Detecting Bias in News Articles using NLP Models

Stanford CS224N Custom Project

Muhammad Umar Nadeem
Department of Computer Science
Stanford University
munadeem@stanford.edu

Sarah Raza
Department of Computer Science
Stanford University
sraza007@stanford.edu

Abstract

This project analyses various natural language processing (NLP) algorithms in order to build a deeper understanding of the machine learning techniques required to detect biased political leanings in news sources. Since news is the first and most direct source from which people learn about current unfolding events, the introduction of unjust subjectivity can be very harmful. Thus, the ability to detect bias in news sources is key to maintaining truth when disseminating information. The first classification model we implement is a Tensorflow deep neural network (DNN) using bag of words (BOW) to represent the input sentences. Then, we build on this deep learning algorithm by adding term frequency–inverse document frequency (TD-IDF) as a weighting factor to the DNN input data. In an attempt to further improve results, we shift towards an unsupervised K-Means clustering algorithm to analyze patterns discovered amongst articles from the various news source. Finally, we implement SimCSE, a contrastive learning framework for sentence embeddings. We find that contrastive learning is the most accurate NLP model of those tested for detecting the nuances of political bias in news article sentences.

1 Key Information to include

- Mentor: Lucia Zheng
- External Collaborators (if you have any): No
- Sharing project: No

2 Introduction

Nowadays, many of us intentionally first encounter breaking news when it appears in our online feeds. However, this also means we unintentionally encounter many instances of fake and biased information in these news sources that claim to be objective. Bias refers to prejudice in favor of or against one thing, person, or group compared with another, usually in a way considered to be unfair. When brought into the context of media, bias refers to the intentional and unconscious perspective from which news stories are written and the language and framing chosen to tell these stories. As news is the first and most direct source from which people learn about current unfolding events, the introduction of subjectivity can be very harmful. Thus, the ability to detect bias in news sources is key to maintaining truth when disseminating information. This pursuit of truthful reporting and non-targeted framing is the basis of our work. More specifically, we aim to find ways to automate the identification of nuanced language and repetitive underlying themes within various news articles published by a specific news source. We begin this automation through first analyzing various NLP algorithms in order to build a deeper understanding of the machine learning techniques required to detect biased political leanings.

3 Related Work

Our interest was initially peaked by a paper that focuses on using NLP models to detect fake and biased information in news sources that claim to be objective [1]. This paper used a combination of sentiment analysis, clustering based on publisher, and clustering based on author to produce a final score representing the bias in an article. The research related to our work because its larger goal was also to detect bias in news sources.

We planned our project using a similar approach to the paper, using multiple algorithms to detect bias. In the process we became intrigued by a newer model called SimCSE. The paper detailing this model [2] describes both an unsupervised and supervised contrastive learning framework that advances the typical sentence embedding methods. See figure beloow to better understand these frameworks. We were especially interested by supervised SimCSE which incorporates annotated pairs from natural language inference datasets into the contrastive learning framework. The framework uses "entailment" pairs as positives and "contradiction" pairs as hard negatives to better understand similarities and differences between sentences.

This paper was different from the first because it provided a model we could use rather than an example of a model similar to the one we wanted to build. However, the two papers were also alike because both described ways to detect nuances in language using NLP methodologies.

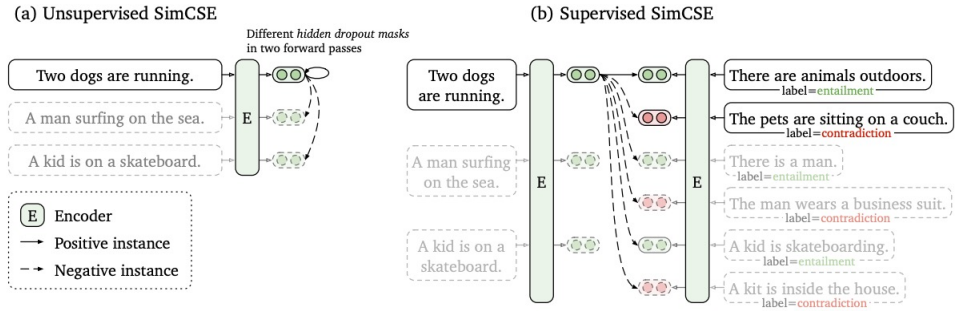


Figure 1: (a) Unsupervised SimCSE predicts the input sentence itself from in-batch negatives, with different hidden dropout masks applied. (b) Supervised SimCSE leverages the NLI datasets and takes the entailment (premise-hypothesis) pairs as positives, and contradiction pairs as well as other in-batch instances as negatives.

