Detecting Political Bias in News Articles through Convolutional and Recurrent Neural Networks

Jason Zhao, Abraham Ryzhik, and Nathaniel Lee
{jzhao23, aryzhik, natelee}@stanford.edu
Stanford University Department of Computer Science
Advisor: Frederic Filloux

Abstract

The aim of this project is to determine effective deep learning models to detect political bias in news articles. We developed two different neural network models that attempt the same classification goal: The first neural network model takes a convolutional approach and the second is structured with a sequential LSTM (long short-term memory) recurrent neural network (RNN) architecture. For our LSTM RNN's, we design both bidirectional and single directional models. Each model takes in as input a series of GloVe vectors representing the article and returns as output whether the article is biased or unbiased (in the case of binary classification) or whether the article is conservative, neutral, or liberal (in the case of three-class classification).

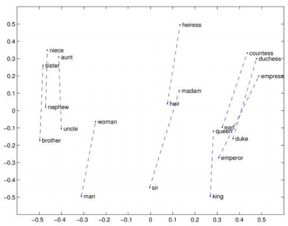
	# Articles	Label
New York Times	8000	Liberal
Atlantic	9000	Liberal
Reuters	10000	Neutral
CNN	10000	Neutral
Fox News	8000	Conservative
Breitbart	7000	Conservative

Data

We used a private dataset collected by our advisor, Frederic Filloux, which included thousands of articles labeled by publication with the text files included. For binary classification, we labeled these publications (and all their articles) as biased or unbiased, and for three-class classification, we labeled the publications liberal, neutral, or conservative.

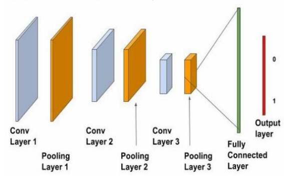
Features

For every article in our dataset, we converted each word into a 100D GloVe vector , which allows us to take advantage of the pre-processed GloVe representations that contain built in relationships with other words in the English language. These GloVe representations allow us to train more quickly, generalize across different syntactical patterns from different articles, and gain a deeper intuition on the meaning of articles due to existing word relationships built into GloVe.

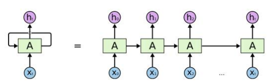


Models

CNN: We trained four convolutional neural networks: one, two, three, and four layer convolutional networks. Each of these four CNN's was trained to perform both binary and three-class classification. Our first layer (after the initial embedding layer used to obtain GloVe vectors) in each CNN was a 1D convolution, followed by a 1D max-pooling layer. This convolution-pooling sequence was repeated for each layer in the CNN's with multiple layers. At the end of each CNN, there was a fully connected (or dense) layer with a ReLu activation, finally followed by a Sigmoid activation (in the case of binary classification) or Softmax (in the case of three-class classification).



LSTM RNN: We trained two sets of three RNN's. Each set contains one, two, and three layer recurrent networks; the first set contains bidirectional layers (which allow for the model to have both backward and forward information at every step) while the second set contains single directional layers. Each of these RNN's was trained to perform both binary and three-class classification. For each sequential layer in the RNN, we implemented dropout regularization. Each RNN ended with one fully connected layer, and then fanally an activation function: either Sigmoid or Softmax.

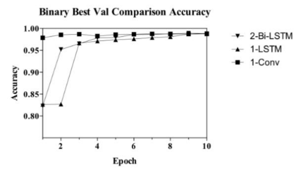


Results

Data Split: Training set included 52252 articles, validation set included 13063 articles, and test set included 19106 articles.

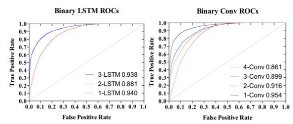
Layers:	1	2	3	4
Conv	81.98%	80.66%	81.28%	83.04%
LSTM	81.31%	81.81%	81.79%	N/A
BiDir	81.68%	81.84%	82.07%	N/A

Table 1: Accuracies of 3-Class Models



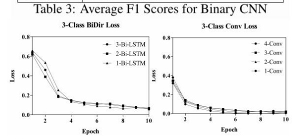
Layers:	1	2	3	4
Conv	96.56%	96.01%	96.59%	96.60%
LSTM	96.20%	96.41%	96.47%	N/A
BiDir	96.39%	96.44%	96.33%	N/A

Table 2: Accuracies of Binary Models



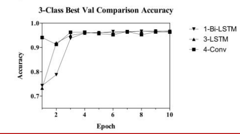
	Conv1	Conv2	Conv3	Conv4
F1	0.85	0.83	0.82	0.81
Precision	0.87	0.85	0.85	0.85
Recall	0.90	0.89	0.88	0.88

Table 3: Average F1 Scores for Binary CNN



Discussion

We discovered that four-layer CNN's were the most accurate bias detectors for our dataset, although LSTM RNN's (both bidirectional and single directional) were quite effective as well. Interestingly, we observed that additional layers could increase the accuracy of model predictions, but often simultaneously lowered precision and recall, perhaps due to imbalances in our dataset.



Future

In the future, we hope to explore nuances beyond simple classification of conservative and liberal, instead performing regression to determine spectrums of economic left/right and political authoritarianism/libertarianism, plotting each article on a continuous 2D Cartesian plane.

References

Lori Young and Stuart Soroka. Affective news: The automated coding of sentiment in political texts. Political Communication, 29(2):205–231, 2012.

Rush Moody. Bias detector: Using language models to identify editorial political slant. Stanford CS229, page 6, 2014.

Arkajyoti Misra and Sanjib Basak. Political bias analysis. CS224n, page 8, 2016.

Yoon Kim. Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882, 2014.

François Chollet et al. Keras. https://keras.io, 2015.