Noise not News: The Shortcomings of a Weak Supervision Model for Ideological Bias

Noah Levy, Samir Datta, K.C. Tobin

May 2018

Abstract

Classification of political sentiment or bias in news articles would go a long way in educating readers of their own biases in their choice of news sources allowing them to make more educated decisions about how to consume news. In order to accomplish that we proposed a semi-supervised method for identifying political sentiment of sentences to improve upon the baseline methods and utilize additional information from additional data sources to allow for the classification of a much larger corpus. We evaluated our techniques by testing its accuracy on a corpus of sentences that have been validated and labeled with their ideological bias. Although we were able to approximately replicate the accuracy of previous work, our attempts at implementing weak supervision did not substantially improve the performance of our model over baseline, likely due to noise or the inherent differences between our datasets. Still, we feel that there is much potential in the future for weak supervision to be effectively used in political lean classification.

Introduction

With the most recent Presidential election, a national conversation arose around identifying unbiased sources of news. While readers may have a preference for the type of information they receive and the opinions they may include it is important that they are able to recognize this choice instead of having it thrust upon them. In the United States, we typically divide the spectrum of political ideologies along party bases with more liberal sentiments being associated with Democratic Party views and more conservative sentiments associated with Republican Party views. Certain news outlets have chosen to identify themselves politically but many do not and many attempt to straddle the middle. However, each article can still run the gamut of reflecting a certain political ideology based on the author or many other factors. When it comes to full articles this problem becomes particularly challenging as the bias is typically localized to a small portion of the document. Due to available data in this area we choose to address this problem on the sentence level where bias is more likely to be detected.

We plan to build a model that classifies the political lean of sentences from American news articles. The model would accept a sentence of an article as an input, and predict its political lean as conservative or liberal, as well as output a probability for each class. Ideally, our model and its techniques could be extended to full article prediction and would enable users to know whether they are creating their own echo chambers or choosing articles that challenge their prior opinions.

Goal: To create a political lean classifier that outperforms baseline models on the Ideological Books Corpus, and effectively classifies sentences as liberal or conservative.

A major challenge many models including ours face is that of availability of labelled data. We attempted to address this problem using a weak supervision technique which would allow us to incorporate a significant amount of additional data into the training task. We trained the model using a combination of labeled data from previous studies on sentence-level political classification and unlabeled data from new articles. The All the News dataset on Kaggle contains tens of thousands of articles from a plethora of sources including Vox, NPR, Breitbart, and Fox News. We also gathered data from various reddit subreddits that identify themselves with a political lean.

Background

Political Lean Prediction

The Ideological Books Corpus (IBC) was developed by Gross et al. in 2013. It includes a myriad of book and magazine articles written between 2008 and 2012, and each document is manually labeled with a general polarity: left, right, or center. Iyyer et al. used a combination of keyword filtering and crowdsourcing to label the polarity of the sentences within the IBC (Iyyer et al., 2014). Their final corpus contains 1,701 conservative sentences, 2,025 liberal sentences, and 600 neutral sentences.

Iyyer et. al excluded the neutral sentences from their analysis, and trained a recursive neural network with word2vec embeddings to predict whether a given sentence had a liberal or conservative lean. Their best classifier predicted the polarity of sentences in the IBC with 69% accuracy.

Iyyer et. al also trained a model to predict the polarity of sentences in the Convote Dataset. This corpus was developed by Thomas et al. in 2006, and it consists of the transcripts of US Congressional floor debates with labels on the speaker and the speaker's political party. They propagated the party labels for the speakers down through the individual sentences in the transcript to obtain ideological labels. Their best recursive neural network model achieved 70.2% accuracy on this dataset.

Weak Supervision

Dehghani et al. recently developed a learning algorithm for tasks that involve a small quantity of true-labeled data and a large quantity of unlabeled data (Dehghani et al., 2017). At a high level, weak supervision involves three key components, a small set of true-labeled data, a large quantity of unlabeled data, and an algorithm for applying noisy or "weak" labels to the unlabeled data.

