RESEARCH **Open Access**

PoLYTC: a novel BERT-based classifier to detect political leaning of YouTube videos based on their titles

Nouar AlDahoul[1], Talal Rahwan[1] and Yasir Zaki[1*]

*Correspondence:
yasir.zaki@nyu.edu

[1] Computer Science, New York University Abu Dhabi, Abu Dhabi, UAE

Abstract

Over two-thirds of the U.S. population uses YouTube, and a quarter of U.S. adults regularly receive their news from it. Despite the massive political content available on the platform, to date, no classifier has been proposed to classify the political leaning of YouTube videos. The only exception is a classifier that requires extensive information about each video (rather than just the title) and classifies the videos into just three classes (rather than the widely-used categorization into six classes). To fill this gap, "PoLYTC" (Political Leaning YouTube Classifier) is proposed to classify YouTube videos based on their titles into six political classes. PoLYTC utilizes a large language model, namely BERT, and is fine-tuned on a public dataset of 11.5 million YouTube videos. Experiments reveal that the proposed solution achieves high accuracy (75%) and high F1-score (77%), thereby outperforming the state of the art. To further validate the solution's classification performance, several videos were collected from numerous prominent news agencies' YouTube channels, such as Fox News and The New York Times, which have widely known political leanings. These videos were classified based on their titles, and the results have shown that, in the vast majority of cases, the predicted political leaning matches that of the news agency. PoLYTC can help YouTube users make informed decisions about which videos to watch and can help researchers analyze the political content on YouTube.

Keywords: YouTube, Political leaning, BERT classifier, PoLYTC

Introduction

The widespread use of the World Wide Web has led to a substantial increase in the number of adults who consume at least some of their news online, reaching nearly 90% of adults in the United States [1]. YouTube, one of the most popular websites on the World Wide Web, is rapidly growing its content, with more than 500 h of video uploaded every minute, amounting to a total of about 30,000 h of new content every hour [2]. Currently, more than two billion people use the platform, and *YouTube Shorts* alone have received 70 billion views to date, according to [3]. Politics is among the many topics covered by the platform. A quarter of adults in the U.S. regularly receive their news from YouTube, making it the second most popular online news source worldwide [4, 5].

Several studies have demonstrated political leaning and bias in the media, particularly news articles [6–10]. These studies proposed classifiers to predict bias using textual data extracted from headlines or content. In the context of YouTube, there have been numerous solutions aimed at categorizing videos into various classes [11–14]. These solutions focused on classifying news documents or video titles using conventional machine learning algorithms. One study used transformer-based embedding models [15], which have been shown to achieve state-of-the-art performance in multiple domains [16–19]. However, the authors utilized several features, including title, description, and tags. Additionally, they classify the videos into just three categories, namely far Right, far Left, and Center. As such, no classifier has been proposed to identify the six categories of political leaning of YouTube videos (Far-Right, Right, Anti-Woke, Center, Left, and Far-Left [20]) based solely on the videos' titles.

The capability of embedding models to learn left-to-right and right-to-left contexts and produce a meaningful representation has been a challenge for a long time. Google's BERT is a language model that addresses this challenge by learning a bidirectional representation. Having an effective representation or embedding of text is a key factor in building a highly accurate text classifier. BERT has shown superior performance as an embedding model for various classification purposes [16–19]. Language models require a large dataset to train on in order to avoid the problem of overfitting. Fortunately, in our context of classifying the political leaning of YouTube videos, a large dataset already exists, consisting of 11.5 million videos labeled based on their political leaning [20–22].

Previous works used traditional machine learning algorithms for embedding, such as TF-IDF [23], word2Vec [24] and GloVe [25]. However, these models cannot adequately find informative word representations from context [26], which could affect the classification accuracy. This problem can be found in several works, such as fake news detection [27], text sentiment analysis [28], and topic classification [29].

Several works have focused on detecting political leaning in newspaper articles [6], tweets [30], and Facebook [31]. However, none of the previous works use the titles of YouTube videos to classify them into six categories of political leaning. To fill this gap in the literature, three pre-trained text classifiers were examined, namely Word2Vec [32], Global Vectors for Word Representation (GloVe) [33], and Bidirectional Encoder Representations from Transformers (BERT) [34]. These classifiers were fine-tuned using the aforementioned dataset, where videos are pre-labeled into six classes, namely Far Left, Left, Center, Anti-Woke, Right, and Far Right.

The proposed approach was further validated by the video content of 15 prominent news channels whose political leaning is widely known. More specifically, five channels had a Left leaning, five had a Center leaning, and five had a Right leaning. Thousands of videos have been collected from each channel to extract titles along with the dates on which the videos were uploaded. The result of this evaluation confirms the ability of the proposed classifier to predict political leaning based on video titles.

The main contributions of this work are summarized as follows:

- This work proposes a fine-tuned BERT classifier, PoLYTC, that predicts the political leaning of YouTube videos, achieving higher accuracy and F1-score than state-of-the-art alternatives.

- PoLTYC is further validated with thousands of videos collected from 15 YouTube channels of prominent news agencies, the results of which confirm the classifier's high accuracy.
- Previous solutions classify YouTube videos into just three classes, namely, Left, Right, and Center. While this over-simplification makes the classification task easier, it disregards crucial differences between left and far-left, between right and far-right, and between center and anti-woke videos. PoLYTC overcomes this limitation by providing a more fine-grained classification.
- PoLYTC relies solely on video titles, which is far more practical than relying on a wide set of features such as video acoustics, comments, and meta-data, as was the case with previous solutions.

This paper is organized as follows: The "Related work" section summarizes the relevant literature. Section "Materials and methods" describes the dataset and discusses numerous text classification models, such as Word2Vec, GloVe, and BERT. The "Experimental pipeline" provides an overview of the different stages used in the experiments. Section "Experimental results" evaluates the different text classifiers. Finally, "Conclusion and future work" summarizes the work and discusses potential future directions.

