model we trained in order to establish a base level of performance before attempting more complex models. As discussed in class, Naive Bayes has an interpretable framework that assumes conditional independence between the various classes. Though Naive Bayes is a simple model, empirically it has not only been an accurate, but also efficient model for text classification. According to Ting and Tsang from the Polytechnic University in Hong Kong, when compared against decision trees, neural networks and support vector machines, Naive Bayes outperformed these models in terms of model performance metrics such as F1 Score and accuracy and also was more computationally efficient [17].

Below are the results from our Naive Bayes Classifier. **Overall Accuracy: 0.7, overall f1-score 0.69.**

Class	Precision	Recall	F1 Score
LEFT	0.75	0.71	0.73
RIGHT	0.74	0.74	0.74
CENTER	0.61	0.61	0.61
UNDEFINED	0.65	0.69	0.67
MACRO AVERAGE	0.69	0.69	0.69



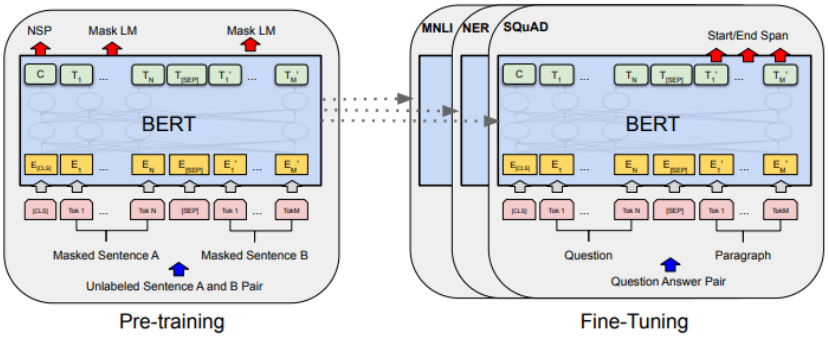
Overall, this model did a fair job in classifying the texts, where left-leaning and right-leaning classification was most accurately labeled by the classifier. Since Naive Bayes considers texts as a bag of words without any positional encoding, it largely depends on the distribution of word frequencies to determine the political leniency of the overall document. However, a large majority of the political jargon is common to all four classes as seen above in the LDA analysis. This leads to noise in the model, making classification more inaccurate. Further, though topics amongst different political groups may be similar, the specific sentiment towards the variety of topics would be an indicator of the political leniency of the article. However, because the model treats a text as a bag of words without enabling positional encoding, it is not able to capture the intricacies in these texts.

4.2. distilBERT

To compare to the results of Naive Bayes, we utilized the pre-trained distilBERT model. Attention mechanisms in neural networks have revolutionized the embedding of language enabling complex analysis that the transformers architecture offers. Bidirectional Encoder Representations from Transformers (BERT), is a transformers-based model that leverages pretraining large unlabeled text by joint conditioning on both the left and right context of the text. BERT can be fine-tuned with just one additional output layer to create cutting-edge models for a wide range of tasks, including sequence classification which is the task in hand (Devlin J., et al 2019) [10].

The authors argue that by using a masked language model as pretraining objective the model overcomes the constraint of unidirectionality of models such as ELMo (Peters et al., 2018a)[11] and Generative Pre-trained Transformer (Radford et al., 2018) [12] therefore understanding general context. The process to train this model consists of assigning random masks to some of the tokens from the input and then optimizing the model to predict the vocabulary id of the masked word based only on its surrounding context.

The authors detail the structure of the model building upon the architecture of self-attention, formulated in the paper "attention is all you need" (Vaswani et al., 2017) [13] a breakthrough in Natural Language processing research. The architecture of BERT consists of using the transformers architecture to pre-train general purpose models on large corpus of data to be used on different task later on with fine tuning.



Source: Devlin et al., (2018) "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding "pg 3.

Building on the foundation of the architecture of BERT, Sanh et al. (2019) introduced DistilBERT, a compact, general-purpose language representation model designed through the innovative process of knowledge distillation. This approach focuses on distilling the knowledge of larger models into a smaller model during the pre-training phase, resulting in a model that is significantly faster and more resource-efficient while retaining 97% of the original model's language understanding capabilities. The primary objective of DistilBERT is to create a model that is smaller, faster, lighter, and more cost-effective to pre-train, all while maintaining competitive performance levels across a wide range of NLP tasks.

Mentioned in the paper, knowledge distillation is a compression technique in which a compact model is trained to reproduce the behavior of a larger model. The authors explain that the embeddings and the pooler are removed while the number of layers is reduced by a factor of two. This restructuration of the model enabled applications where speed and compactness are important.