4 Approach

Our goal is to effectively detect the news source a sentence is from. This functionality will allow us to build a model that picks up on nuances in texts and that model can later be used to understand subtle signs of bias in text. We apply a phased approach to doing this.

First, we build a baseline model by implementing a Tensorflow deep neural network (DNN) using bag of words (BOW) to represent the input news article sentences. BOW is a way of extracting features from text to build a model. That model uses the occurrence of words within a document to build a vocabulary of known words and their frequencies. Then, the model can then be used to identify common words within articles to attribute sentences to their sources. One drawback of this model is that only considers whether known words occur in the document and, thus, information about the order or structure of words is discarded.

Then, we build on our baseline algorithm by adding term frequency–inverse document frequency (TD-IDF) as a weighting factor to the DNN input data. TF-IDF is a statistical measure that evaluates how relevant a word is to a document in a collection of documents. This is done by multiplying two metrics: how many times a word appears in a document, and the inverse document frequency of the word across a set of documents. This model allows us to not only identify common words within articles, but also identify words that are unique to specific news sources.

Next, we shift towards an unsupervised K-Means clustering algorithm using TD-IDF weighted word embeddings to analyze patterns discovered amongst articles from the various news source. A K-means clustering algorithm groups similar items into a set number, namely k, of clusters. It does so by first selecting initial cluster points, assigning each observation or data point to the closest cluster, and calculating new means for the clusters. This process repeats until a desired clustering is reached. This model allows us to identify underlying similarities between sentences from across all news sources. As each news source may not have its own unique political leaning, K-Means allows us to group news article sentences with shared subjective language and framings together.

Finally, we implement supervised SimCSE, a contrastive learning framework for sentence embeddings. Contrastive learning is an approach to formulate the task of finding similar and dissimilar features. The inner working of contrastive learning can be formulated as a score function, which is a metric that measures the similarity between two features. More specifically, x+ is data point that is similar to x, and is referred to as a positive sample. On the other hand, x- is a data point dissimilar to x, and is referred to as a negative sample. Over this, SimCSE uses a logistic regressing classifier with an Adams optimizer to then identify positive and negative samples correctly. This model allows us to recognize entire sentences that are similar across articles published by a specific news source. This helps us determine consistent nuances and framing choices in political discourse for each news source.

5 Experiments

5.1 Data

The datasets we are using for training and testing our algorithm are Wei's NewB news source sentences about former United States President Donald Trump [3]. For training data, Wei has compiled approximately 250,000 sentences from 11 news sources, five liberal sources (Newsday, New York Times, CNN, LA Times, Washington Post), one neutral source (Politico), and five conservative sources (Wall Street Journal, New York Post, Daily Press, Daily Herald, Chicago Tribune) into a .txt file that labels each sentence with the corresponding source. We preprocessed this training data using pandas dataframes into a .csv file, and then extracted 1000 sentences from each source. We stored these 11,000 training sentences as eleven lists of strings that we inputed as JSON data to an nltk tokenizer.

Similarly, for the testing data, Wei has compiled approximately 11,000 sentences from the same 11 news sources. We preprocessed this testing data using pandas dataframes into a .csv file, and then extracted 100 sentences from each source. We stored these 1100 training sentences as eleven single lists of strings that we inputted as labeled dictionaries to our evaluation function.

5.2 Evaluation method

The final evaluation for each algorithm is computed through an accuracy function for each news source: the number of correctly predicted sentences for the news source divided by the total number of sentence predictions for that specific news source. For our baseline and the DNN with TF-IDF, the model predicts a label for each input sentence and thus the number of correctly predicted sentences is easily obtained.