Related work

Several research articles have examined the political leaning in media, focusing on various applications and use cases. One such application is algorithmic recommendations [35]. This study examined YouTube's recommendation algorithm in the context of U.S. politics to determine whether the algorithm is neutral or leans in a certain political direction. The authors found evidence that the recommendation algorithm is left-leaning, as it pulls users away from Far-Right content stronger than from Far-Left content. Another application in which the examination of political leaning can be helpful is the study of radical content consumption [20]. Here, the authors showed that the trends in video-based political news consumption are determined by various factors, the most important of which is individual preferences.

Perhaps the application most relevant to the context of our study is the prediction of political leaning in videos, which has been explored in numerous articles [6–10, 20]. Specifically, in [20], a binary random forest classifier consisting of 96 predictors was trained. To identify the political leaning of any given video, the authors utilize a feature engineering method by analyzing the web partisan score of news domains viewed by users before and after the video in question, as well as the political leaning of all videos watched within the same session. The authors also rely on user-level features, such as the individual's monthly consumption and web categories. In [6], the authors proposed a generalized SVD-modeling of phrase statistics to infer a leaning conditional probability distribution in a given newspaper article. In [7], Kulkarni et al. explore the possibility of using an article's title and link structure to predict any biases therein. The authors capture cues from both textual content and the network structure of news articles using a novel attention-based multi-view model. In [8], Li and Goldwasser demonstrate how social content could be utilized to improve bias prediction by using graph convolutional networks to encode a social network graph. The study of political bias has been extended to other languages such as German and Indian [9, 10]. More specifically, a dataset of German news articles labeled by a fine-grained set of labels was utilized for political bias classification [9]. The authors explored various feature extraction models, including bag-of-words, term-frequency times, inverse-document-frequency, and BERT, along with various classifiers, including logistic regression, naive Bayes, and random forest. Gangula et al. [10] analyzed news articles in the Indian language Telugu to detect political bias using 1329 headlines of articles. The authors compared several models, such as Convolutional Neural Network (CNN), Long Short-term Memory (LSTM), and attention network, and found that the latter model outperformed the other ones.

Recently, the BERT model has been used in several studies for the purpose of detecting political leaning. For example, the authors of [30] used BERT to study the political discourse on Twitter. The authors utilized the "RetweetBERT" model to identify the political leanings of Twitter users based on their profile descriptions. Similarly, the authors of [31] estimated the political leaning of U.S. adult Facebook users. To this end, they utilized DistillBERT—an externally trained classifier on Facebook content using text-based features and text extracted from images using Optical Character Recognition (OCR) techniques. The authors utilized this classifier to generate predictions for Facebook posts that were created, seen, or engaged. The classifier produced predictions at the user level, ranging from 0 (left-leaning) to 1 (right-leaning).

Although BERT has been used in previous studies as a classifier to detect political leaning, these studies only considered two social media platforms, namely Facebook and Twitter. As such, no previous studies have targeted YouTube videos to automatically detect one of six categories of political leaning based solely on video titles.

Materials and methods

Data overview

The classification of the political leaning of YouTube videos has been examined in two studies, each using a different categorization of videos [21, 22]. To unify the categories used in this context, Hosseinmardi et al. [20] proposed a dataset of 11.5 million YouTube videos that were collected in 2016–2019 and labeled into six political categories, namely: Far Left, Left, Center, Anti-Woke, Right, Far Right. The vast majority of videos in this dataset are primarily concerned with the U.S. political zeitgeist. It should be noted that the videos are classified based on the political leaning of the channels they fall under, rather than the videos themselves. For instance, given a channel that is categorized as Left, all videos therein are also categorized as Left. While this approach has the advantage of being scalable, it could assign inaccurate labels to any videos whose leanings may differ from those of the channel under which they fall.

In our experiment, the dataset of Hosseinmardi et al. [20] is used. The titles of the videos therein were retrieved and cleaned to avoid duplicates and missing values, resulting in a dataset consisting of 10,216,502 video titles. These titles were utilized to train and evaluate three text classifiers; see Methods for more details. Figure 1a depicts the distribution of the six political categories in our dataset, showing that the dataset is imbalanced, with the majority of videos falling under the Center category. Figure 1b shows that the testing dataset exhibits a similar imbalance. Thus, to obtain high prediction accuracy, it is essential that the training stage takes into account this imbalance.

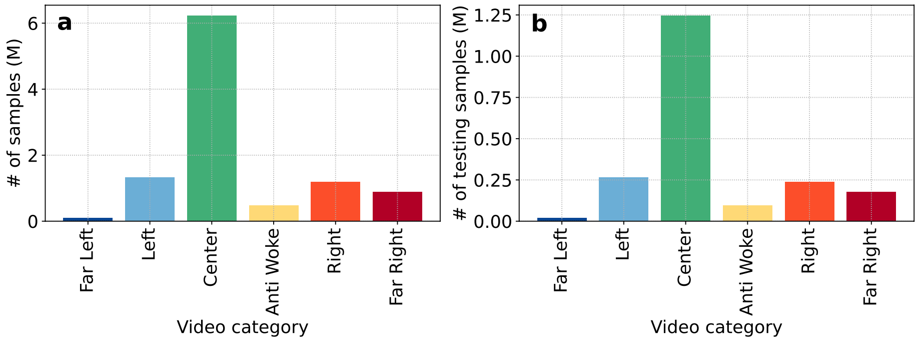


Fig. 1 Distribution of categories in our dataset. The left plot depicts the distribution of category in the entire dataset, while the right plot depicts the distribution in the testing dataset

The 10,216,502 video titles in our dataset were split into three disjoint sets: (i) a training set consisting of 6,538,557 titles used to train the text classifier on video titles; (ii) a validation set consisting of 1,634,642 titles used to validate the classifier, optimize the architecture, and fine-tune the hyperparameters; and (iii) a testing set consisting of 2,043,303 titles used to evaluate the classifier prediction capability.