Component	**Details**
Embedding Layer	Word Embeddings (30522, 768), Position Embeddings (512, 768), LayerNorm, Dropout (0.1)
Transformer Layers	6 Transformer Blocks
Attention Mechanism	Self-Attention with Query, Key, Value (768), Linear Projections, Dropout (0.1)
Feedforward Network (FFN)	Linear Layers (768 to 3072 to 768), GELU Activation, Dropout (0.1)
Output Layer Normalization	LayerNorm (768)

For the implementation discussed in this document distilBert is used based on the pretrained checkpoint "distilbert-base-uncased" a distilled version of the BERT base model, the model was trained in BookCorpus, a dataset consisting of 11,038 unpublished books and English Wikipedia (Sahn et al., 2019) and made available in the Huggingface repository. The model takes as input a 768-dimensional vector consisting of tokenized inputs. DistilBERT uses a WordPiece tokenizer that tokenizes text into subword units, handling out-of-vocabulary words effectively by breaking them into smaller pieces. This tokenizer converts text into input tokens, which are then embedded into dense vector representations through the embedding layer. DistilBERT is then fine-tuned for classification tasks by adding a classification head on top of the pretrained model.

4.2.1 distilBERT embeddings and Cosine Similarity classification head

After training the distilBERT model, the first classification head we attempted was using cosine similarity on the embeddings produced by distilBERT. The methodology behind classification starts with capturing the embeddings from each body of text using distilBERT. Specifically, we captured the embedding from the last hidden state which is a 784-dimension vector. Then, for each label, we computed the mean embedding for that class. To predict the class of any embedding, we used the cosine similarity between the embedding of that body and the mean embedding for each class. Finally, a class was predicted for the body based on the label that produced the highest similarity score. Below is the function used to compute the similarity between two vectors represented by x.

$$\text{similarity} = \frac{x_1 \cdot x_2}{\max(\|x_1\|_2, \epsilon) \cdot \max(\|x_2\|_2, \epsilon)}$$

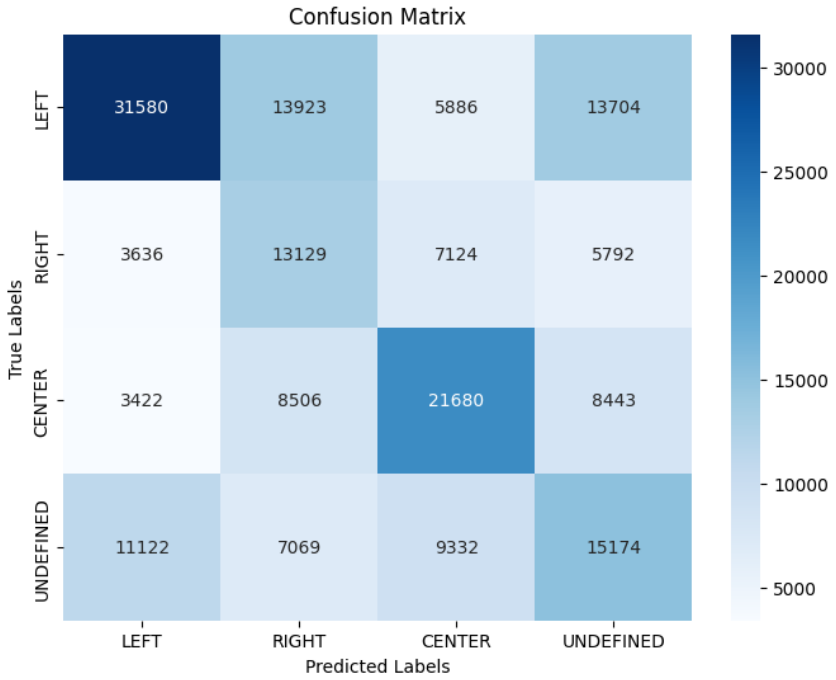
Overall Accuracy on 2017_2 (test set): 0.45

Class	Precision	Recall	F1 Score
LEFT	0.63	0.49	0.55
RIGHT	0.31	0.44	0.36
CENTER	0.49	0.52	0.50
UNDEFINED	0.35	0.36	0.35
MACRO AVERAGE	0.45	0.45	0.44

Holistically, distilBert combined with the cosine similarity classification did not perform highly accurately and was worse than the accuracy, precision, and recall achieved by our base Naive Bayes classifier. Of all four classes, the LEFT class was best classified where of all the texts that were predicted as left-leaning, 63% matched the observed classification of LEFT. Two reasons for this relatively higher accuracy could be the large representation of this class amongst the training data and the distinctiveness of the language. In the training dataset$, LEFT labels consisted of 42,599 records while RIGHT labels only consisted of 17,028 records. This imbalanced dataset allows LEFT to capture more patterns in the mean embeddings than RIGHT is able to, explaining its low model metrics. Further, there would be more linguistic features and vocabulary in the LEFT

labeled texts in comparison to the UNDEFINED. This would explain the low model metrics for the UNDEFINED class.

The distribution of the classified and misclassified labels on this test set is portrayed in the confusion matrix below. Similar to the pattern seen in the performance metrics, the LEFT label was best captured.



This methodology was not ideal for capturing the political sentiment of the news articles as it is incapable of capturing the complexities of the relationships. First, using the final hidden state instead of all states and using the mean embedding leads to information loss that could be leveraged by other models. Next, cosine similarity is not sophisticated enough to capture the complex relationship between the embeddings and the classes. Based on these results, we decided to test a different method for our final layer.

