Joint Classification for Political Bias Prediction

Vishal Rajkumar   Varsha Venkata Krishnan   Satya Sai Bharath Vemula
Department of Computer Science, Purdue University, West Lafayette.
rajkumav@purdue.edu, venka104@purdue.edu, vsatyasa@purdue.edu

Abstract

The use of joint models in predicting multiple attributes simultaneously has been shown to improve performance and reduce training time when correctly modeled. For example, predicting political bias and other attributes such as the topic and source of the news can be done more efficiently and accurately through joint modeling. However, the selection of which attributes to model together, the model architecture, and the choice of text representation are all important factors that can affect the performance of such models. In this work, we study and demonstrate the importance of these factors on a political bias dataset, showing how changing the second attribute being predicted, the model architecture, and the learning representation can impact the performance of bias prediction. This research has important implications for the development of more efficient and accurate models for predicting multiple attributes in news data.

I. Introduction

Joint classification models are a promising approach for predicting multiple output variables in machine learning. Unlike traditional binary classification models, they can estimate the probabilities of multiple output variables. This can be particularly useful in natural language processing, where joint models can handle multiple tasks simultaneously or sequentially. These models use shared linguistic features to enhance performance and generalization capabilities.

Correctly modeling joint classification models can lead to significant improvements in both accuracy and training time. However, careful consideration must be given to various factors such as model architecture, the selection of appropriate attributes to model together, and choosing the right learning representation. If these factors are not selected correctly, the performance of joint models may deteriorate instead of improving. Therefore, it is essential to tune and optimize these factors to build accurate and efficient joint models. By doing so, we can leverage the shared features across multiple tasks and improve the performance of our models, leading to better efficiency and use of data.

In this work, we focus on the use of joint models for predicting political bias and other attributes such as topic and source in news data. We study and demonstrate the importance of selecting the appropriate attributes to model together, the model architecture, and the learning representation for improving the performance of bias prediction. Specifically, we conduct experiments on a political bias dataset to investigate the impact of these factors on joint modeling performance. Our results highlight the importance of carefully selecting attributes and designing models for improving the accuracy and efficiency of joint models for news data prediction.

II. Research Question

In the context of the joint modeling for political bias dataset, we answer the following research questions

1. Does joint-modeling for a political bias dataset improve the performance in terms of accuracy and time ?

2. If yes, Is there any importance of relation between two attributes that are learnt together ?

3. How does different representation learning and model architecture affect the performance improvement ?

III. Methodology

The key problem in any textual classification is learning the representation of the sentence, which significantly affects the problem. In our case, we are attempting to learn a representation that can aid in joint classification. As a part of this work, we experimented with LSTM (which views each news article as a sequence of tokens) and BERT (which uses an attention mechanism for structured learning) to contrast different types of representation learning approaches. More details on the different architectures and attributes on which we ran experiments are explained in the Experiments section.

We faced two challenges in applying these methods. The first challenge was with the availability of the dataset. Initially, we wanted to experiment with Political Bias - Sentiment based joint classification. However, since none of the publicly available datasets had sentiment data tagged, we came up with a hybrid pseudo-labeling approach to label the news data with sentiment tags. The details of this approach are mentioned in the Dataset Preparation section.

To verify the validity of our methods, we wanted to run the experiments multiple times and calculate the average accuracy increase. However, since BERT and LSTM were computation-intensive tasks, we could only test the applied methods on a sub-sample of the dataset, which had a uniform distribution among all the classes.

IV. Dataset

A. Dataset Description

The dataset selected for the experiments was sourced from a well-known study [1], which collected the data from Allsides (http://allsides.com/). The dataset has a cardinality of 34,737 data samples and is publicly available. The dataset is pre-augmented with many interesting attributes such as topic, bias, source, and cleaned text (for NLP experiments). The pre-augmented attributes (topic and source) are used in our experiments to answer the aforementioned research questions.

B. Dataset Preparation

The dataset consists of 34,737 articles published by 73 news media outlets and covering 109 topics. Due to computational limitations, we selected a subset of 4,070 data samples where each article's length was less than 512 tokens. After subsampling the data, our dataset covered data from 105 topics and 60 news media outlets.