Methods

This section describes the algorithm, architecture, and hyperparameters used in the experiments. It also describes the three embedding models used, namely Word2Vec [32], GloVe [33], and BERT [34]. Each of these models has its own algorithm and architecture. Several experiments were conducted to determine the optimal architecture of each model, i.e., the one that yields the highest accuracy based on the validation data. To build the video title classification models, other layers were added, such as convolutional 1-D, LSTM, and dense layers. Furthermore, a weighted loss function was utilized to assign greater weights to the classes that have minority samples—a technique commonly used when dealing with imbalanced datasets [36].

Word2Vec

Word2vec is a Natural Language Processing (NLP) technique that utilizes a shallow, two-layer neural network trained to reconstruct the linguistic contexts of words. This technique usually learns word representations by representing each word in a large corpus of text as a vector called an embedding vector. Using this technique, the semantic and syntactic qualities of words can be captured by calculating the cosine similarity between the words represented by embedding vectors [32].

In our experiment, a Word2Vec embedding model was used. It was trained on a Google News dataset with a corpus of six billion tokens and a vocabulary size of one million, consisting of the most frequent words [32]. The model was fine-tuned on our video title dataset, with 700,000 vocabularies and a maximum sentence length of 100. Each word is represented by 300 dimensions. A sequence of layers was used, including a convolutional 1-dimensional layer, a batch normalization layer, and a max pooling layer, followed by two dense layers. The last dense layer produced six probabilities corresponding to the six political leaning categories, i.e., Far Left, Left, Center, Anti-Woke, Right, and Far Right. This architecture is the one that yielded the highest validation accuracy when

Table 1 The architecture of the Word2Vec-CNN fine-tuned model

Layers	Hyperparameters
Embedding model	Embedding dimension = 300 Vocabulary size = 700,000 Max sentence length = 100
Convolutional 1D	512, 3, activation = 'relu'
Batch normalization	N/A
Max pooling 1D	3
Global max pooling 1D	N/A
Dense	512, activation = 'relu'
Dropout	0.7
Dense	6, activation = 'Softmax'

Table 2 The hyperparameters for the Word2Vec-CNN fine-tuned model

Hyperparameters	Values
Optimizer	Adam
Loss function	Sparse categorical cross entropy
Learning rate	1e−04
Batch size	256
Epochs	25

optimizing the hyperparameters. See Table 1 for an overview of the Word2Vec architecture, and Table 2 for a summary of the other hyperparameters used.

GloVe

GloVe is an unsupervised learning method that is also used to obtain vector representations of words, but with a different training process compared to Word2Vec. The training targets a word-word co-occurrence matrix, and is carried out by finding aggregated global word-word co-occurrence statistics in a corpus to capture the frequency with which words co-occur with one another [33].

In our experiment, a GloVe embedding model was trained on the Wikipedia 2014 + Gigaword 5 datasets (6 billion tokens, 400,000 vocab, uncased, 300 dimension vectors), and fine-tuned using our video titles dataset. The resulting GloVe model consists of 50,000 vocabularies with a maximum sentence length of 100. Each word is represented by 300 dimensions. A sequence of two Bidirectional LSTM layers was added before the dense layers. The last dense layer produced six probabilities corresponding to the six political leaning categories. This architecture is the one that gave the highest validation accuracy while tuning the hyperparameters. Table 3 summarizes the GloVe model's architecture, while Table 4 specifies the other hyperparameters used.

BERT

The state-of-the-art text classifier Bidirectional Encoder Representations from Transformers (BERT), which is based on the transformer architecture, was used as the base upon which PoLYTC is built. BERT provides a dense vector representation of natural

Table 3 The architecture of the GloVe-LSTM fine-tuned model

Layers	Hyperparameters
Embedding model	Embedding dimension = 300 Vocabulary size = 50,000 Max sentence length = 100
Bidirectional LSTM	64
Bidirectional LSTM	64
Dense	6, activation = 'Softmax'

Table 4 The hyperparameters for the GloVe-LSTM fine-tuned model

Hyperparameters	Values
Optimizer	Adam
Loss function	Sparse categorical cross entropy
Learning rate	0.001
Batch size	256
Epochs	8

language using a deep, pre-trained neural network [34]. To train BERT, the developers used both Masked Language Model (MLM) pre-training as well as Next Sentence Prediction (NSP) techniques. The design of BERT is based on pre-training deep bidirectional representations from unlabeled text by jointly conditioning on both right and left context in all layers. The advantage of using BERT in PoLYTC is the fact that the preprocessing stage is not required, given that the WordPiece tokenization technique is already involved. This technique was designed to tokenize sentences based on out-of-vocabulary words.

The BERT pre-trained preprocessor and encoder were trained on the Wikipedia and BooksCorpus datasets for general tasks like MLM and NSP. Despite this training, the model cannot simply be used with its current parameters for the fine-grained political classification tasks that PoLYTC seeks. Hence, for this study, the BERT pre-trained's layers should first be fine-tuned with a large-scale YouTube video title dataset in order to achieve the desired classification task. There are several approaches to fine-tuning the BERT model: (1) fine-tuning the classification layers only; (2) fine-tuning the classification layers and a few previous layers; and (3) transfer-learning by fine-tuning all the model's layers. The latter approach has the potential to produce superior performance in terms of accuracy, but it requires a large dataset for fine-tuning. In this study, given the availability of big data, including millions of video titles, it was possible to opt for the latter approach.