4.2.2. Fully connected layer with neural networks

In order to capture the complex distributions modeled by distilBERT we designed a fully connected layer to perform efficient document classification on documents. Leveraging distillation to perform document classification reduces computational expenses while maintaining similar levels to larger models (Adhikari et al., 2019) [22].

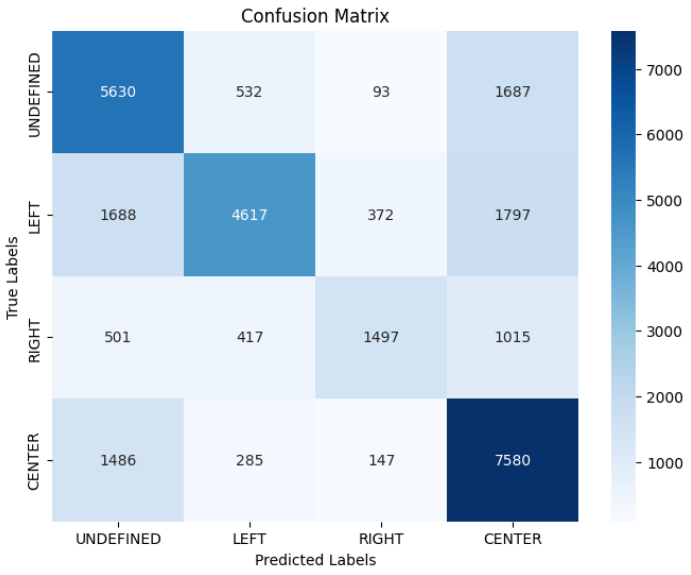
The dataset is split into 3 sets, a training loader composed of 80% of records, a validation set composed of 10% of records, and a test set composed of 10% of the information from 2017. The configuration used consists of a preprocessing step of standardizing the body of the news to lowercase words, performing de-contractions (transforming "you're" to "you are") removing non-ASCII characters (Chinese characters and other special characters not used in the English language) and filtering URLs and emojis with the help of the twitter-preprocessor library. The

imbalanced data was addressed by adding classweights to the CrossEntropyLoss function making the weights of the class RIGHT twice as the other classes.

The maximum length of the word sequence was limited to 512 adding padding, the train batch size as well as the validation batch size were of size 10. The loss function was set to CrossEntropyLoss for multiclass classification, the optimizer was set to Adam and a 1e-04 learning rate is chosen. A scheduler with a step size of 5 (activates after 5 epochs) and a gamma hyperparameter of 0.4 was used for learning weight decay. A Nvidia GTX 1660ti GPU was used to perform a training of over 10 epochs obtaining accuracy and loss evaluations every 500 steps. The configuration of this architecture is used in reference to implementation for similar tasks (AlDahoul et al., 2024) [8]

Component	**Details**
DistilBERT tokenizer	Input layer 1x512
DistilBERT encoder (embedding)	Word Embedding, Position encoding, 6 Transformer Blocks with self-Attention mechanism, Linear Layers (768 to 3072 to 768), GELU Activation, Dropout (0.1) (all weights and biases are frozen and do not get updated during training)
Linear layers	(768, 768), dropout 0.3, RELU Activation, (768, 1024)
Classifier	(1024, 4)
Output	Softmax layer

This architecture allowed the fully connected layer to learn complex nonlinear distributions, an improvement from only calculating the distance between an embedding and its class. The configuration leveraged the embeddings generated from distilBERT adapted for the classification task. The results from the testing set on 2017 data are shown in the table below. The overall accuracy is **0.6585** and the overall F1-score is **0.6534.**



By incorporating classification heads, such as a fully connected (dense) layer, into the DistilBERT architecture, the model was fine-tuned to adapt its pretrained language representations for the

specific classification task. The pretrained DistilBERT model produced rich, contextual embeddings that captured wide range of linguistic features, but these embeddings alone were not sufficient for direct classification. The addition of the classification head allowed the model to learn task-specific patterns by mapping the learned representations to the output space, which corresponds to the target classes. This process enabled the model to learn the decision boundaries necessary for accurate classification while leveraging the complex, high-level features encoded in the pretrained DistilBERT representations.

4.2.3 LoRA(Low-Rank Adaptation) Integration with distilBERT

In order to improve the results presented in the iterations described above, we used Low-Rank Adaptation (LoRA) to fine-tune. As mentioned previously, before BERT, transfer learning was limited as models often needed to be trained from scratch for each task, but after BERT, the widespread adoption of pre-trained models fine-tuned for various tasks made transfer learning a standard approach in both research and practical applications. [19][20]

As language models have grown exponentially in size, traditional fine-tuning methods that update all parameters have become increasingly inefficient and computationally expensive. For example, BERT in 2018 had 340 million parameters, while Turing-NLG in 2020 contained up to 170 billion parameters [21]. LoRA, or Low-Rank Adaptation, is an efficient alternative to full finetuning, improving speed and cost without significant performance sacrifices [15]. LoRA proves an effective finetuning technique as it does not require modifying all weights, but rather two adapter matrices, A and B, scaling them by a value α, and then updating a subset of the model's weights: $W' = W + \alpha(A \bullet B)$
Thus, LoRA operates by freezing the original weights and updating only the key components of the layer using low-rank matrices. The two adapter matrices are of low rank, constraining the rank of W and ultimately reducing the number of parameters to be tuned, thus improving space and time needs in the training process, as shown by Hu et al. (2021) [15].

