Abdygulov Timur

NLP Individual Project Report

Introduction:

News reporting is increasingly influenced by the biases of authors and media organizations, posing a challenge to the public's access to unbiased information. Our project aimed to address this challenge by developing a pipeline for media consumers to detect bias in news articles. We compared various text classification models and integrated transformer models for abstractive summarization to provide users with center-biased summaries of news articles.

Utilizing a labeled dataset provided by Baly et al. (2020), we explored classical and transformer-based models for political bias classification. Additionally, we integrated a pretrained Pegasus model for abstractive text summarization.

I will talk about my contributions to the effort in this report. Namely the RoBerta model, and the app development for interactive presentation.

Experimental Design:

We adopted an iterative approach to model design, starting with simpler models and progressing to more complex ones. This allowed us to identify the optimal solution for classifying political bias in news articles. I began with an MLP (Multi-Layer Perceptron) model using a custom data loader and word2vec embeddings. We systematically evaluated each model using a split dataset (training, validation, and testing) and performance metrics like accuracy and F1-score. The best-performing model was ultimately deployed into our Streamlit application for real-time bias detection and summarization.

Description of My Contributions:

I played a pivotal role in developing and evaluating text classification models and integrating the Pegasus summarization functionality as well as trained RoBERTa transformer model on the found dataset into our Streamlit application.

In building a strong foundation for our project, I initiated the model development process by constructing baseline models, Naive Bayes, and Logistic regression, of increasing complexity. This initial exploration provided valuable insights into model selection, guiding our subsequent efforts toward more sophisticated approaches. The results of the classical NLP models on the test set can be seen below in Table 1.

Table 1: Results on the test set of classical NLP models.

Naive Regression	Logistic Regression
Accuracy: 0.541	Accuracy: 0.71
F1-score (weighted): 0.504	F1-score (weighted): 0.71

A significant portion of my work involved fine-tuning a pre-trained RoBERTa base model for our specific task of political bias classification, as well as developing and integrating the model into our Streamlit application. After multiple pieces of training, I found the best-performing tuning parameters they are: 7 epochs, 3e-5 points of learning rate, and a batch size of 32. This process required meticulous data loading, preprocessing using a RoBERTa tokenizer, and fine-tuning the model's architecture. In Table 2 below you will find the screenshot showing a training process and its results. The code I developed for this task exhibited excellent structure, organization, and documentation, facilitating collaboration among team members. Our approach yielded promising results, achieving high accuracy and F1-score, as demonstrated in the confusion matrix Table 3 and Table 4 of RoBERTa results on test and validation data sets.

Table 2: RoBERTa training process screenshot

```
Map: 100%|██████████| 30043/30043 [02:33<00:00, 195.20 examples/s]
Map: 100%|██████████| 3755/3755 [00:20<00:00, 182.79 examples/s]
Map: 100%|██████████| 3756/3756 [00:20<00:00, 179.83 examples/s]
Some weights of RobertaForSequenceClassification were not initialized from the model checkpoint at roberta-base and are
You should probably TRAIN this model on a down-stream task to be able to use it for predictions and inference.
 14%|█▍        | 939/6573 [12:49<1:13:43,  1.27it/s]Epoch: 01 | Train Loss: 0.800 | Val Loss: 0.483 | Val Acc: 81.15%
 29%|██▊       | 1878/6573 [26:12<1:01:25,  1.27it/s]Epoch: 02 | Train Loss: 0.381 | Val Loss: 0.353 | Val Acc: 86.63%
 43%|████▎     | 2817/6573 [39:34<49:05,  1.28it/s]Epoch: 03 | Train Loss: 0.300 | Val Loss: 0.325 | Val Acc: 87.94%
 57%|█████▋    | 3756/6573 [52:57<36:50,  1.27it/s]Epoch: 04 | Train Loss: 0.239 | Val Loss: 0.316 | Val Acc: 88.39%
 71%|███████▏  | 4695/6573 [1:06:20<24:32,  1.28it/s]Epoch: 05 | Train Loss: 0.160 | Val Loss: 0.309 | Val Acc: 90.33%
 86%|████████▌ | 5634/6573 [1:19:42<12:16,  1.28it/s]Epoch: 06 | Train Loss: 0.086 | Val Loss: 0.313 | Val Acc: 91.03%
100%|██████████| 6573/6573 [1:33:04<00:00,  1.28it/s]Epoch: 07 | Train Loss: 0.038 | Val Loss: 0.364 | Val Acc: 91.03%
```

Table 3: RoBERTa confusion matrix.