BERT utilizes only the encoder part of the transformer and learns a multi-head attention mechanism consisting of heads that operate in parallel to one another. This mechanism learns the contextual relations between sub-words in a text. Attention has the ability to assign weights to each sub-word in a sentence based on its importance. Figure 2 illustrates BERT's classifier architecture, encoder architecture, and multi-head attention mechanism. As shown in the figure, the multi-head attention

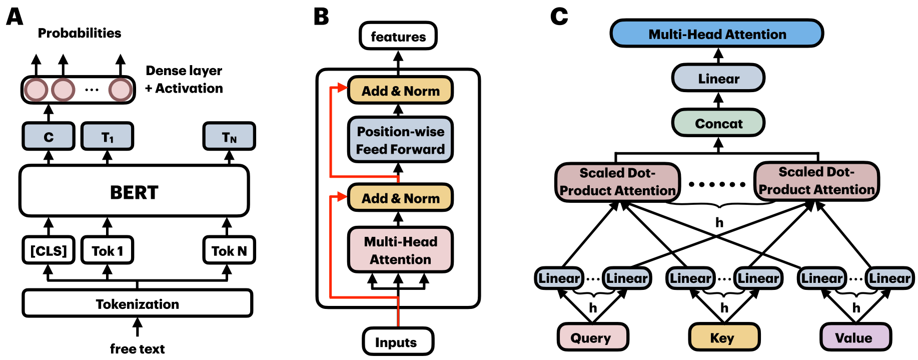


Fig. 2 BERT's architecture. **A** BERT classifier architecture, **B** BERT encoder architecture, and **C** multi-head attention mechanism

Table 5 The architecture of the BERT fine-tuned model

Layers	Hyperparameters
BERT preprocess	
BERT encoder	
Dropout	0.3
Dense	512, activation = 'relu'
Dropout	0.3
Dense	1024, activation = 'relu'
Dropout	0.3
Dense	6, activation = 'Softmax'

mechanism follows a special scaled dot-product attention calculation approach. This scaled dot-product attention can be expressed as:

$$\text{Attention}(Q, K_i, V_i) = \text{Softmax}\left(\frac{Q \times K_i^T}{\sqrt{d_k}}\right) \times Vi \tag{1}$$

where Q, K, and V are 'Query', 'Key', and 'Value' matrices, respectively. and $\frac{1}{\sqrt{d_k}}$ is a scale factor used to adjust the calculation result.

The BERT model was fine-tuned using our video titles dataset in end-to-end fashion (training all layers from the video title at the input to the political leaning category at the output), resulting in a model in which each word is represented by 768 dimensions. A sequence of dense and dropout layers was added. The last dense layer produced six probabilities corresponding to the six political leaning categories. This architecture, which yielded the highest validation accuracy, is summarized in Table 5, and the other hyperparameters used are specified in Table 6.

The implementation of BERT was done using the TF Hub model from the TensorFlow Models repository on GitHub [37]. It uses L = 12 hidden layers (i.e., Transformer encoder blocks), a hidden size of H = 768, and A = 12 attention heads. All parameters in the BERT model were fine-tuned using video titles.

Table 6 The hyperparameters for the BERT fine-tuned model

Hyperparameters	Values
Embedding dimension	768
Optimizer	Adam
Learning rate	1e−04
Loss function	Sparse categorical cross entropy
Batch size	128
Epochs	10

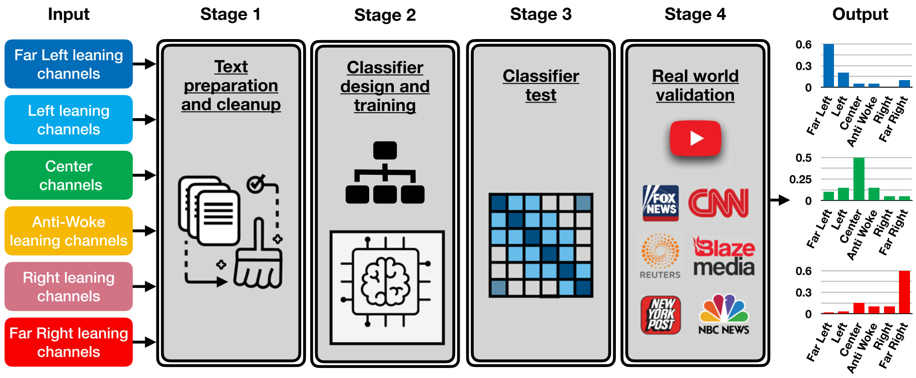


Fig. 3 Experimental pipeline

The training epoch was set to 100 epochs, but the early stopping technique was activated. More specifically, during the training phase, the model validation loss was monitored, and the training was terminated automatically as soon as the validation loss remained unchanged for five consecutive epochs, indicating model convergence. Following this approach, the training stopped after 10 epochs. Given the ability of BERT's embedding model to capture the text representation from both directions, it is sufficient to add dense classification layers with a predetermined number of categories. The dropout layer was added to avoid overfitting and improve accuracy. Given that the official BERT implementation uses the Adam optimizer [38], it was also used during the fine-tuning phase of PoLYTC.

Experimental pipeline

This section explains the different stages undertaken during our experiment. In Stage 1, the labeled video titles are prepared and cleaned. In Stage 2, the classification model is designed, trained, and validated utilizing the video title dataset. In Stage 3, the model is tested using a separate set of video titles to evaluate its performance. Finally, in Stage 4, video titles collected from 15 YouTube channels are used for model validation. Figure 3 illustrates the pipeline used in the experiments.

Experimental results

This section discusses the results after conducting several experiments to train and validate the aforementioned text classifiers—Word2Vec, GloVe, and BERT—using video titles as textual data. Here, the categorization proposed by Hosseinmardi et al. [20] was employed. It consists of six classes: Far Left, Left, Center, Anti-Woke, Right, and Far Right. The three classifiers were trained with these six classes using the video title dataset [20–22]. The models have been implemented after carefully configuring the architectures and hyperparameters that yielded the best performance in terms of accuracy and F1-score. Given the ability of BERT's embedding model to capture the text representation from both directions, adding dense layers for classification purposes is sufficient. On the other hand, adding dense layers to Word2Vec or GloVe did not yield better performance because of their representation limitations. Hence, the performance of Word2Vec and GloVe was improved either by replacing the dense layers with convolutional and pooling layers (1-D CNN) or by adding bidirectional long short-term memory layers (LSTM). It was found that Word2Vec-CNN outperforms Word2Vec-LSTM, while GloVe-LSTM outperforms GloVe-CNN.