This reduction in time and space does not compromise performance as the use of low-rank matrices in LoRA focuses its tuning of parameters on key attention layers, maximizing changes for task-specific adaptation and eliminating possible redundant updates on weights that contribute minimally to its task [15].

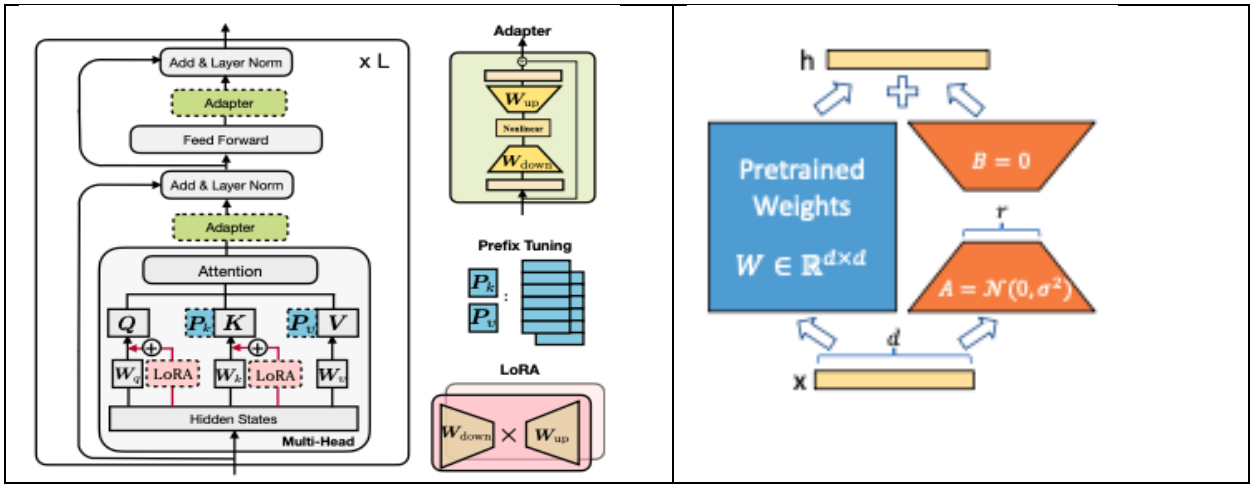
LoRA+, or Efficient Low-rank Adaptation, further improves on LoRA by addressing limitations in updating A and B. By allowing separate learning rates for A and B, efficiency in feature learning improves as does performance and speed [16]. Overall, LoRA's novel use of low-rank matrices to update certain layers for task-specific adaptations leading to time and speed improvements rendered it a promising candidate for our political leaning classification task.

	LoRA	LoRA+
Parameterization	Pretrained Weights $W \in \mathbb{R}^{n \times n}$ $+$ B $\times$ A	
Training	$A \leftarrow A - \eta \times G_A$ $B \leftarrow B - \eta \times G_B$	$A \leftarrow A - \eta \times G_A$ $B \leftarrow B - \lambda\eta \times G_B$ $\lambda \gg 1$

Source: Soufiane Hayou, Nikhil Ghosh, Bin Yu (2024) "LoRA+: Efficient Low Rank Adaptation of Large Models"

LoRA with distilBERT

In integrating LoRA with a pre-trained BERT model, LoRA is plugged into the Multi-Head Attention (MHA) layer. Hyperparameters such as the 'r', determine the dimensionality of the low-rank matrices A and B. This approach allows efficient updates to the Query projections in MHA. This approach allows LoRA to achieve both model performance and efficient fine-tuning, enabling seamless adaptation to downstream tasks such as classifying political leanings within documents. Note: in our implementation of LoRA, it is only applied to the Q vector. The figure below shows LoRA and its integration with BERT:



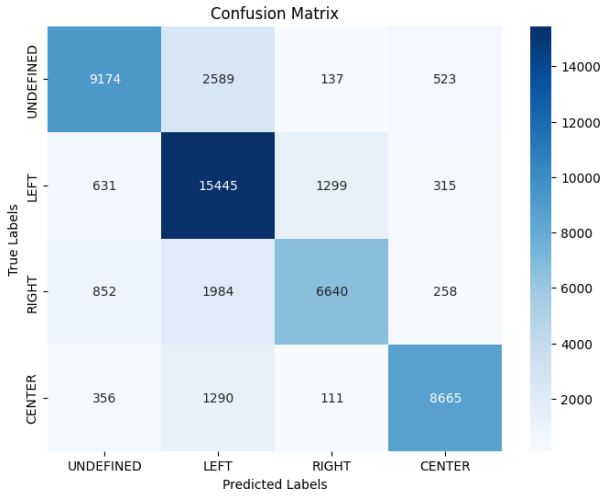
Sources: (left) Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, Graham Neubig(2022), Towards a unified view of parameter-efficient transfer learning, p.3
(right) Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen (2021). LoRA: Low-Rank Adaptation of Large Language Models. P.2