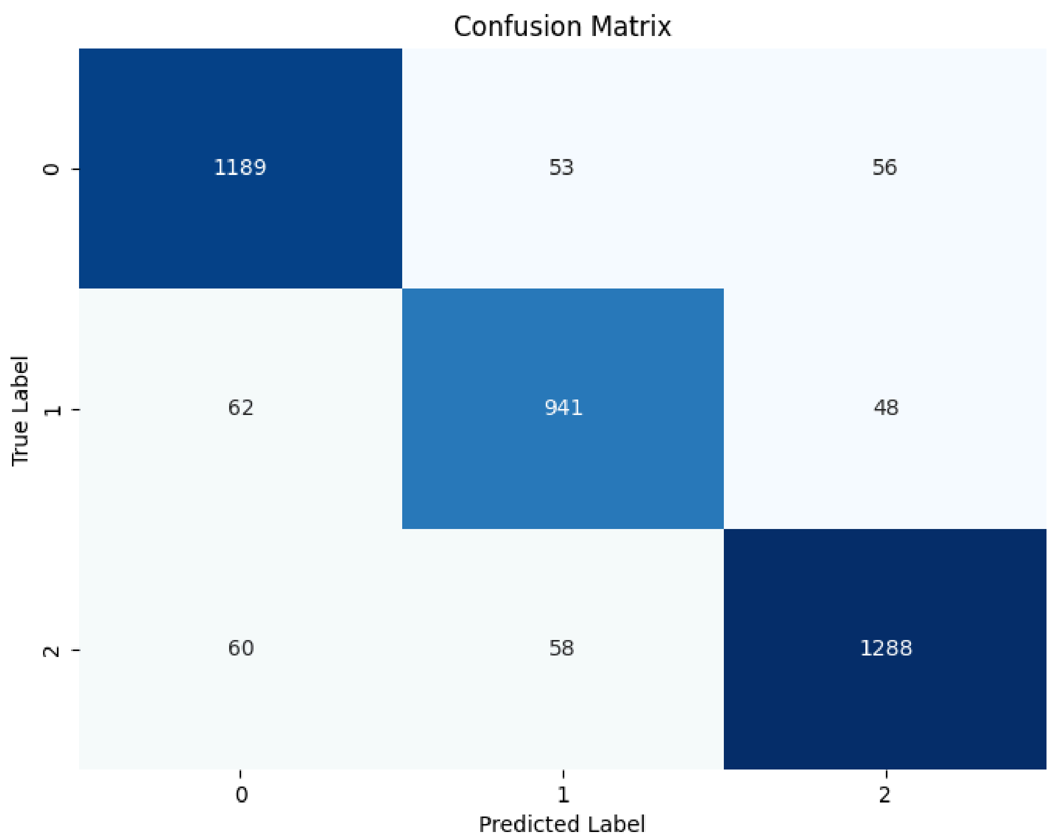

Table 4: RoBERTa results on test and validation data sets.

Validation Set Evaluation:	Test Set Evaluation:
Accuracy: 0.9102529960053263	Accuracy: 0.906283280085197
F1 Score: 0.9102723449081925	F1 Score: 0.9063117528773919

Furthermore, I collaborated with teammates to develop the Streamlit application, where I played a central role in integrating the Pegasus model for abstractive summarization and trained the RoBERTa model. This involved ensuring a clear separation of concerns and logical flow within the code, which enhanced readability and usability. Additionally, I contributed to other aspects of the application's development, including user interface design and implementing interaction features. By integrating summarization alongside bias detection, we significantly enhanced the user experience, making news articles more consumable for readers.

Conclusion:

Our team successfully evaluated multiple NLP models for political bias classification and deployed a fine-tuned transformer model into a user-friendly Streamlit application. The integration of abstractive summarization enhanced the utility of the application, providing users with concise summaries alongside bias detection.

In this project, I played a key role in achieving the highest accuracy and F-1 score during the training of the RoBERTa transformer model. I designed a well-structured Python script with detailed comments to facilitate clear communication and understanding among team members. Additionally, I independently developed a Streamlit application, ensuring its functionality and user-friendly interface.

Furthermore, the project yielded promising results, with high accuracy and F1-scores attained through the evaluation of multiple NLP models for political bias classification. Specifically, the fine-tuned RoBERTa transformer model demonstrated the highest performance. Integrating abstractive summarization using the Pegasus model enhanced the user experience in our Streamlit application.

For future improvements, expanding the dataset, exploring ensemble learning techniques, refining the summarization model, and conducting user feedback surveys could enhance the application's effectiveness and usability.

Citations:

Baly, R., Da San Martino, G., Glass, J., Nakov, P. (2020). We can detect your bias: Predicting political ideology of news articles. Archive. https://arxiv.org/pdf/2010.05338

Zhang, J., Zhao, Y., Saleh, M., & Liu, P. (2020, November). Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In International conference on machine learning (pp. 11328-11339). PMLR.

Liu, Yinhan, et al. "Roberta: A robustly optimized bert pretraining approach." arXiv preprint arXiv:1907.11692 (2019).

Kamath, Cannannore Nidhi, Syed Saqib Bukhari, and Andreas Dengel. "Comparative study between traditional machine learning and deep learning approaches for text classification." Proceedings of the ACM Symposium on Document Engineering 2018. 2018.