The results are evaluated and compared in terms of accuracy and F1-score, with a greater emphasis on F1-score due to the imbalanced nature of our dataset. To qualify the upcoming analysis on the word representation of different embedding models, the performance of the models used is first discussed; see Table 7 for a summary of the results.

For GloVe, the model was trained under three scenarios: (i) starting from random embedding weights and then fine-tuning on our data; (ii) transfer-learning by utilizing the pre-trained embedding model without fine-tuning; and (iii) transfer-learning by utilizing the pre-trained embedding model and fine-tuning on our data. In these three scenarios, the weights of the convolutional and dense layers were tuned to customize the model to fit our task, producing six political leaning categories at the output layer. As can be seen in Table 7, training from random weights (scenario i) and using a pre-trained embedding model without fine-tuning (scenario ii) are less efficient than fine-tuning the pre-trained GloVe model (scenario iii); the latter yields the highest accuracy (70%) and F1-score (72%).

For Word2Vec, the models were trained under two scenarios: (i) utilizing the pre-trained embedding model without fine-tuning; and (ii) utilizing the pre-trained

Table 7 A comparison between the proposed video title classifier (BERT) and the other baseline classifiers (Word2Vec and GloVe) in terms of weighted average accuracy, precision, recall, and F1-score

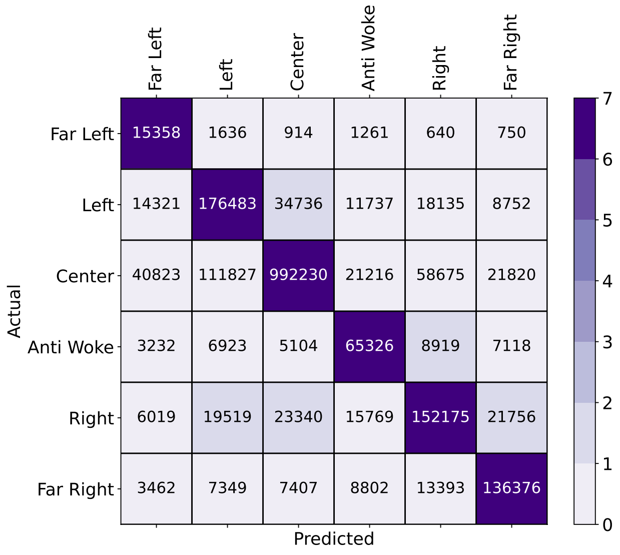
Methods	Average accuracy	Average precision	Average recall	Average F1-score
GloVe trained from random embedding weights (baseline) [33]	0.67	0.75	0.67	0.70
Pre-trained GloVe without fine-tuning (baseline) [33]	0.66	0.74	0.66	0.68
Pre-trained GloVe with fine-tuning (baseline) [33]	0.70	0.77	0.70	0.72
Pre-trained Web2Vec without fine-tuning (baseline) [32]	0.63	0.73	0.63	0.66
Pre-trained Web2Vec with fine-tuning (baseline) [32]	0.71	0.78	0.71	0.73
BERT-based classifier (our proposed model)	**0.75**	**0.80**	**0.75**	**0.77**

Bold values indicate the best performance embedding model and fine-tuning our data. In both scenarios, the weights of the bidirectional LSTM and dense layers were tuned to customize the model to fit our task and produce six political leaning categories at the output layer. As shown in Table 7, utilizing the pre-trained Word2Vec without fine-tuning is less efficient compared to fine-tuning the pre-trained Word2Vec, which has the highest accuracy (71%) and F1-score (73%).

Given that the fine-tuning of a pre-trained model yielded the highest accuracy and F1-score for both GloVe and Word2Vec, a similar approach was followed for BERT. This model includes a pre-processor and an encoder, both of which were fine-tuned on our dataset. Additionally, the weights of the classification dense layers were tuned to customize the model to fit our task and produce six political leaning categories at the output layer. As can be seen in Table 7, the fine-tuned BERT model yielded the highest accuracy (75%) and F1-score (77%), outperforming the other classifiers used in the experiments. This can be attributed to BERT's attention mechanism, which plays a significant role in learning powerful word and text representations.

It is worth noting that, with every additional 1% of accuracy, the classifier is able to correctly predict an additional 20,000 videos. As such, the fact that the fine-tuned BERT classifier achieves a 4% increase in accuracy compared to the second-best alternative (i.e., the fine-tuned, pre-trained Web2Vec) translates to a substantial improvement in performance, as it implies that the former classifier can correctly predict an additional 80,000 videos compared to the latter. Motivated by these results, the focus is on our fine-tuned BERT classifier, PoLYTC, for the remainder of this study.

Figure 4 depicts the confusion matrix of PoLYTC. Given the imbalanced nature of the dataset, the visualization of each row is improved by splitting the range of values therein into equal bins, and assigning a different color to each bin (greater values correspond to darker colors). Looking at the confusion matrix, it becomes clear that the data is imbalanced, as the majority of samples belong to the Center category. This implies that the false predictions come largely from incorrectly classifying the videos as Center.