C. Pseudo-Labelling of sentiment data

In all the datasets we explored, there were no sentiment labels available. Therefore, we used a form of pseudo-labeling based on the predictions of three state-of-the-art models: XLNet, RoBERTa, and XLM-RoBERTa, which were trained on different datasets for sentiment analysis. This approach allowed us to generate approximate sentiment labels through majority voting.

1) Hybrid pseudo-labeling approach:

The first step of the active learning approach for ideology and sentiment analysis involves training three sentiment analysis models on labeled data. Next, a subset of unlabeled articles is selected from the dataset. The three trained models are then used to predict the sentiment of each unlabeled article, and the majority vote of the three models' predictions is used to determine the final sentiment label for each article. The sentiment label is then assigned to the corresponding article in the dataset. The labeled sentiment data is combined with the original labeled data to create a new labeled dataset.

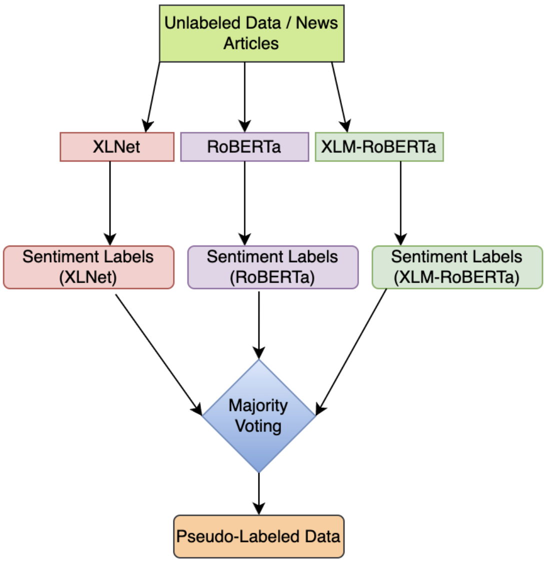
Unlabeled Data / News Articles
XLNet
RoBERTa
XLM-RoBERTa
Sentiment Labels (XLNet)
Sentiment Labels (RoBERTa)
Sentiment Labels (XLM-RoBERTa)
Majority Voting
Pseudo-Labeled Data

Fig. 1: Pseudo-Labelling of sentiment data.

Since the sentiment labels from each of XLNet, RoBERTa, and XLM-RoBERTa will not be accurate, we do majority voting by combining the predictions of these models to improve the overall performance and reliability of the final output as shown in (Figure 1). majority voting is employed to enhance the accuracy, robustness, and reliability of the final output by leveraging the strengths of multiple models, capitalizing on their diversity and complementarity, and reducing the impact of individual errors or biases.

D. Dataset Statistics

As mentioned we took 4070 samples out of the 34,737 articles present and the size of each of the article selected are less then or equal to 512 token. The attributes and the number of classes are provided in the Table below (Table 1).

TABLE I: Attribute and Number of Classes

Attribute	Number of Classes
Bias	3
Topic	105
Sentiment	2
Source	149

V. Experiments

This section outlines the various experiments conducted in predicting bias jointly with other attributes. The following three sub-sections provide information on the different variations of model architecture for joint learning, representation learning, and joint attributes, respectively.

A. Model Architectures

We tried two different architectures for joint modeling, described in Figure 2 and Figure 3.