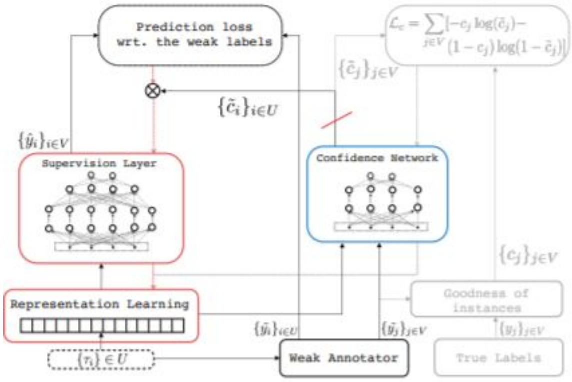
Prediction loss wrt. the weak labels
$\mathcal{L}_c = \sum_{j \in V} [-c_j \log(\tilde{c}_j) - (1-c_j)\log(1-\tilde{c}_j)]$
$\{\tilde{c}_j\}_{j\in V}$
$\{\tilde{c}_i\}_{i\in U}$
$\{\hat{y}_i\}_{i\in V}$
Supervision Layer
Confidence Network
$\{c_j\}_{j\in V}$
Representation Learning
Goodness of instances
$\{\tilde{y}_i\}_{i\in U}$
$\{\tilde{y}_j\}_{j\in V}$
$\{y_j\}_{j\in V}$
$\{\tau_i\} \in U$
Weak Annotator
True Labels

Figure 1: Weak Supervision

The first step in the weak supervision approach involves using the weak labelling algorithm to generate noisy labels for the unlabeled data. The output in Dehghani's Sentiment Classification task for K possible output classes was a 1 x K vector where the sum of the probabilities for every possible class was equal to 1. The weak labelling algorithm is also used to generate noisy labels for the true-labelled data.

Next, for the true-labelled data, the true label and the noisy label are used to generate a "confidence score", which measures the distance between the true label and the weak label. For the Sentiment Classification task in the Dehghani paper, the following algorithm was used to calculate the confidence score for a task with k possible output classes, a true label c_t and a weak label c_w:

$$1 - \frac{1}{k}\sum_{i=1}^{k}|c_i^t - c_i^w|$$

A fully-connected neural network is then trained on the true-labeled data to predict the confidence score of new sentences. This “confidence network” is then used to generate confidence scores for all of the weakly-labeled data.

Finally, a “target” network is then trained on the weakly-labeled data to predict their true labels. This network is generally more complex than the confidence network, e.g. it could include a combination of fully connected layers, convolutional layers, and recurrent layers. The loss function for this network is calculated with respect to the weak labels. However, when the parameters of the model are updated via backpropagation, the gradients are weighted using the predicted confidence score for each unlabeled point thus allowing the model to update errors it has more confidence in greater than other errors in which the weak annotator may not perform as well.

Methods

Weak Supervision on IBC Data

Our primary objective was to use Dehghani et al.’s weak supervision method to beat Iyyer et al.’s accuracy score on the IBC data. We drew our unlabeled data from All the News dataset on Kaggle. Our sample includes 193,744 sentences from two reputably conservative sources (Breitbart and National Review), two reputably liberal sources (CNN and Guardian), and two reputably neutral sources (Reuters and Business Insider).

For the weak annotator, we used a corpus of sentences from articles posted to ideologically-focused Reddit pages from January 2016 to July 2017, the approximate time-frame during which the All The News sample was collected. Specifically, we pull the right-leaning sentences from /r/conservative/ and the left-leaning sentences from /r/liberal/. The labels from the subreddit posts are propagated through the sentences in those posts, similarly to how Iyyer et al. processed the Convote dataset. The final training set contained roughly 39 thousand sentences from each subreddit. We then trained a logistic regression bag of words model to predict whether a given sentence came from a liberal or a conservative subreddit. Finally, this model was used to generate noisy labels for each of the sentences in the All the News sample and the IBC sample.

We employed the same method that Dehghani et al. used to calculate the confidence scores for all of the IBC sentences, and then trained a neural bag-of-words model on the IBC sentences to predict the confidence score. We used the same pre-trained 300-dimensional Google Word2Vec embeddings that Iyyer et al. used to embed the words in the sentences. Then, we used the 95th-percentile sentence length as the maximum length for our training data, and padded the sentences with dummy embeddings per the methods used in assignment 2. We then set the padded indices to 0s in the first layer of the neural network, and summed the remaining embeddings to derive the inputs for the hidden layer.