Table 1: Architecture of distilBERT-LoRA fine-tuned model

Layer	**Hyperparameters**	**Role**
BERT Tokenizer	Max length: 512, Padding: True	Converts raw text into tokenized input for the model
DistilBERT Preprocess	Word embeddings (30522x768) Layer Norm	Generates contextualized embeddings from text
Transformer Encoder	6 blocks with multi-head Attention & FFN	Encodes token relationships and contextual information
- Multi-Head Attention	12 heads, LoRA on Query (q_lin), Rank: 4	Prevents overfitting by deactivating random neurons.
Feedforward Network	Hidden size 3,072, Activation: GELU	Processes attention outputs into deeper representations.
- LayerNorm	eps=1e-12	Normalizes outputs for stable training
Pre-Classifier Layer	Dense: 768 -> 768	Prepares feature for final classification
Classifier Layer	Dense :768 -> 4	Maps features to class probability probabilities.
Dropout	Dropout: 0.1(embeddings), 0.2(final)	Prevents overfitting by deactivating random neurons.

The key innovation in this fine-tuning process was the integration of LoRA through the LoraConfig class. In this case, the target module was set to q_lin (query projection in the attention heads). The LoRA configuration included a rank r of 4, a scaling factor (lora_alpha) of 32, and a dropout probability of 0.01 to introduce regularization. This allowed the model to efficiently adapt to the classification task while reducing the number of parameters and computational costs compared to standard fine-tuning.

For the training procedure, the learning rate was set to 1e-3 with a batch size of 10 and a total of 10 epochs. The training was handled using the TrainingArguments class, with an evaluation strategy based on the completion of each epoch and saving the best model at the end. The model was trained with a weight decay of 0.01 to prevent overfitting. The overall accuracy for this model is 0.91 and F1-Score 0.9066



Overall Prediction Test Results

Incorporating Low-Rank Adaptation (LoRA) by appending trainable layers to the attention heads of DistilBERT has been shown to improve the model's ability to capture complex patterns, enhancing its performance for the specific classification task at hand. By optimizing the attention mechanism with LoRA, the model's capacity to adapt and refine its predictions was significantly increased. This approach led to improved accuracy and a better understanding of intricate relationships within the input data. The prediction results on the test set of 2017_1 and a test set of 2019_1, shown below, demonstrate the effectiveness of this technique. The metrics clearly indicate that the inclusion of LoRA for fine-tuning not only optimizes the model's behavior but also leads to a considerable improvement in performance, outperforming the baseline in terms of predictive accuracy.

Model	**Accuracy**	**Precision**	**Recall**	**F1-Score**
distilBERT + Cosine Similary on 2017_1	0.45	.445	.453	0.44
distilBERT + FC on 2017_1	0.63	0.55	0.61	0.57
distilBERT + FC on 2019_1	0.4162	0.33	0.41	0.36
distilBERT + LoRA on 2017_1	0.91	0.9083	0.9074	0.9066
distilBERT + LoRA on 2019_1	0.7942	0.7931	0.7942	0.7937

5. Training on data configurations

For the following sections, distilBERT+LoRA was used to perform various iterations on different configurations of data, using fewer tokens (taking only the headline or lead), performing semantical summarization (comprise data in lower levels), and using only two labels (LEFT and RIGHT) to analyze extremes. These analyses convey the empirical contrast of the initial

configuration of distilBERT having as a base level that the model beneficiates from large corpus as opposed to short sentences therefore reflecting the importance of some of the limitations of the model and addressing data drift while testing on different snapshots of data.

5.1. Using headlines, lead and summarization

The model was trained using different configurations of input data, the test consisted of training the model only in the headline mentioned in the dataset, the rationale to perform this training was to assess if the model could perform on similar levels of accuracy when given drastically less information. The minimum number of words in headlines in the 2017_1 dataset was 1 and a maximum of 40. In contrast, the body had a minimum number of words of 15 and a maximum of 16,698, the max_length of the configuration of distilBERT+LorRA allows up to 512 tokens.

Model	**Accuracy**	**Precision**	**Recall**	**F1-Score**
Headline only– tested on 2017_1 data	0.6250	0.6236	0.6250	0.6187
Lead only– tested on 2017_1 data	0.7908	0.7984	0.7908	0.7928
Summarization only – tested data extracted from 2017_1 data	0.5923	0.5857	0.5923	0.5862

With the empirical results shown above we can attest that distilBERT, particularly when augmented with LoRA, is optimized to leverage nuanced relationships present in large-scale, richly contextualized corpora. LoRA's low-rank adaptation techniques, appended to the attention heads, enhance the model's ability to learn fine-grained dependencies between textual elements, which are crucial for understanding subtle ideological cues in political text. However, when the input data is confined to headlines or summaries, the model is deprived of sufficient semantic depth and contextual richness. Headlines, by their nature, are succinct and often lack elaboration on the underlying topics or ideological stances. Similarly, summaries, while condensed, may omit critical details and implicit nuances necessary for accurately determining political leanings.

