Modeling Bias and Ideological Leaning in Journalism

Bradley Abelman
Purdue University
babelman@purdue.edu

1 Introduction

As ideological leaning and author bias has seeped further into journalism, it is increasingly difficult to determine the credibility of the articles we come across each day. Consuming media without considering the broader context of the author's views and biases hurts the reader and damages their ability to view the news clearly. Identifying the "factuality" and bias of articles was attempted by the Massachusetts Institute of Technology's Computer Science and Artificial Intelligence Lab (CSAIL) in 2018. It was achieved with a fair level of accuracy for determining factuality and political leaning, but this was done on the news organization level, determining if the source as a whole is trustworthy. Several outlets have multiple different journalists, all of whom have differing ideologies and values for objectivity, meaning an orgainization-level evaluation is not ideal. We propose a model that identifies factuality, bias, and political ideology on a per-article basis.

2 Methods

We designed a hybrid model consisting of both a bidirectional LSTM and aspects of a feed-forward neural network. The bidirectional LSTM was the ideal choice for handling the text of the article. It can capture the context of the entire article since it processes the data in both directions, making it more effective than a regular LSTM. It also can capture the complexity of long-term dependencies across the lengthy articles being used as input, meaning that it will vastly outperform a traditional RNN. The other inputs to the model are excluded from the LSTM as to not interfere with the accuracy of the relationships observed by the LSTM. They meet back up with the output of the LSTM in the hidden layer.

We also trained a custom embedding for the text. We had originally intended to use BERT as the embeddings but we were concerned about it being too broad to capture the nuance of political commentary since it was trained on such a wide corpus. The training data we had access to has over 500,000 words, so it is sufficiently large of a corpus for us to use as a basis for the embeddings.

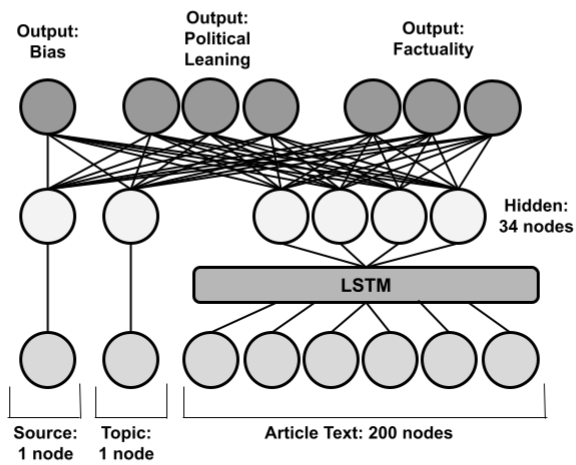

Figure 1: Graph representation of the model.

The model takes three inputs, the topic, the media outlet, and the text of the article. The article text is preprocessed and tokenized, limited at a sequence length of 200 before being used in the model. Then it goes through the embedding layer, reducing the node count to 50. It is then fed through the LSTM with a dropout rate of 20% and reduced to 32 nodes using a ReLU activation function. Now, in the hidden layer these nodes and the other inputs are fed through a sigmoid function for the bias label, a softmax for the 3 political leaning labels, and a different softmax for the 3 factuality labels.

Since there are only two options for the bias label, a softmax function is appropriate. Since there are three outputs for both the political leaning and factuality labels, the softmax function is best for these.

3 Experiments

We used a dataset called “MBIC - A Media Bias Annotation Dataset” found on Kaggle. It has 17,775 annotated articles with bias, factuality, and political leaning labels. Some of the data included with each article in addition to the text are the topic, the media outlet, specific “bias words”, and a link to the article. We chose to only include the outlet and the topic since these are pieces of information that are easy to quickly determine when looking at an article, whereas finding “bias words” would require the user to already have read the article and searched for the words which defeats the purpose of the model.

We tuned several hyperparameters to determine the optimal settings for the model. The first hyperparameter we tuned was the number of epochs. This is because we wanted to find where the accuracy curves stabilized so that we didn’t have to train the model for the other hyperparameters longer than was useful. We found that the ideal number of epochs is 7 since that appeared to be the peak across all 3 accuracy curves. We tested batch sizes of 16, 32, and 64. All of these sizes had nearly identical accuracies for all three metrics: bias, political leaning, and factuality. Since training takes 2 hours for the size 16 batch and 1 hour for both the 32 batch and the 64 batch, it is ideal to use the 32 batch given that it is the fastest batch without a significant drop in accuracy.

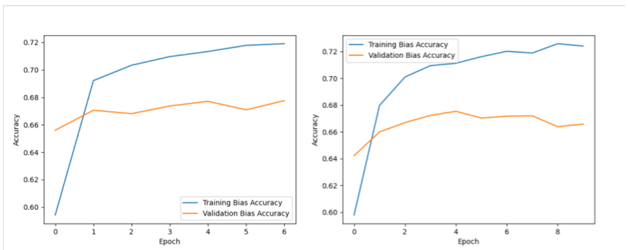

Figure 2: Accuracy of the bias label for batch sizes 16 (left) and 32 (right).