However, the K-Means algorithms is unsupervised, pattern based learning and thus no single cluster can be attributed to a source. This means we cannot identify the number of "correctly" predicted sentences. Rather, we look at the spread across clusters that the test sentences from each news source are assigned to.

For SimCSE, the output from comparing two sentences is the Spearman correlation coefficient. To obtain a label prediction from the model, we first compare each test sentence to multiple sentences from the same source. Then, we average those Spearman coefficents from one news source to get an average coefficient. Finally, whichever news source had the highest positive average Spearman coefficient is considered the model's prediction.

5.3 Experimental details

In total, we ran four different experiments: our baseline DNN with a BOW model, addings TF-IDF as a weighting factor for the input vectors to our baseline DNN, K-Means Clustering, and SimCSE. All of these experiments required the data preprocessing explained above. The model configurations for each experiment are as follows:

5.3.1 BOW and TF-IDF

We ran our experiment by developing a baseline neural network model using the bag-of-words representation. Our model inputs word vectorized news source sentences into a Tensorflow DNN with a Softmax activation function and an Adam optimizer. Our batch size is 8 and there are 4 layers in our DNN model. The first layer of our model is the visible/input layer where the information is known, the second two layers are hidden, and the last layer is the output layer which directly links to the label the model is trying to predict. We did not have a dropout layer. To train over 100 epochs, it required two hours to complete on our Azure VM with 56 GiB of memory. To then implement TF-IDF, we maintained the same model and model configurations, just replacing the input with TF-IDF weighted, vectorized news source sentences.

5.3.2 K-Means

Similar to the first experiment, our K-Means model trains on TF-IDF weighted, vectorized news source sentences. We initialized the cluster centers using *kmeans++* from *sklearn* with a maximum number of iterations *max_iter* set to the default value of 300 and *n_init* set to 10. *n_init* is the number of time the k-means algorithm will run with different centroid seeds. The final results will be the best output of *n_init* consecutive runs in terms of inertia. We trained the model first with 11 clusters in hopes of identifying 11 unique patters for 11 unique sources. We then used the Elbow Method to find the optimal number of clusters, namely 2, and also ran the model with the updated parameter. Both iterations of training the model on our Azure VM with 56 GiB of memory required 45 minutes to one hour.

5.3.3 SimCSE

Because our training data was clearly labeled and the supervised bag of words model was much more effective than the unsupervised k means model, we chose to implement supervised SimCSE. Similar to the paper, we use an Adam optimizer with warmup steps and a linear learning rate scheduler. The warmup ratio was .05. We use a 12 layer BERT model to train and test data. More specifically, the cl_init function is used to initialize the model and the cl_forward function is used for training and testing. The parameters for our model are as follows:

- batch size is 256
- number of epochs is 3
- learning rate for training is .0001
- learning rate for testing is .00005

In terms of training details, for pooler type, we chose the 'cls' in which there is a simple MLP layer that gets sentence representations over BERT's CLS representation. We chose this because it had the highest success rate for the supervised model in the SimCSE paper [2]. Our similarity function calculates cosine similarity between the embeddings and thus our loss function calculates the cross entropy loss for the cosine similarity scores. Those cosine scores are then used to calculate the spearman coefficient. To train 3 epochs with 11,000 lines of testing data, it required 8 hours to complete on our Azure VM with 56 GiB of memory.

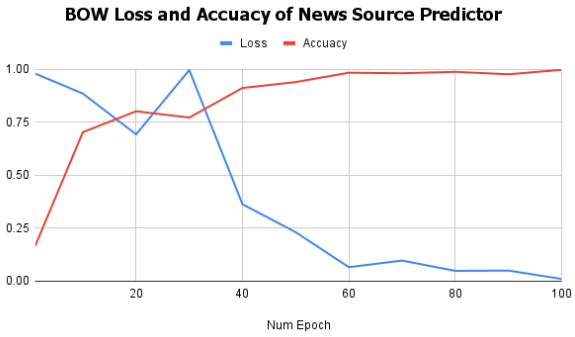
Once we train our model, we test its accuracy by using a separate python script in which 500 unknown sentences were each compared to eleven sets of 1000 labeled sentences (one for each news source). Each comparison resulted in a Spearman coefficient and the the highest positive value was considered the model's prediction. This process for testing accuracy was completely original.