Actual \ Predicted	Far Left	Left	Center	Anti Woke	Right	Far Right
Far Left	15358	1636	914	1261	640	750
Left	14321	176483	34736	11737	18135	8752
Center	40823	111827	992230	21216	58675	21820
Anti Woke	3232	6923	5104	65326	8919	7118
Right	6019	19519	23340	15769	152175	21756
Far Right	3462	7349	7407	8802	13393	136376

Fig. 4 Confusion Matrix of predictions made by PoLYTC

The confusion matrix also shows that the incorrect predictions are mostly concentrated around the correct class. For example, looking at Far Right videos (bottom row), it can be deduced that most of the incorrect predictions are actually classified as Right. While this is an incorrect classification, it is closer to the ground truth than incorrectly classifying the videos as, say, Left or Far Left. Overall, the classifier rarely classifies right-leaning videos as left-leaning, or vice versa.

The classification report is provided in Table 8, specifying the accuracy, precision, recall, and F1-score of PoLYTC for each of the six political leaning categories. As can be seen, Center has the best accuracy (80%), recall (80%), precision (93%), and F1-score (86%); this is probably due to the fact that Center has the largest number of samples compared to other categories. The second-best prediction is for Far Right; while the accuracy, recall, precision, and F1-score are all lower than the corresponding values for Center, they are all higher than the corresponding values for any of the remaining categories. The worst F1-score is for the Far Left category, probably due to the fact that it has fewer samples compared to any other category.

Having evaluated the classifiers using the testing data with two million video titles, the evaluation focuses on a real-world application. In particular, given the YouTube channels of news agencies, the goal is to predict the distribution of the political leaning of the videos in each of these news channels. The ground-truth political leaning of each channel was obtained using the "Allsides Media Bias Chart" [39]. Fifteen news agencies were selected, consisting of five Right, five Center, and five Left. To collect videos from the YouTube channel of each news agency, the *YouTube Search Python* package was used. This package caps the number of videos per channel at around 20,000. For channels containing fewer than 20,000 videos, all the videos therein were collected. Table 9 specifies the ground-truth political leaning of each news agency, along with the number of videos collected from the YouTube channel of each agency.

Figure 5 shows the distributions of the political leaning of videos in each of the 15 YouTube channels. As can be seen, the distributions predicted by PoLYTC are consistent with the ground-truth political leaning for all five Left channels, as well as all five Right channels; see how the most frequent prediction in the blue-bared subplots is Left, and the most frequent prediction in the red-bared subplots is Right. Notice that the channels on each side rarely cover content from the opposite side. However, Left channels are more likely to cover Center content than Right channels, suggesting that the former

Table 8 classification report specifying the accuracy, precision, recall, and F1-score of PoLYTC for each category

Category	Accuracy	Precision	Recall	F1-score
Far left	0.74	0.18	0.75	0.30
Left	0.67	0.55	0.67	0.60
Center	0.80	0.93	0.80	0.86
Anti-woke	0.68	0.53	0.68	0.59
Right	0.64	0.60	0.64	0.62
Far right	0.77	0.69	0.77	0.73
Weighted average	0.75	0.80	0.75	0.77

The bottom row shows the weighted average, taken over all categories

Table 9 The ground-truth label of each news agency, and the number of videos collected from the YouTube channel of each agency

Ground truth category	YouTube channel	Number of videos
Center	Forbes	6390
	The Hill	19,988
	Reuters	19,796
	The Wall Street Journal	19,674
	BBC news	19,547
Left	MSNBC	19,947
	CNN	19,268
	New York Times	10,116
	NBC news	19,215
	The Guardian	7126
Right	Fox news	19,942
	New York post	12,839
	CBN news	19,754
	Blaze media	11,394
	News Max	19,778

ones are less extreme. As for Center channels, the distribution is clearly consistent with the ground truth in three cases (Reuters, Forbes, and The Wall Street Journal), as the most frequent prediction for these channels is Center. As for The Hill, it can be argued that the distribution is also consistent with the ground truth. After all, if the majority of the videos in that channel are split somewhat equally between Right and Left, then the most plausible conclusion would be that the channel is neither Right-focused nor Left-focused, thereby arguably serving as a Center channel. The only channel for which the distribution is inconsistent with the ground truth is BBC. While the channel is classified as Center according to the AllSides media bias chart, almost all its videos are classified as Left according to PoLYTC.

Conclusion and future work

This study contributes to the literature in two ways. First, the transfer-learning approach was utilized by fine-tuning three pre-trained text classifiers, namely Word2Vec, GloVe, and BERT, and fine-tuning them on a dataset consisting of 11.5 million video titles labeled according to their political leaning. Two million videos were reserved for testing purposes, revealing that the proposed classifier, PoLYTC, has an accuracy of 75% and an F1-score of 77%, outperforming other baseline classifiers such as Word2Vector-CNN and GloVe-LSTM. Second, to validate the findings, thousands of videos from 15 YouTube channels were collected from prominent news agencies with widely-known political leanings, such as Fox News and New York Times, and plotted against their leaning distributions, as predicted by PoLYTC. In the vast majority of cases, PoLYTC's predictions are consistent with the political leaning reported by the AllSides Media Bias Chart [39]. Overall, PoLYTC is able to detect the political leaning of YouTube videos, and classify them into six categories—Far Left, Left, Center, Anti-Woke, Right, and Far Right—based solely on the videos' titles. PoLYTC can be a practical tool to analyze the political leaning of any YouTube channel.