The inability of such configurations to provide a comprehensive context limits the model's capacity to capture complex, latent patterns that distinguish various political ideologies. Political leaning classification often involves decoding subtleties in rhetoric, framing, and lexical choices, which are better represented in extended textual data. Consequently, the suboptimal performance observed with headline-only and summarization-only datasets illustrates the necessity of utilizing more expansive and detailed textual inputs to enable DistilBERT+LoRA to fully leverage its architecture and achieve state-of-the-art performance. These findings emphasize the importance of curating training data with sufficient scope and detail to align with the model's capacity for nuanced understanding.

5.2 Filtering Left-Right classes only

To further understand and explore the scope of distilBERT and LoRA's performance, the model was trained on a subset of the data with 'LEFT' and 'RIGHT' tags only to investigate whether forcing the model to make extreme decisions would improve or reduce model performance. By eliminating the ambiguity of 'CENTER' and 'UNDEFINED' labels and reducing the decision boundary to a binary classification, the model would be forced to learn and classify left and right which we hypothesized would improve accuracy and overall performance.

5.2.1 Body

When trained on the body of news sources, this subset of left-right-only data saw model performance improve from the base implementation. The model achieved an accuracy of 0.92 when tested on its 2017 test dataset during the train-test split process and an F1-score of 0.86 as shown in the table below. This confirms our hypothesis that forcing the model to classify extremes resulted in improved performance. In order to ensure the model was not overfit, we tested on a dataset from 2019 as well, which introduced possible differences in news topics as 2019 was approaching the 2020 election cycle. Results again in the table show a high accuracy and F1-score.

We attribute this improvement in model performance on left-right only labels to three potential factors: (a) simpler decision boundary, (b) less class ambiguity, and (c) data imbalance.

(a) Simpler decision boundary: the task becomes less complex and is essentially binary classification. For BERT, this means focusing on polar extremes which results in better separability in the learned embedding space and previous research has shown BERT performs better in binary classification compared to multiclass classification tasks [23]. With more classes introduced like "CENTER", the output layer must learn more complex boundaries which could increase the risk of misclassification.
(b) Class ambiguity: including classes like "CENTER" and "UNDEFINED" which could semantically or contextually overlap with "RIGHT" or "LEFT" could make classification difficult and lead to lower performance. LoRA and its use of low-rank adaptations perform well for binary classes but may struggle with more nuanced and overlapping classes due to LoRA only adapting targeted parts of the pre-trained model and leaving the original weights frozen [24]. Thus, tuning and adaptation are likely focused on the most extreme or salient features such as "LEFT" and "RIGHT" rather than ambiguous classes.
(c) Data imbalance: from the label distributions in the dataset, about 32% of the data is labeled as "CENTER" and about 27% is classified as "UNDEFINED". The large portion of the data being labeled as "CENTER" or "UNDEFINED" could lead to models over-labeling "CENTER" or "UNDEFINED" as an option, especially with class ambiguity. Thus, by eliminating those labels, the model may have had enhanced performance.

Overall, previous research has shown that trained models like BERT are designed to generalize well in tasks where the decision boundaries are strong and LoRA performance is improved with less ambiguity. These factors may explain why classification accuracy is best for "LEFT" and "RIGHT" only labels compared to four classes of labels.

5.2.2 Headlines

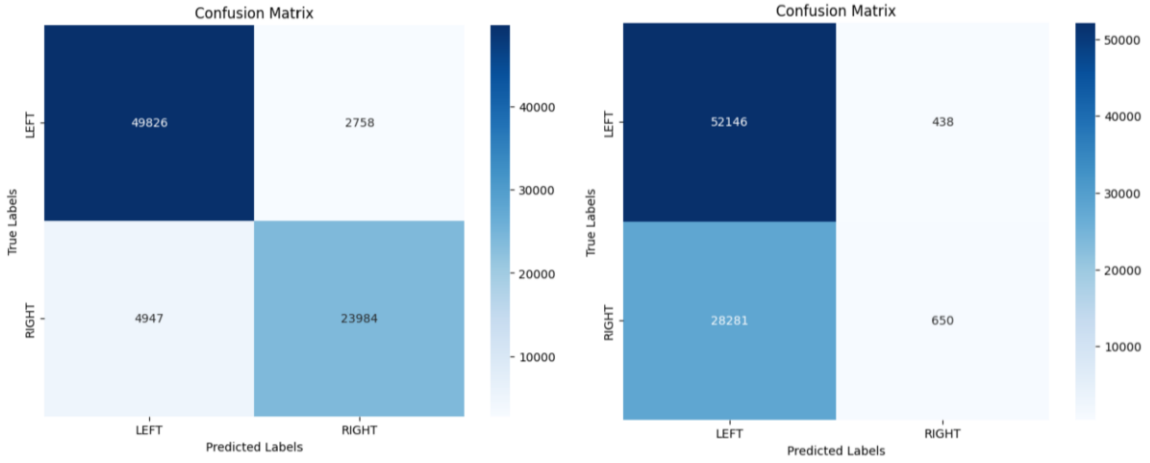
Echoing and reproducing the results from 5.2 comparing headline and body training, the "LEFT" and "RIGHT" only subset was also trained separately on headlines and body of the news articles. Similar results were seen in which accuracy and F1 score metrics were better for the full body text, reinforcing the idea that headlines lack the semantic representation and perhaps the amount of data to build a robust classification model for political leanings.

