Bias Detection in News Articles

Abstract

These days it's incredibly hard to find a unbiased news source. With most news channels becoming increasingly polarized, they are no longer a source of factual information but merely a tool of the political parties for their own agenda. Bias in news directly translates to bias in the viewers, hence blindfolding them from the facts. Hence this issue is very critical. An external mechanism of detecting bias can be a incredibly useful tool to incentify unbiased news hence undoing the damage to certain extent. Here we present our observations from experiments conducted with various Natural Language models and present a detailed analysis. The dataset we used is available from a shared task 'SemEval-2019 Task 4 Hyperpartisan News Detection'.(Johannes Kiesel, 2019)

1 Introduction

The task it detect whether or not a news article is hyper-partisan. We have experimented with various models, starting with some classical models, then LSTM based and then finally some transformer based models. This was also one of tasks of 'SemEval 2019 - the International Workshop on Semantic Evaluation'. Hence we surveyed the work of all the participating teams with good results and got a very good insight about the available dataset and which models worked best for what kind of tasks. In this paper we present a detailed analysis of this task and some of the best working models.

1.1 Dataset Analysis

The organizers provided two datasets:

- **byarticle**: It is composed of over 1273 articles, manually annotated on the basis of article content.
- **bypublisher**: This is a much larger dataset with over 750k articles annotated on the basis of reported bias of the publisher.

According to majority of participating teams, the 'byarticle' dataset was a high quality one but comprised of relatively few articles. On the other hand 'bypublisher' was a large dataset but a very noisy one. An appropriate reasoning for this could be the fact that not all news articles need to have a scope for bias hence this annotation scheme is extremely noisy and almost always affects model performance negatively.

1.2 Related Work

We did a thorough survey of work done by the participating teams and analysed the following:
How was the data preprocessing done?
Did they make any use of the 'bypublisher' dataset?
If yes, then how and what was the test time performance?
What was the performance of various models they tried?

Many top positions in the final ranking were taken up with classical models. The reason for this is that they only made use of the 'byarticle' dataset and classical models don't require too much data. Despite this challenge we see a lot of deep learning models too. Teams experimented and achieved good results with models like Hierarchical Attention models and BiLSTMs with Attention. 2 teams made use of Transformer based BERT and scored high.

2 Experiments

We decided to set some baselines with classical models and then experiment with various DL models to beat this baseline. The following Results are w.r.t to 2 label classification

2.1 Classical Models

We tested with the following models:

- SVM with Linear Kernel
- SVM with RBF Kernel
- Random Forest
- Adaboost

Results:

Model	LinearSVM	SVM-RBF	Random Forest	Adaboost
Accuracy	0.8	0.72	0.78	0.64
Precision	0.72	0.5	0.67	0.61
Recall	0.75	0.72	0.74	0.64
F1-score	0.73	0.58	0.69	0.62

As we can see, the best performing model is SVM with a linear kernel.

2.2 BiLSTM

We experimented with both Vanilla BiLSTM and BiLSTM with Attention.

2.2.1 Without Attention

The model comprises of a BiLSTM Encoder and a simple fully connected layer as decoder. Final Hidden states of the forward and backward encoder are concatenated and fed to the decoder FC layer. The model was trained on bypublisher dataset. Here, Accuracy of 0.69 was achieved on bypublisher dataset.

2.2.2 With Attention

This is very similar to the last model, except that instead of final hidden states, we use context vector of the entire sequence w.r.t the final hidden state, and give that to the decoder. The model was pretrained on bypublisher dataset. Here, Accuracy of 0.73 was achieved was achieved on bypublisher dataset.

As we can see Model with Attention performed better. (Chiyu Zhang, 2019)

2.3 BERT

BERT is a transformer based model. Its a relatively new model and is known to give SoTA results on lots of NLP tasks. The way we trained this was pretraining on bypublisher, then finetuning on the byarticle dataset. Results: Gave an accuracy of 0.86 on bypublisher.

2.3.1 BERT with content

Here, the content is fed to the bert with a classification head.

2.3.2 BERT with content+title

Here first we get the features representations for both content and title separately and the concatenate and feed them to a two-layer feedforward neural network with 32 neurons in the hidden layer. Results: Gave an accuracy of 0.53 on bypublisher. Here we don't see any improvement as opposed to results mentioned in paper by Jack Ryder, but most likely this appears to be a implementation error. (Daniel Shaprin, 2019)

Analysis: BERT Performed the best on the bypublisher dataset, given by organizers. However,on further digging and literature survey we observed very different results when the test wasn't completely random, but just like it's done in the actual competition. The publishers of train and test taken as a disjoint set. When we did that with 20 percent of publishers in test and rest in train, we saw a severe drop in performance of the model. This suggests that instead of learning to detect bias in the article, the model learned the writing style of the publisher, hence on seeing a new publisher in test set, the model performed poorly.

3 Overall Analysis

This a very interesting problem as it is faced with challenge of using a high quality but low resource data, and a low quality but high resource data. The general way most teams adopted is pretraining with publisher data, then finetuning with byarticle. However some studies suggested that any use of bypublisher only brought down the performance and that by just finetuning the original model with byarticle gave better results

References

Muhammad Abdul-Mageed Chiyu Zhang, Arun Rajendran. 2019. Ubc-nlp at semeval-2019 task 4: Hyperpartisan news detection with attention-based bi-lstm.

Alberto Barron-Cede Preslav Nakov Daniel Shaprin, Giovanni Da San Martino. 2019. Team jack ryder at semeval-2019 task 4.

Rishabh Shukla Emmanuel Vincen Johannes Kiesel, Maria Mestre. 2019. Semeval-2019 task 4: Hyperpartisan news detection.