Identification of Political Ideology

Project Report

Background

Newspapers and online publications play an important role in shaping public opinion, but they are not immune to biases. Very often, they reflect the political leaning of their publisher that affects the content they produce. This phenomenon is particularly present in American politics which has become more polarizing. As a result, politicians draw on sources which lean to their preferred political ideology, dismissing differing perspectives. Furthermore, changes to the way the public consumes news media have exasperated these biases. More than ever people are consuming news from non-traditional sources, such as the internet and social media platforms. By turning to independent news websites, blogs, or social media influencers, news readers face the challenge of identifying the underlying political agendas of the content they read.

Objective

In light of the ongoing election cycle and the ramping up of politically slanted news-content, our group seeks to provide news readers with tools that promote balanced and less time-consuming news consumption. Our primary objective is to develop a natural language processing pipeline to detect political biased ideology in news articles. In addition, we aim to create a prototype of a tool that would allow readers to input the news article/the link and get its summarized version.

Method

To address these objectives, we came up with a comparison of different NLP classification models, including classical Naïve Bayes and Logistic Regression, MLP, CNN, LSTM, and RoBERTa transformer finetuning. We then built our Streamlit bias identification tool on the

model with the leading performance. Additionally, we used Pegasus transformer-based model for the summarization process given its popularity in use for news aggregation and text summarization. We incorporated it into the Streamlit app., providing real-time summarization of the article of choice.

Data

We used the Article-Bias-Prediction dataset which consists of 37,554 articles stored as a JSON object. The dataset is prepared by Baly et al. (2020), annotated on the article level, categorizing 3 glasses of bias – left, center right. It also contains information about the topic, the source URL, the URL of the actual article, the publication date, authors, title, content, and political bias.

Model Architectures

An MLP is a basic feed forward neural network that is useful for basic classification problems. The model built for this project used a basic three-layer architecture. The model consisted of an input layer which takes the input tensors with an embedding dimension of 300, based on the word2vec embedding used (Kamath et al). The model also contained two fully connected hidden layers and finally the output layer. This model uses the Rectified Linear Unit (ReLU) activation function, this function enables more complexity. To train the model, cross-entropy loss function and AdamW optimizer were used. Overall this model achieved accuracy of 58.7% and an F1 score of 0.57 (Table 1).

A CNN model is also a feedforward based neural network model. The CNN model is more complex than the MLP with added convolutions into the neural network module. This model starts with a similar input layer. Then the model has multiple convolutional layers, which mathematically learn and extract features from the input depending on the filter size parameter.

Some other features included in this model architecture include the pooling, concatenation, and dropout which are applied to the outputs of the convolution layers. This model uses the Rectified Linear Unit (ReLU) activation function in the convolutional layers. To train the model, I used a cross-entropy loss function and AdamW optimizer. Overall, the model performance improved from the MLP with an accuracy of 73% and F1 score of .73 (Table 1).

LSTM (Long Short-Term Memory) is a type of recurrent neural network (RNN) architecture that is designed to overcome the limitations of traditional RNNs. They are designed to be able to remember information for long periods of time, unlike traditional RNNs which have difficulty maintaining long-term dependencies in the data. LSTM models determine what historical data to retain and how to update that information over long sequences based on the parameters they learn from the data provided for training(Chung and Masterson, 2020). The preprocessing of the LSTM model was done through a mix of custom-made and PyTorch functions(such as nn.embedding and pack-padded-sequence with a max length of 512. The input size, i.e. dimension of the LSTM layer is set to 100 and the number of layers for the model is set to 3. The model architecture consists of an embedding layer, an LSTM layer, a dropout layer, and two fully connected layers with a ReLU activation function. This LSTM model reached 47.3% test accuracy (Table 1).

RoBERTA is a transformer model and a variant of BERT which has been pretrained on a large corpus of text data using a masked language modeling objective. The preprocessing for this model was done using a RoBERTa Tokenizer with a max length of 512. The base RoBERTa model was fine-tuned using the news bias dataset. To fine tune, the model uses two tensors, inputs and attention masks – which indicate which tokens should be attended to and which

should be ignored. The model is trained with a supervised learning approach to adapt the model to make more accurate predictions for our dataset. The RoBERTa model achieved the best performance with an accuracy of 90%.

Pegasus model for summarization was used to summarize the news articles. Pegasus is based on the Transformer architecture and uses a novel pre-training objective called "Gap Sentence Generation" to learn how to generate abstractive summaries that rephrase or paraphrase the key information in the input text. We chose this model given its strong performance in various summarization benchmarks. We downloaded specifically the MRPC (Microsoft Research Paraphrase Corpus) dataset from the GLUE (General Language Understanding Evaluation) benchmark from the HuggingFace Transformer Library and used it for our summarization objective.

Experiments and Results

We trained, validated and tested our models based on text of the article content. We deliberately did not provide the models with any additional information such as the source or the authors of the articles because we did not want the models to train by familiarizing with some information which might not be provided during testing.

Table 1 summarizes the performance metrics of Naïve Bayes, Logistic Regression, MLP, CNN, LSTM and RoBERTA.

Table 1

Summary of F-1 scores and Testing Accuracy of ML models used in this project

Model	Naïve Bayes	Logistic regression	MLP	CNN	LSTM	RoBERTa
F-1 score	0.50	0.71	0.57	0.73	0.45	0.902
Testing Accuracy	0.54%	0.71%	58.47%	73%	47.3%	0.902%

The comparison of models showed that transformer fine-tuning with RoBERTA model produced the best F-1 core and testing accuracy at 0.902 and 90.2% respectively. Therefore, we incorporated it into our Streamlit demo and we used it as our baseline model for article bias prediction showcase.

Limitations

For the LSTM model, we experimented with custom-made packing and packing sequence as well as used GloVe(Pennington et al. 2014) embedding with 50 dimensions, yet we noticed that using embeddings at a higher dimension, such as the GloVe pre-trained model with 300 dimensions, could potentially produce a better performance score. Since our initial LSTM performance accuracy(36%) and F-1(0.26) scores were not satisfactory, we decided to use neural network specific embedding and padding, which increased our score. In addition, for the LSTM model, we used classical NLP methods of pre-processing(tokenizing, removing stopwords) our data, which may be the reason why our model underperforms transformers' similar functions.

Implications

The results of this project showed a better improvement over using the pre-trained transformer, RoBERTa. We hope that a more rigorous finetuning of the model will help us with a more

correct classification of political ideology. This in turn will enable news readers to be more cognizant of the bias present in news outlets. In addition, our summarization tool will help them have access to bite-size information pieces. For future research, we could explore ways to assess whether the summarized articles maintain their original ideology or not.

References

Baly, R., Da San Martino, G., Glass, J., Nakov, P. (2020). We can detect your bias: Predicting political ideology of news articles. *Archive.* https://arxiv.org/pdf/2010.05338

Chang, C., & Masterson, M. (2020). Using Word Order in Political Text Classification with Long Short-term Memory Models. Political Analysis, 28(3), 395–411.doi:10.1017/pan.2019.46

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. *GloVe: Global Vectors for Word Representation.*

Liu, Yinhan, et al. (2019). Roberta: A robustly optimized bert pretraining approach. *ArXiv preprin*t. arXiv:1907.11692

Nidhi, K.C., Bukhari, S.S., and Dengel, A. (2018). Comparative study between traditional machine learning and deep learning approaches for text classification. *Proceedings of the ACM Symposium on Document Engineering 2018*.

Zhang, J., Zhao, Y., Saleh, M., & Liu, P. (2020, November). Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. *In International conference on machine learning* (pp. 11328-11339). PMLR.