Table: Performance Metrics for left-right label only

Model	Accuracy	Precision	Recall	F1-Score
Full body – tested on 2017 data	0.9212	0.8929	0.8378	0.8645
Full body – tested on 2019 data	0.9055	0.9051	0.9055	0.9046
Headline Only – tested on 2017 data	0.8039	0.7259	0.5824	0.6463
Headline Only – tested on 2019 data	0.6477	0.6303	0.6477	0.5212

Note: "Left" was labeled as 0 and "Right" was labeled as 1

Confusion Matrix: Left-Right labels trained on body (left) and headline (right) text and tested on 2019 data



6. Predictions on test sets and overall comparison between models

A side-by-side comparison between models is shown below to evaluate the results for the different models used. The table presented below correspond to the accuracy for the sets listed. For the test set of 2017_1 the data has been partitioned from the one it was originally trained for the remainder datasets, it was performed the prediction with the corresponding trained model on the entire dataset.

Test set	Naive Bayes Classi-fier	Distil BERT+ Cosine Simi-	Distil BERT+ FC NN	Distil BERT+ LoRA	Distil BERT+ LoRA: Left/right	Distil BERT+ LoRA:	Distil BERT+ LoRA: headline

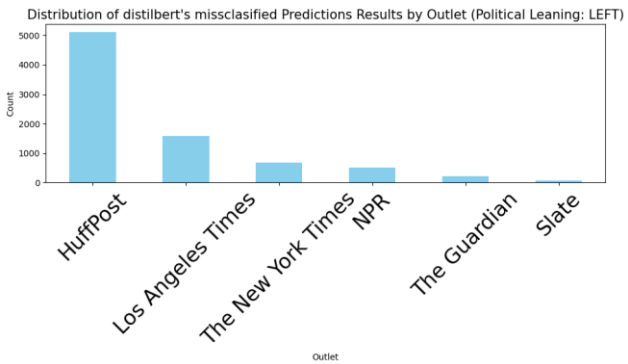
		larity			**only, body**	**left/right only, headline**	
2017_1 (146,728 records)	0.7	0.5	0.65	0.91	0.9212*	0.8039*	0.6250
2017_2 (179,520 records	0.64	0.45	0.5629	0.827			0.6165
2019_2 (52,935 records)	0.53	0.37	0.4162	0.794	0.9055	0.6477	0.6146

* The training dataset is an integration of 2017_1 and 2017_2

As seen above, the best performing model is DistilBERT+LoRA - left righ, this model takes only the classes right and left and makes a binary classification, since the model had more defined patterns to capture the difference between the political leaning it performs better on a different year as well. The best overall model with all the classes is DistilBERT+LoRA maintaining an accuracy of 0.79 for the test set 2019_1.

7. Incorrect classification case analysis

The concentration by outlet from the incorrect classification on the test set of 2019_1 brings some insights in regard of some patterns identified. We identified that for left-leaning documents, the outlet HuffPost was the principal news source that was misclassified. Inspecting their documents, we observed that the news source could have a specific tone in their news report such that the model did not correctly capture that behavior.



Using the following corpus as example, the correct classification would have been LEFT, however the prediction yielded UNDEFINED. We identified that the document references a quote from a government official which can be seen as referencing negatively to the officer.

“That's a doozy of a subheading, but we felt it was completely appropriate this week. It is a direct quote, from conservative (and "Never Trump") commentator Ana Navarro. During an interview with Wolf Blitzer, Navarro responded to Trump's recent tweetstorm attacking Mika Brzezinski by calling on Republicans to say to

Trump (either on television or personally) the following: "Listen, you crazy, lunatic, 70-year-old man-baby, stop it. You are now the president of United States, the commander-in-chief and you need to stop acting like a mean girl because we just won't take it." We've saved her entire rant for the talking points, because it is indeed worth reading in full; but because it was the most forceful pushback on Trump we heard all week, we thought it deserved headline status. Tell us what you really think, Ana! Of course, she wasn't the only conservative to chime in on Trump's petulance. Charles Krauthammer pulled no punches in his response either: "Presidents don't talk like this..."

8. Conclusions

Given that our model performed best when using the pre-trained distilBERT and applying LoRA for finetuning on the body text of our data, we can conclude that in terms of classification techniques, transformer-based technologies are the most robust.

Additionally, the models seem to work best in unambiguous, binary classification settings, such as when sub-setting for "LEFT" and "RIGHT" labels only, and when there is sufficient data to capture the nuances of sentiment in political leanings, such as when trained on the body versus headline text. The latter can also be seen in the higher accuracy of distilBERT and LoRA on "LEFT", "CENTER", and "UNDEFINED" than on "RIGHT" as the data had the least number of "RIGHT" labels. Thus, given the models perform worse on headline text and "RIGHT" labels only, it can also be reasonably concluded that overall, the accuracy of the model is dependent on the proportion of data.