Another hyperparameter that we tuned was the number of embedding nodes. We tried 50, 80, and 120 nodes. All of the accuracies were within 1% of each other, so it appeared that there was not a major difference between the three values. The best accuracy was achieved when there was 50 embedding nodes. This makes sense since we want to reduce the dimension of the output as much as possible without losing the information stored in the text.

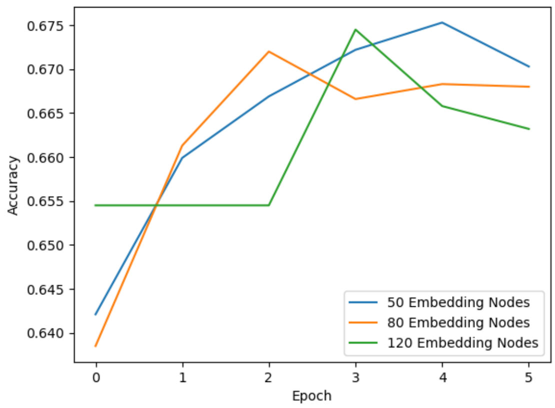

Figure 3: Accuracies of the bias label for 50, 80, and 120 embedding nodes.

We also did an ablation study on the inputs to the model. We trained the model with just the text of the article without any information about the topic of the text or the outlet that the article came from. We were shocked to find that there was almost no change to the accuracies in any of the 3 labels. In fact, the bias accuracy was 0.7% higher than it was in the standard model.

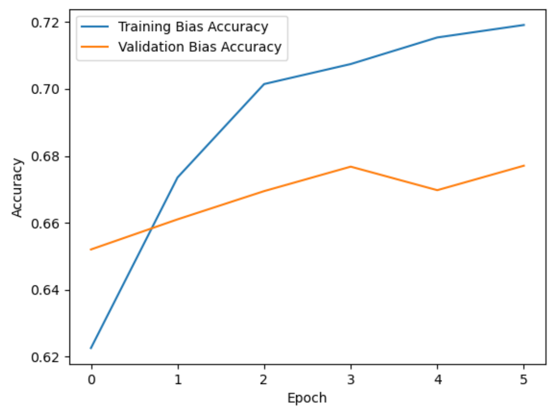

Figure 4: Accuracies of the bias label in the ablation study.

After tuning all of the hyperparameters, we were able to increase the accuracy of the model by between 2 to 3 percent. We found that the bias accuracy was 67.03%, the factuality accuracy was 50.21%, and the political ideology accuracy was nearly 100% (> 99.9%).

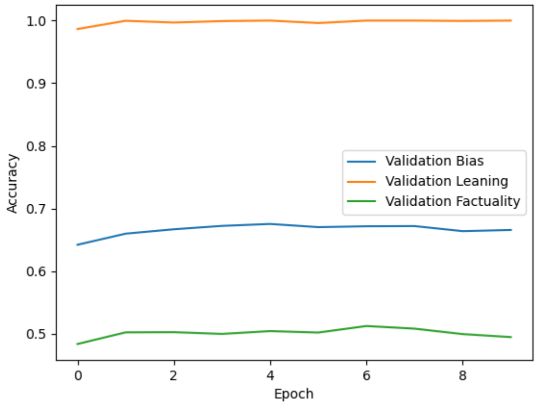

Figure 5: Accuracies of the final tuned model.

4 Discussion

Determining the political leaning of a text is much easier that you might expect. For all versions of the model, the accuracy of identifying the ideology of an article as left, center, or right was nearly 100%. This seems to indicate that there is a substantial difference in the word choice and diction between journalists with different ideologies. We were surprised that even without knowing what the source of the article was, the model had no issue identifying the political leaning.

Determining bias is a much more nuanced thing to do. It can be very subtle. Just slightly changing the order of words or swapping one out for a close synonym can completely change the feel of the article and convert it from an objective standpoint to very opinionated. The accuracy for determining if the article was biased came in at 67%, which is not as high as we had hoped. However, it is unlikely that regular people reading these articles would be able to identify bias accurately in a reasonable way. Their own opinions would likely cloud their judgement and make it nearly impossible for them to get much better than a coinflip accuracy, so perhaps a 67% score for the model is pretty good.

We were somewhat disappointed with the just over 50% accuracy for the factuality. However, since there were three choices ("Completely factual", "Some facts, some opinions", and "Completely opinion"), this accuracy is significantly better than a random guess. It is likely much more difficult for a language model to determine if a text is factual since it would need to identify logical inconsistencies throughout the text. This model is not trained to do that. We expect that it is identifying the non-factual articles by looking for hyperbolic and exaggeratory language that wouldn't fit in a purely fact based article.

If we had significantly more time, possibly a year or two, we would like to integrate a Natural Language Inference model (NLI) into this system. It would take each sentence in the article and compare them, checking for inconsistencies. This would likely boost the accuracy of the factuality label significantly. Additionally, this would make it more likely to differentiate satirical articles from true articles, something that we wanted to do but could not fit well into the model. There is a lot more that can be done to improve this kind of model but this is a great first step in the right direction.