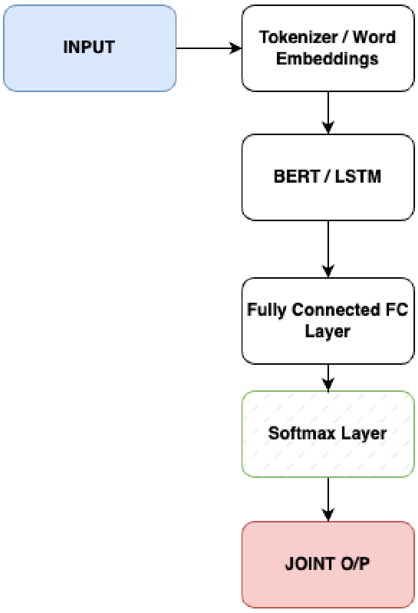

Fig. 2: Joint FC layers

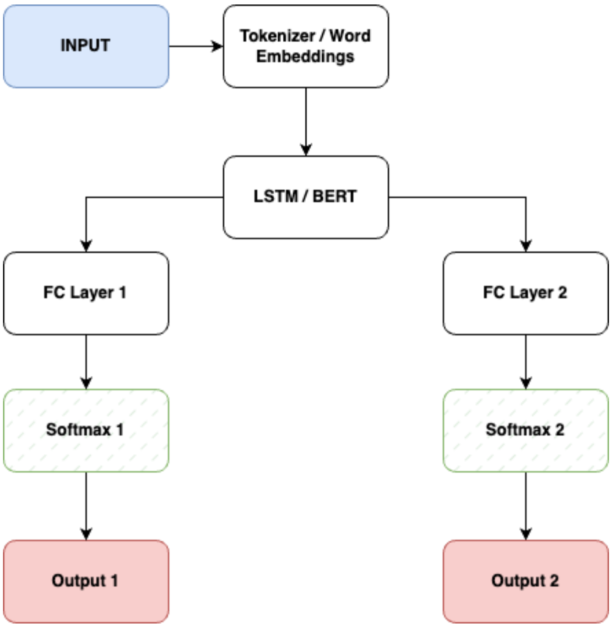

Fig. 3: Separate FC layers

- In the first architecture (Figure 2), the two classes that are modeled together are combined into one fully connected layer. If the number of class labels in each of the attributes is m and n, the architecture has m*n outputs. Each of the outputs correspond to a combination of classes from both attributes.
- In the second architecture (Figure 3), the two classes that are modeled together, are represented by using separate fully connected layers. If the number of class labels in each of the attributes is m and n, the architecture has m+n outputs.

B. Representations

We have experimented with two different representations - LSTM and BERT. For LSTM based approach, GloVe embeddings were used and BERT tokenizer was directly used for BERT based approach.

C. Attribute selection

We tried modeling bias with three different attributes - sentiment, bias and source.

VI. Results

The section describes the results presented in figures 4, 5, and 6. For each joint-modeling result introduced below, all experiments and corresponding results are included in the appendix section.

Regarding joint modeling for bias-sentiment, we conducted experiments using both LSTM and BERT for representation learning and utilized the two model architectures described in section 4.1. Figure 4 presents a comparison of the joint modeling results of our best model, selected based on hyper parameter tuning, to a baseline model trained using LSTM/BERT with a single fully connected layer at the end.

Regarding joint modeling for bias-topic and bias-source, we conducted experiments only using both LSTM for representation learning and utilized the two model architectures described in section 4.1. Figure 5,6 presents a comparison of the joint modeling results of our best model for respective joint modeling, selected based on hyper parameter tuning, to a baseline model trained using LSTM with a single fully connected layer at the end.

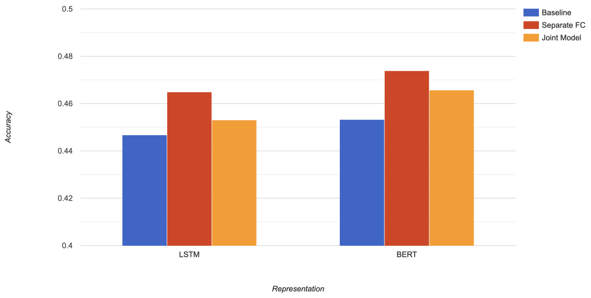

Fig. 4: Performance of bias prediction when modeled with sentiment

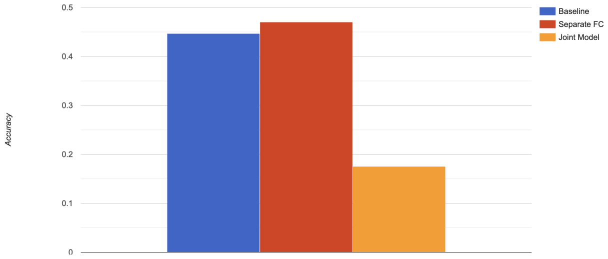

Fig. 5: Performance of bias prediction when modeled with topic

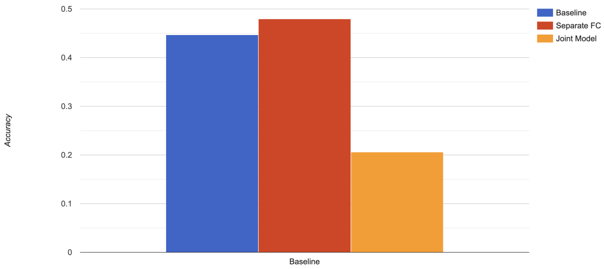

Fig. 6: Performance of bias prediction when modeled with source

VII. Findings

This section briefly describes the findings and observations made based on the results obtained across the set of experiments (all the results of experiments mentioned in appendix section), also answers the research questions raised in the Research Questions section.