Given that the confidence score is a continuous number, we used root-mean-squared-error as the cost function in the neural network. The code for this network is available in Weak_Supervision.ipynb.

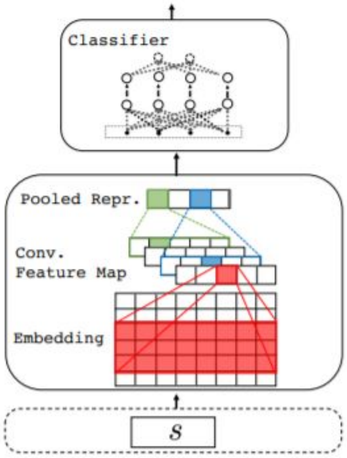

Figure 2: Target Network Architecture

Then, we trained a target network to predict the weak label of the all the news sentences. This network was roughly based on the CNN architecture outlined by Zhang, Y. and Wallage, B. in 2015. (Zhang, Y. & Wallace B. (2015). The network takes a masked embedding of sentences up to a maximum length. The embeddings are the word2vec embeddings published by Google. The embeddings are then fed into a convolutional layer that builds 2, 3, and 4 word convolutions using 128 different filters which are then passed through a RELU layer and max pooled. The max pooled results are then fed into a fully connected layer with a hidden layer size of 100.

The final output is determined by running that output through another layer to produce logits for the two classes [liberal, conservative]. The loss is calculated by passing the logits through a softmax layer and calculating cross entropy of the distributions. During training the network employs dropout rates of 10% at each layer as well as applying the confidence scores as weights to the loss function. These parameters were determined in part based on previous architecture and parameter tuning. The results of the weak supervision approach are listed in the row for WSO in the table below.

To account for the noisiness of the weak label, an additional attempt at weak supervision was conducted with a filtered set of All the News sentences whose weak labels had a probability of 0.9 or higher (WSO - Filter in the results table).

Results

	Method	**Target**	**Accuracy**
1	LR BOW	IBC	60.73%
2	LR BOW	Reddit	63.17%
3	WA	IBC	54.46%
4	WSO	IBC	52.75%
5	WSO - Filter	IBC	53.96%
6	RN	IBC	63.49%
7	RN	Reddit	57.24%

The main baseline for the weak supervision model is the performance of a basic, fully-connected neural network that directly predicts the label of the IBC data (RN - IBC). Using pre-trained word2vec embeddings and a single hidden layer, our model achieved 63% accuracy on the IBC data. This figure is roughly equal to the accuracy achieved by Iyyer et al. using the same baseline approach. The code for this baseline can be found in IBC_Direct_Prediction.ipynb.

To assess the benefit of using neural networks over less complex models, we have also implemented a logistic regression bag of words model to both the IBC data (LR BOW - IBC) and the Reddit data (LR BOW - Reddit) alone. For this model, words were preprocessed using a count vectorizer. We specified the vectorizer to exclude words that appeared only once in the entire corpus to remove noise, and allowed it to create features for unigrams and bigrams. This model achieved 60.73% accuracy on the IBC data, again similar to the accuracy achieved by Iyyer et al.

Although not a part of the primary goal of weak supervision, we decided to assess two models for the reddit data alone, to parallel the models run on the IBC. The logistic regression baseline achieved an accuracy of 60.73%. The data was also evaluated using a fully-connected neural network (RN - Reddit). However, in this case the neural network did worse than the baseline model, only reaching 57.24% accuracy.

Finally, the LR BOW - Reddit model was used as the weak annotator (WA) to produce weak labels and probabilities in preparation for the confidence network. This model itself achieved 54.46% accuracy when predicting the IBC data.