5.4 Results

5.4.1 BOW (Baseline)

Our baseline model correctly classifies news source sentences to their respective new sources with an average accuracy of 15.6% (see table below for full breakdown). This is about what we expected from this model because it trains on word frequency and all articles talk about Donald Trump and thus share a similar vocabulary. The total loss and model accuracy over the 100 training epochs are shows below:

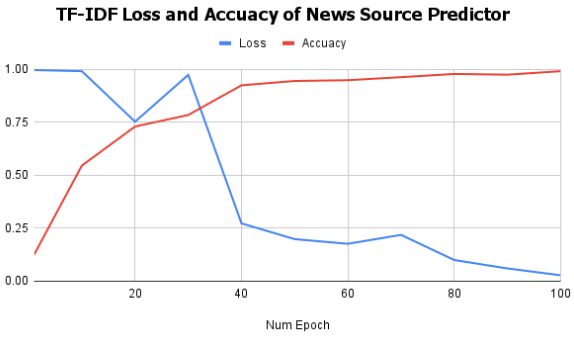
News Source	Baseline (%)
Newsday	19.2
NYT	22.6
CNN	15.6
LA Times	12.2
WashPost	12.8
Politico	13.8
WSJ	14.2
NYPost	22.6
DailyPress	10.6
DailyHerald	21.6
ChicagoTribune	9



5.4.2 TF-IDF

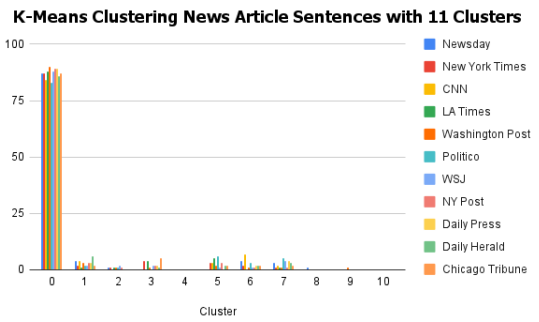
Our DNN with TF-IDF as a weighting factor for the input vectors correctly classifies news source sentences to their respective new sources with an average accuracy of 15.1% (see table below for full breakdown). This is about what we expected from this model because it trains on word frequency across sources and all sources also talk about Donald Trump and thus share a similar overall vocabulary. The total loss and model accuracy over the 100 training epochs are shows below:

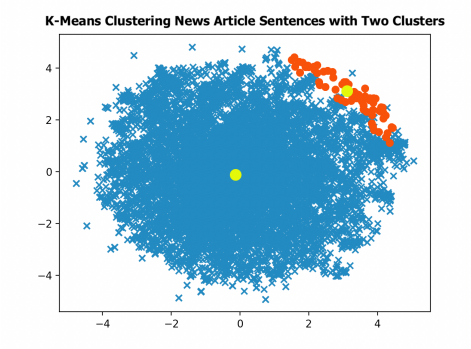
Baseline vs TF-IDF Prediction Accuracy		
News Source	Baseline (%)	TF-IDF (%)
Newsday	19.2	18.6
NYT	22.6	23
CNN	15.6	11.6
LA Times	12.2	9.6
WashPost	12.8	11.2
Politico	13.8	12.6
WSJ	14.2	16.2
NYPost	22.6	19.2
DailyPress	10.6	11.6
DailyHerald	21.6	21.4
ChicagoTribune	9	10.6



5.4.3 K-Means

Our K-Means algorithm was not effective in finding patterns amongst article sentences from the same source or even amongst article sentences across different sources. This is worse than what we initially expected from this model, but we can now attribute a similar limitation to this model as discussed with our baseline and TF-IDF regarding a limited variance of vocabulary across the entire article sentence domain. When the model was run with 11 clusters, 95.8% of the test sentences were assigned to cluster 0. The full breakdown of the test sentence assignments are shown below to the left. When the model was run with 2 clusters, 95.0% of the test sentences were assigned to cluster 0, as visualized below to the right:



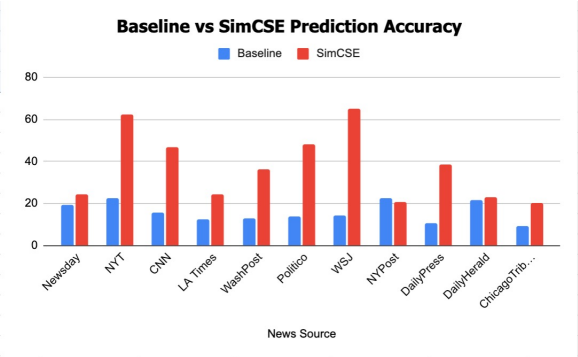


5.4.4 SimCSE

Our SimCSE model classifies news source sentences to their respective new sources with an average accuracy of 24.3% (see table below for full breakdown). This is better than what we expected from this model for certain sources. We attribute the high accuracy of certain sources to a more structured sentence format for all articles published by those specific news sources. The model accuracy compared to our baseline model is shown in the bar graph below:

Baseline vs SimCSE Prediction Accuracy

News Source	Baseline (%)	SimCSE (%)
Newsday	19.2	24.3
NYT	22.6	62.5
CNN	15.6	46.8
LA Times	12.2	24.1
WashPost	12.8	36.3
Politico	13.8	48.3
WSJ	14.2	64.9
NYPost	22.6	20.5
DailyPress	10.6	38.7
DailyHerald	21.6	22.8
ChicagoTribune	9	20.2



6 Analysis

Our analysis of four different NLP algorithms for bias detection in news source highlights a couple key insights regarding the machine learning techniques required to detect biased political leanings in news sources. First, we see that TF-IDF as a feature scaling method does not improve or even really change the classifier's accuracy above our BOW baseline DNN. After further research and examination of the data, this can be attributed to the fact that TF-IDF weighting does not change the column space of the data matrix. The right feature scaling can be helpful for classification because it accentuates the informative words and downweghts the common words. However, in this case uniform column scaling was not helpful in any way due to the fact that there were no "informative words" or "common words" that stood out amongst the different new article sentences. All the sentences, in training and testing data and from all sources, are all versions of someone talking about Donald Trump. Therefore, the vocabulary is very very similar across all sentences, and word vectorization is unable to pick up the nuances of political bias in how those words were ordered, phrased, or emphasized. This same reasoning regarding word vectorization can extend to K-means and its inability to find patters amongst sentences from the same or varying sources. Second, we see that SimCSE outperforms all the other models. This tells us that contrastive based learning is the most successful for detecting the nuances between different news sources. This likely because SimCSE is the only model taking into account factors such as framing rather than just word choice because it considers the whole sentence.

7 Conclusion

It is clear that sentence embeddings and overall greater contextual understanding are necessary to detect the nuances of political bias and leaning in published news articles. Simple word embedding models such as BOW with TF-IDF are not successful in capturing varying framings of similar topics because the overall vocabulary of each news source does not have significant variance. Similarly, an unsupervised model such as K-Means with word embedding input vectors is not able to find patterns across articles from a specific news source or across news sources because of a limited variance in vocabulary. The limitations of our work are the specific news sources from which we obtained our training and testing data, and well as the computing power required to train and test on entire articles rather than sentences. As follows, future avenues of work include expanding this research for a wider range of online news sources, and scaling the models to larger input embeddings.

References

[1] Premanand Ghadekar, Mohit Tilokchandani, Anuj Jevrani, Sanjana Dumpala, Sanchit Dass, and Nikhil Shinde. Prediction and classification of biased and fake news using nlp and machine learning models. In Debabala Swain, Prasant Kumar Pattnaik, and Tushar Athawale, editors, Machine Learning and Information Processing. Springer Singapore, 2021.

[2] Gao, Tianyu, Xingcheng Yao, and Danqi Chen. "Simcse: Simple contrastive learning of sentence embeddings." arXiv preprint arXiv:2104.08821 (2021).

[3] Wei, J (2020) NewB, Github Repository. Software available from github.com/JerryWei03/NewB.