1. Does joint-modeling for a political bias dataset improve the performance in terms of accuracy and time?
From figures 4, 5, and 6, it can be clearly observed that joint modeling has boosted the performance of bias prediction across all joint modeling combinations of separate fully connected layer based models and for the joint fully connected layer for bias-sentiment prediction. In terms of time, we have noted a speedup of 1.7x when modeled jointly, in contrast to training the baselines separately.

2. If yes, Is there any importance of relation between two attributes that are learnt together ?

Attribute selection greatly affects the performance of the model. When we performed CHI-Square test we have observed that the co-relation between bias and sentiment / topic / source are in the order of source, topic and sentiment in descending order.

Highly correlated attributes provide a better performance boost than low correlation attributes. This can be attributed to direction of loss, which is boosted better when we train highly correlated attributes.

3. How does different representation learning and model architecture affect the performance improvement ?

From Figure 4,5,6, we can see that modeling the attributes as separate FC layers boosts the performance of bias prediction as the accuracy values are higher than the baseline accuracy. However, modeling them using a combined FC layer, represented as 'Joint Model' in the results, give a slight boost if modeled with sentiment and reduce the performance for other two attributes. This could be attributed to the number of classes in the joint labels. Number of labels is much lesser in the sentiment-bias combination when compared to topic-bias and source-bias combinations and the dataset is too small to learn 150+ output labels. With a larger dataset, the joint architecture could behaviour differently.

VIII. Source Code

The source code for the data preparation and experiments can be found at: https://github.com/vsatyasa/Joint-Political-Bias-Prediction

IX. Conclusion

The observations made from the experiments conducted as a part of this work has yielded the following inferences that training joint modeling could improve performance in terms of both accuracy and training time. It is very interesting to observe that having a separate fully connected layer based architecture has consistently better performance for joint modeling tasks even when cardinality of the number of classes increased and the performance boost is directly proportional to correlation between two attributes being learnt along. Though correlation also played some key in improving the performance for single layer based joint prediction the performance started dropping as the number of classes started to increase.

The experiments conducted in this study have been done on a relatively smaller dataset. It would be interesting to see if the same trends hold true for different datasets and sizes. Also the experiments haven't touched less correlated attributes to be predicted alongside. It would be interesting to experiment joint modeling with more than two attributes at the same time as a part of future work and observe the trends.

X. Proposal vs Project

This section briefly contrasts from the experiment's commitment made in the proposal phase and the final implementation of the project.

The commitment made in the proposal phase was to explore joint modeling of bias detection with sentiment data and evaluate if it would improve performance while analyzing the reasons for the improvement.

In the project phase, this commitment was fulfilled, and further combinations of joint modeling for bias-topic, bias-sentiment, and bias-source were explored. Their performance was compared, and reasoning was provided on why one approach worked better than the other across two joint modeling architectures. Additionally, rules were proposed that could be applied to any general joint model classification.

Though initially, we only wanted to explore bias-sentiment joint classification, we became curious about why these results worked. This led us to explore other attributes and examine their results. We then developed a framework of rules for joint modeling tasks

References

Qian Chen, Zhu Zhuo, and Wen Wang. 2019. BERT for joint intent classification and slot filling. *CoRR*, abs/1902.10909.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

XI. Appendix

Hidden nodes	Learning rate	Bias Accuracy	Loss
25	1.00E-03	0.4466	1.1049
50	1.00E-03	0.4226	1.1289
100	1.00E-03	0.4201	1.0836
200	1.00E-03	0.3963	1.1050
25	1.00E-05	0.4405	1.0744
25	1.00E-04	0.4467	1.0575
25	1.00E-03	0.4160	1.0742
25	1.00E-02	0.4437	1.1077
25	1.00E-01	0.4437	1.1077

TABLE II: Baseline performance of LSTM

Hidden nodes	Learning rate	Bias accuracy	Loss
25	1.00E-03	0.3133	1.2382
50	1.00E-03	0.3133	1.2382
100	1.00E-03	0.3133	1.2382
200	1.00E-03	0.3245	1.2453
300	1.00E-03	0.4533	1.0981