Potential limitations of our implementation and decisions could include failing to address the imbalanced or uneven dataset in terms of political leaning labels. Additionally, all models were trained on 2017 datasets only and, other years may have contained different information, though our high accuracy testing on 2019 data likely addressed this. Moreover, there may exist other factors as to why headline text performance was worse than full body text performance beyond the amount of data which we could investigate more. For example, it could be that the body text contains distinct identifiers, such as news source, which the headline lacks and this could help the model in creating its decision boundary. Finally, experimenting with various parameters for LoRA such as changing the rank or applying LoRA to Q and K vectors was not explored.

Future studies and iterations could address some of these limitations by more in-depth investigation of the differences between headline predicted and actual labels to understand why our headline-only models are classifying incorrectly. We could also adjust LoRA's parameters and explore other pre-trained model options such as Roberta or distilRoberta.

Overall, while our classification of political leanings task showed that a combination of transformer-based pre-trained models with LoRA fine-tuning performs best on unambiguous and more frequent classes, more investigation can be done to optimize performance and further explain why and how these models' architectures are conducive to these tasks.

9. References

[1] Mikolov, T. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781, 3781*.

[2] Kulkarni, V., Ye, J., Skiena, S., & Wang, W. Y. (2018). Multi-view models for political ideology detection of news articles. *arXiv preprint arXiv:1809.03485*.

[3] Dardis, F. E., Baumgartner, F. R., Boydstun, A. E., De Boef, S., & Shen, F. (2008). Media framing of capital punishment and its impact on individuals' cognitive responses. *Mass Communication & Society*, *11*(2), 115-140.

[4] Iyyer, M., Enns, P., Boyd-Graber, J., & Resnik, P. (2014, June). Political ideology detection using recursive neural networks. In *Proceedings of the 52nd annual meeting of the Association for Computational Linguistics (volume 1: long papers)* (pp. 1113-1122).

[5] Gerrish, S. M., & Blei, D. M. (2011, October). Predicting legislative roll calls from text. In *Proceedings of the 28th International Conference on Machine Learning, ICML 2011*.

[6] Klebanov, B. B., Beigman, E., & Diermeier, D. (2010, July). Vocabulary choice as an indicator of perspective. In *Proceedings of the ACL 2010 conference short papers* (pp. 253-257).

[7] Devlin, J. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

[8] AlDahoul, N., Rahwan, T., & Zaki, Y. (2024). A Novel BERT-based Classifier to Detect Political Leaning of YouTube Videos based on their Titles. *arXiv preprint arXiv:2404.04261*.

[9] Martin, J. H. (2009). Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition. Pearson/Prentice Hall.

[10] Kenton, J. D. M. W. C., & Toutanova, L. K. (2019, June). Bert: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of naacL-HLT (Vol. 1, p. 2).

[11] Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep Contextualized Word Representations. *ArXiv, abs/1802.05365*.

[12] Radford, A. (2018). Improving language understanding by generative pre-training.

[13] Vaswani, A. (2017). Attention is all you need. Advances in Neural Information Processing Systems.

[14] Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. arXiv 2019. arXiv preprint arXiv:1910.01108.

[15] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen (2021). LoRA: Low Rank Adaptation of Large Models. arXiv:2106.09685.

[16] Soufiane Hayou, Nikhil Ghosh, Bin Yu (2024). LoRA+: Efficient Low Rank Adaptation of Large Models. arXiv:2402.12354.

[17] Kristiadi, Agustinus, Matthias Hein, and Philipp Hennig. "Being Bayesian, Even Just a Bit, Fixes Overconfidence in ReLU Networks." *Zenodo*, 2020, doi:10.5281/zenodo.3813664. Accessed 27 Nov. 2024.

[18] Rennie, Jason D. M., et al. "Is Naive Bayes a Good Classifier for Document Classification?" *ResearchGate*, 2003, www.researchgate.net/publication/266463703_Is_Naive_Bayes_a_Good_Classifier_for_Document_Classification. Accessed 27 Nov. 2024.

[19] Devlin et al., BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding (2018) arXiv

[20] Colin Raffel et al., Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer" (2019) arXiv

[21] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, Sylvain Gelly, Parameter-Efficient Transfer Learning for NLP. 2019, arXiv:1902.00751T

[22] Adhikari, A. (2019). DocBERT: BERT for Document Classification. arXiv preprint arXiv:1904.08398.

[23] Eang, C., & Lee, S. (2024). Improving the Accuracy and Effectiveness of Text Classification Based on the Integration of the Bert Model and a Recurrent Neural Network (RNN_Bert_Based). *Applied Sciences*, *14*(18), 8388. https://doi.org/10.3390/app14188388

[24] Yuren Mao, Yuhang Ge, Yijiang Fan, Wenyi Xu, Yu Mi, Zhonghao Hu, Yunjun Gao, A Survey on LoRA of Large Language Models (2024), arXiv:2407.11046

[25] Jiang, J., Ren, X., & Ferrara, E. (2023, June). Retweet-bert: political leaning detection using language features and information diffusion on social networks. In Proceedings of the international AAAI conference on web and social media (Vol. 17, pp. 459-469).