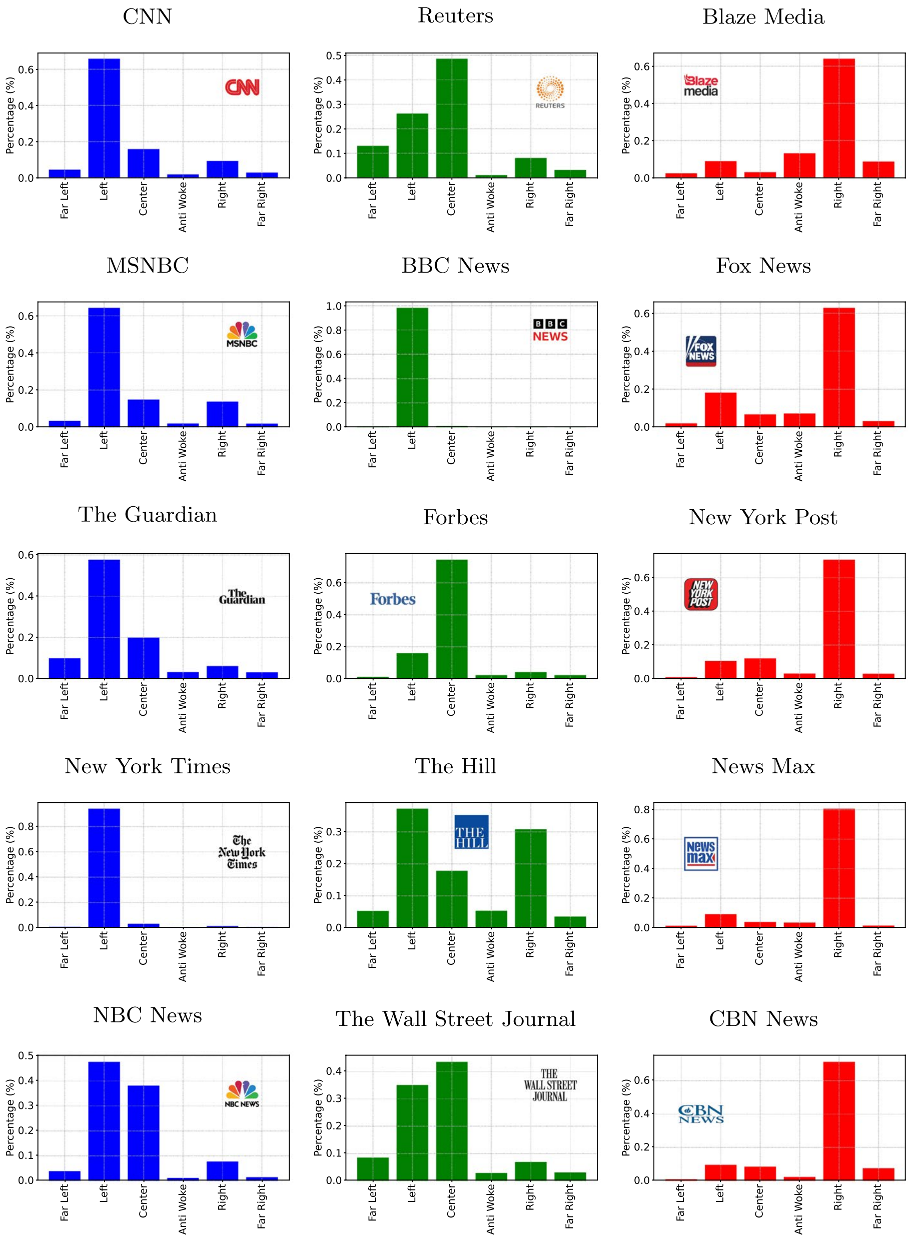


Fig. 5 Distribution of political leaning predictions of videos in 15 YouTube channels. The left, center, and right columns correspond to channels whose ground truth political leaning is Left, Center, and Right, respectively

In future work, one could obtain superior performance by training a classifier on a dataset in which every video is labeled based on its content and not just the channel it falls under. Additionally, to improve the prediction of political leaning, utilizing the transcripts of videos may be valuable, as it allows for videos with similar titles to vary in terms of their political leaning. The transcript may be overly long, and thus summarizing the transcript (e.g., using a Large Language Model) may be required to feed the classifier with relatively shorter transcripts. Furthermore, the study could be extended by targeting other video streaming platforms, such as TikTok and Instagram, which are more popular among young people.

Acknowledgements
Not applicable.

Author contributions
N.A., T.R. and Y.Z. conceived the study, designed the research, produced the visualizations, and wrote the manuscript; N.A. performed the literature review, collected and analyzed the data, and ran the experiments.

Funding
No funding to declare.

Availability of data and materials
The datasets generated and/or analysed during the current study are available in the following repository: https://github.com/comnetsAD/YouTube_political_leaning.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Received: 16 February 2024 Accepted: 31 May 2024
Published online: 05 June 2024