TABLE III: Baseline performance of BERT

Hidden nodes	Learning rate	Bias accuracy	Sentiment accuracy	Loss
25	1.00E-03	0.4533	0.6462	3.1656
50	1.00E-03	0.4453	0.6437	3.1564
100	1.00E-03	0.4674	0.6412	3.1972
200	1.00E-03	0.4739	0.6566	3.1810
300	1.00E-03	0.4437	0.6491	3.1723

TABLE IV: Performance of BERT having separate FC layers for bias and sentiment

Hidden nodes	Learning rate	Joint accuracy	Loss
50	1.00E-05	0.0993	5.0023
50	1.00E-04	0.1145	4.9829
50	1.00E-03	0.1686	4.8476
100	1.00E-04	0.1722	4.9259
100	1.00E-03	0.1686	4.8469
200	1.00E-04	0.1686	4.8609
200	1.00E-03	0.1686	4.8469

TABLE V: Performance of LSTM having joint FC layer for bias and source

Hidden nodes	Learning rate	Bias accuracy	Sentiment accuracy	Loss
25	1.00E-03	0.4466	0.6505	3.1681
50	1.00E-03	0.4466	0.6505	3.1681
100	1.00E-03	0.3889	0.6553	3.1386
200	1.00E-03	0.4157	0.6491	3.1829
50	1.00E-05	0.4376	0.6489	3.1078
50	1.00E-04	0.4649	0.6899	3.0544
50	1.00E-03	0.4450	0.6491	3.1301
50	1.00E-02	0.4437	0.6491	3.1723
50	1.00E-01	0.4437	0.6491	3.1723

TABLE VI: Performance of LSTM having separate FC layers for bias and sentiment

Hidden nodes	Learning rate	Joint accuracy	Loss
25	1.00E-03	0.2930	1.7506
50	1.00E-03	0.2930	1.7506
100	1.00E-03	0.2779	1.7490
200	1.00E-03	0.2904	1.7503
50	1.00E-05	0.2899	1.7527
50	1.00E-04	0.2953	1.7239
50	1.00E-03	0.2764	1.7301
50	1.00E-02	0.2904	1.7532
50	1.00E-01	0.2904	1.7532

TABLE VII: Performance of LSTM having joint FC layer for bias and sentiment

Hidden nodes	Learning rate	Joint accuracy	Loss
25	1.00E-03	0.1941	1.8495
50	1.00E-03	0.1941	1.8495
100	1.00E-03	0.3133	1.2382
200	1.00E-03	0.1941	1.8495
100	1.00E-05	0.2998	1.7327
100	1.00E-04	0.2998	1.7438
100	1.00E-03	0.1941	1.8495
100	1.00E-02	0.2998	1.7438
100	1.00E-01	0.1941	1.8495

TABLE VIII: Performance of BERT having joint FC layer for bias and sentiment

Hidden nodes	Learning rate	Bias accuracy	Topic accuracy	Loss
50	1.00E-05	0.4386	0.0204	9.0222
50	1.00E-04	0.4636	0.0521	8.9691
50	1.00E-03	0.4437	0.1273	8.8573
100	1.00E-05	0.4435	0.0381	8.9945
100	1.00E-04	0.4700	0.0830	8.9452
100	1.00E-03	0.4437	0.1273	8.8559
200	1.00E-05	0.4437	0.0378	8.9698
200	1.00E-04	0.4690	0.1273	8.9005
200	1.00E-03	0.4437	0.1273	8.8559
300	1.00E-05	0.4437	0.0835	8.9609
300	1.00E-04	0.4437	0.1273	8.8676
300	1.00E-03	0.4437	0.1273	8.8559

TABLE IX: Performance of LSTM having separate FC layers for bias and topic

Hidden nodes	Learning rate	Joint accuracy	Loss
50	1.00E-05	0.0307	5.6729
50	1.00E-04	0.0518	5.6669
50	1.00E-03	0.0600	5.6216
100	1.00E-05	0.0516	5.6724
100	1.00E-04	0.0555	5.6555
100	1.00E-03	0.0600	5.6197
200	1.00E-05	0.0563	5.6694
200	1.00E-04	0.0597	5.6276
200	1.00E-03	0.0600	5.6193
300	1.00E-04	0.0582	5.6231
300	1.00E-03	0.0543	5.6244

TABLE X: Performance of LSTM having joint FC layer for bias and topic