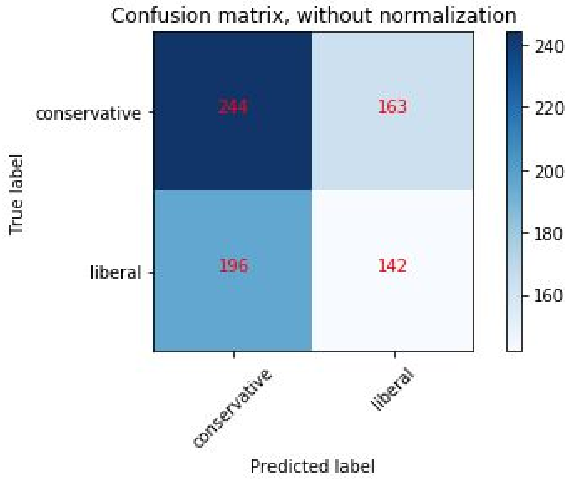

Figure 3: Confusion Matrix for WSO

Conclusion

The potential for weak supervision to assist in a task such as political lean classification is large, but ultimately requires a cleaner data set than what we were able to obtain. There are several issues with the reddit data set that make it insufficient for this task despite its size. One is the assumption that a sentence from a link posted to a political subreddit represents the ideology of that subreddit. This assumption is probably safe for a majority of articles, but there are likely many cases of the opposite being true. For example, if somebody posted a liberal-leaning news article to a conservative subreddit for the sake of debating against it, that article would have been downloaded and all its sentences labeled as “conservative”.

Furthermore, although the sentences from the IBC were collected from journals and books, they generally avoid referring to specific names or current events, and instead represent more general conservative and liberal attitudes. The sentences downloaded from reddit links are very different in this regard. A brief look at the strongest weights associated with either ideology from a logistic regression model reveals that many of the strongest predictors are names, i.e. Bernie Sanders and Ted Cruz predicting liberal and conservative labels respectively. This is a problem that likely affects the All The News dataset as well. Since our training data is so focused on current events and people, it is hard to generalize to a dataset like the IBC.

Finally, there is an inherent problem with taking every sentence from an article and labeling them as all having the same ideological bias. While an article as a whole may have a political lean, that does not necessarily mean that every sentence carries the same detectable political lean. The articles are likely filled with a large number of neutral sentences (i.e. describing an event, or a generic filler sentence), or sentences that are only indicative of political lean in a certain context. This problem is meant to be addressed by assigning sentences like these a lower confidence score. However, due to previously mentioned differences from the news sentences and the IBC sentences, the confidence score may not have filtered out noise as well as we intended.

A curious result from our models was the lower accuracy achieved by our convolutional neural networks on the reddit data compared to our simpler logistic regression baseline. One of the main differences between the preprocessing for each model was the use of word2vec embeddings for the neural network. As a part of the embedding, words that were not in the word2vec model were replaced by a randomly initialized vector of the same size. These unknown words may have included names, places, slang terms, abbreviations, or other features that were vectorized and viewed as useful for the logistic regression model but essentially taken out of the neural network’s vocabulary. While the representation of meaning provided by word2vec has been useful for a number of study, proper implementation for this dataset would likely require a better representation of unknown or rare words.

Future Work

In order to address some of the shortcomings of our model we would propose future work in three main areas:

- Additional work could be done in the area of data gathering. The IBC data provided an excellent dataset for the prediction task of liberal vs conservative however the majority of sentences in news articles do not explicitly contain bias. Either incorporating a neutral class into the model or gathering additional sentences that contain real bias, we believe would lead to higher predictive power.
- We chose to use a CNN architecture for our target network for efficiency purposes and based on previous results in this area and sentiment classification tasks. To improve performance further the target network could incorporate an RNN architecture similar to that employed by Iyyer et al. The target architecture could easily be modified while still utilizing the weak supervision techniques of weighting the losses during training.
- Finally, future work could concentrate on the task of article level prediction. This could take the form of a model that aggregates sentence level predictions to form an overall prediction or could incorporate summarization or article level embedding techniques towards the prediction task.

Works Cited

Iyyer, Mohit, et al. "Political ideology detection using recursive neural networks." *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vol. 1. 2014.

Dehghani, Mostafa, et al. “Avoiding Your Teacher’s Mistakes: Training Neural Networks with Controlled Weak Supervision.” *arXiv preprint arXiv:1711.00313* (2017).

Zhang, YE & Wallace, Byron C. “A Sensitivity Analysis of (and Practictioners’ Guide to) Convolutional Neural Networks for Sentence Classification. “ 2016.

Britz, Denny. “Understanding Convolutional Neural Networks for NLP”. http://www.wildml.com/2015/11/understanding-convolutional-neural-networks-for-nlp/. 2015.

Britz, Denny. “Implmenting a CNN for Text Classifciation in TensorFlow”. http://www.wildml.com/2015/12/implementing-a-cnn-for-text-classification-in-tensorflow/. 2015.