References

1. Americans almost equally prefer to get local news online or on TV Set; 2019. https://www.pewresearch.org/journalism/2019/03/26/nearly-as-many-americans-prefer-to-get-their-local-news-online-as-prefer-the-tv-set/. Accessed 8 Nov 2023.
2. Ceci, L.: YouTube: Hours of video uploaded every minute 2022; 2023. https://www.statista.com/statistics/259477/hours-of-video-uploaded-to-youtube-every-minute/. Accessed 8 Nov 2023.
3. YouTube: YouTube for Press; 2023. https://blog.youtube/press/
4. Konitzer T, Allen J, Eckman S, Howland B, Mobius M, Rothschild D, Watts D.J. Comparing estimates of news consumption from survey and passively collected behavioral data. Public Opin Quart. 2021;85(S1):347–70.
5. Schomer, A. US YouTube advertising 2020. eMarketer; 2020. https://www.emarketer.com/content/us-youtube-advertising-2020/
6. D'Alonzo S, Tegmark M. Machine-learning media bias. PLoS ONE. 2022;17(8):0271947.
7. Kulkarni V, Ye J, Skiena S, Wang WY. Multi-view models for political ideology detection of news articles; 2018. arXiv preprint arXiv:1809.03485.
8. Li C, Goldwasser D. Encoding social information with graph convolutional networks for political perspective detection in news media. In: Proceedings of the 57th annual meeting of the association for computational linguistics; 2019. p. 2594–604.
9. Aksenov D, Bourgonje P, Zaczynska K, Ostendorff M, Schneider JM, Rehm G. Fine-grained classification of political bias in German news: a data set and initial experiments. In: Proceedings of the 5th workshop on online abuse and harms (WOAH 2021); 2021. p. 121–31.
10. Gangula RRR, Duggenpudi SR, Mamidi, R. Detecting political bias in news articles using headline attention. In: Proceedings of the 2019 ACL workshop BlackboxNLP: analyzing and interpreting neural networks for NLP; 2019. p. 77–84.
11. Karpathy A, Toderici G, Shetty S, Leung T, Sukthankar R, Fei-Fei L. Large-scale video classification with convolutional neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2014. p. 1725–32.
12. Abu-El-Haija S, Kothari N, Lee J, Natsev P, Toderici G, Varadarajan B, Vijayanarasimhan SYouTube-8m: A large-scale video classification benchmark; 2016. arXiv preprint arXiv:1609.08675
13. Kalra GS, Kathuria, RS. Kumar A. YouTube video classification based on title and description text. In: 2019 international conference on computing, communication, and intelligent systems (ICCCIS). IEEE; 2019. p. 74–9.
14. Savigny J, Purwarianti A. Emotion classification on YouTube comments using word embedding. In: 2017 international conference on advanced informatics, concepts, theory, and applications (ICAICTA). IEEE; 2017. p. 1– 5.
15. Dinkov Y, Ali A, Koychev I, Nakov P. Predicting the leading political ideology of youtube channels using acoustic, textual, and metadata information; 2019. arXiv preprint arXiv:1910.08948.
16. Mock F, Kretschmer F, Kriese A, Böcker S, Marz M. Taxonomic classification of DNA sequences beyond sequence similarity using deep neural networks. Proc Natl Acad Sci. 2022;119(35):2122636119.
17. Lee J, Yoon W, Kim S, Kim D, Kim S, So CH, Kang J. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. Bioinformatics. 2020;36(4):1234–40.
18. Beltagy I, Cohan A, Lo KS. Pretrained contextualized embeddings for scientific text. In: Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP), Hong Kong, China; 2019. p. 3–7.

19. Peng Y, Yan S, Lu Z. Transfer learning in biomedical natural language processing: an evaluation of BERT and ELMo on ten benchmarking datasets; 2019. arXiv preprint arXiv:1906.05474.
20. Hosseinmardi H, Ghasemian A, Clauset A, Mobius M, Rothschild DM, Watts DJ. Examining the consumption of radical content on YouTube. Proc Natl Acad Sci. 2021;118(32):2101967118.
21. Ledwich M, Zaitsev A. Algorithmic extremism: examining YouTube's rabbit hole of radicalization; 1912. arXiv.
22. Ribeiro MH, Ottoni R, West R, Almeida VAF, Jr, WM. Auditing Radicalization Pathways on YouTube. CoRR; 2019. arXiv: 1908.08313
23. Gu F, Jiang D. Prediction of political leanings of chinese speaking twitter users. In: 2021 international conference on signal processing and machine learning (CONF-SPML). IEEE; 2021. p. 286–9.
24. Tasnim Z, Ahmed S, Rahman A, Sorna JF, Rahman M. Political ideology prediction from Bengali text using word embedding models. In: 2021 international conference on emerging smart computing and informatics (ESCI); 2021. p. 724–7. https://doi.org/10.1109/ESCI50559.2021.9396875
25. Xiao Z, Zhu J, Wang Y, Zhou P, Lam WH, Porter MA, Sun Y. Detecting political biases of named entities and hashtags on twitter. EPJ Data Sci. 2023;12(1):20.
26. Di Gennaro G, Buonanno A, Palmieri FA. Considerations about learning word2vec. J Supercomput. 2021;77:1–16.
27. Essa E, Omar K, Alqahtani A. Fake news detection based on a hybrid bert and lightgbm models. Complex Intell Syst. 2023;9:1–12.
28. Shen Y, Liu J. Comparison of text sentiment analysis based on bert and word2vec. In: 2021 IEEE 3rd international conference on frontiers technology of information and computer (ICFTIC). IEEE; 2021. p. 144–7.
29. Wang C, Nulty P, Lillis D. A comparative study on word embeddings in deep learning for text classification. In: Proceedings of the 4th international conference on natural language processing and information retrieval; 2020. p. 37–46.
30. Jiang J, Ren X, Ferrara E. 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. 2023;17:459–69.
31. Nyhan B, Settle J, Thorson E, Wojcieszak M, Barberá P, Chen AY, Allcott H, Brown T, Crespo-Tenorio A, Dimmery D, et al. Like-minded sources on facebook are prevalent but not polarizing. Nature. 2023;620(7972):137–44.
32. Mikolov T, Chen K, Corrado G, Dean J. Efficient estimation of word representations in vector space; 2013. arXiv preprint arXiv: 1301.3781.
33. Pennington J, Socher R, Manning CD. GloVe: Global Vectors for Word Representation. In: Empirical methods in natural language processing (EMNLP); 2014. p. 1532–43 . http://www.aclweb.org/anthology/D14-1162
34. Devlin J, Chang M-W, Lee K, Toutanova K. Bert: Pre-training of deep bidirectional transformers for language understanding; 2018. arXiv preprint arXiv:1810.04805.
35. Ibrahim H, AlDahoul N, Lee S, Rahwan T, Zaki Y. YouTube's recommendation algorithm is left-leaning in the United States. PNAS Nexus. 2023;2(8):264.
36. Fernando KRM, Tsokos CP. Dynamically weighted balanced loss: class imbalanced learning and confidence calibration of deep neural networks. IEEE Trans Neural Netw Learn Syst. 2021;33(7):2940–51.
37. tensorflow: BERT (Bidirectional Encoder Representations from Transformers); 2023. https://github.com/tensorflow/models/tree/master/official/legacy/bert.
38. google-research: BERT implementation; 2018. https://github.com/google-research/bert/blob/master/optimization.py#L74
39. AllSides media bias chart; 2023. https://www.allsides.com/media-bias/media-bias-chart#biasmatters

Publisher's Note