Evaluating Modern Transformer Models for Multi-Label Propaganda Classification

Minal Arunashalam
marunash@gmu.edu

Joniel Augustine Jerome
jjerome4@gmu.edu

Akhil Krishnan
akrish6@gmu.edu

1 Recap: Brief Project Description

This project focuses on detecting fine-grained propaganda techniques in political news articles. Prior work introduced large annotated datasets and demonstrated that propaganda often involves multiple techniques within a single sentence, making the task inherently multi-label (Da San Martino et al., 2019). Earlier approaches, including those from the SemEval-2020 shared task, emphasized span-level identification, where systems locate the exact propagandistic text spans (Da San Martino et al., 2020). In contrast, our work focuses on sentence-level classification, predicting which techniques appear in a sentence without requiring span extraction.

To build this system, we implemented a full training and evaluation pipeline using several transformer-based models, including RoBERTa-base, DeBERTa-v3-base, and a QLoRA-fine-tuned LLaMA-3.1-8B-Instruct. Our pipeline handles preprocessing, tokenization, dataset loading, fine-tuning, and inference, and reports micro and macro F1 scores. Prior work has shown that fine-tuning transformer models is effective for propaganda detection, motivating our architectural choices (Yoosuf and Yang, 2019).

Overall, the project extends existing research by shifting the focus to sentence-level multi-label prediction and by comparing the performance of multiple modern transformer architectures on this task.

2 Detailed System or Methodology Description

Our system is built around the SemEval 2020 Task 11 dataset, which contains 446 political news articles annotated with approximately 7,500 spans covering fourteen different propaganda techniques. These articles originate from a variety of mainstream news outlets, ensuring diversity in political perspectives and writing styles. Because SemEval provides annotations at the span level (character spans within which propaganda occurs) rather than at the sentence level, our first step was to convert the data into a form suitable for multi-label classification. Each article was segmented into sentences, and each sentence was assigned a vector of fourteen binary indicators corresponding to the presence of the associated techniques. This transformation simplified the prediction task by shifting from span extraction to sentence-level classification.

The resulting processed dataset served as the input for all of our models, which were trained on the same three CSV files. We implemented our system using the Hugging Face Transformers library for model configuration, PyTorch for model training, and Scikit-learn for evaluation metrics. For parameter-efficient fine-tuning, we used QLoRA along with the `bitsandbytes` and `accelerate` libraries to enable quantized training and reduce GPU memory usage. All experiments were conducted in Google Colab using A100 or T4 GPUs, with the full pipeline requiring roughly five to six hours of compute time.

We trained three primary models: RoBERTa-base, DeBERTa-v3-base, and a LoRA-fine-tuned version of LLaMA-3.1-8B-Instruct. The first two serve as encoder-based baselines, which are commonly strong performers in sentence classification tasks. The LLaMA model was included to investigate how well a decoder-only architecture can handle multi-label classification when trained with parameter-efficient fine-tuning. Each model followed a standard training loop using class-weighted binary cross-entropy loss (BCEWithLogitsLoss with per-label pos_weight) to compensate for strong label imbalance. We monitored training and validation loss across epochs to iden-

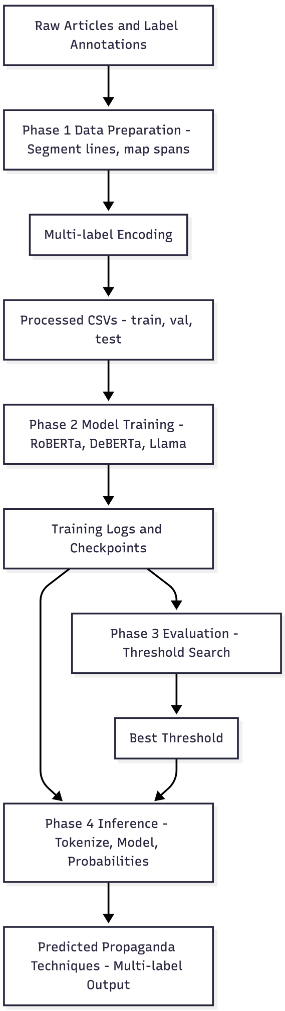
Raw Articles and Label Annotations
Phase 1 Data Preparation - Segment lines, map spans
Multi-label Encoding
Processed CSVs - train, val, test
Phase 2 Model Training - RoBERTa, DeBERTa, Llama
Training Logs and Checkpoints
Phase 3 Evaluation - Threshold Search
Best Threshold
Phase 4 Inference - Tokenize, Model, Probabilities
Predicted Propaganda Techniques - Multi-label Output

Figure 1: Overview of our end-to-end system workflow, including data preparation, multi-label encoding, model training, evaluation, and inference.

tify the strongest performing checkpoints for final evaluation. RoBERTa and DeBERTa are both loaded with 14 output labels using Hugging Face's roberta-base and microsoft/deberta-v3-base. Both models load their corresponding pretrained tokenizers from Hugging Face and tokenize inputs to a maximum sequence length of 256, and are fine-tuned with AdamW optimizer for 5 epochs at a learning rate of 2×10^{-5}. The only difference in their setup is the batch size (8 for RoBERTa and 6 for DeBERTa), because DeBERTa has higher parameter count and memory usage as it is a larger model.

For LLaMA, we use a meta-llama/Meta-LLaMA-3.1-8B-Instruct model as a 14-label classifier with the same data and max sequence length, but use a much smaller batch size (2) and a higher learning rate (1×10^{-4}) over only 2 epochs, due to computational limitations. To make 8B-parameter fine-tuning feasible on a single GPU, we use QLoRA: the base model is loaded in 4-bit quantization so it uses much less memory. 4-bit quantization means that the model uses 4 bits for each weight value instead of 32, which makes the model much smaller and faster, at the cost of losing some precision in the numbers. Then, instead of updating all 8 billion model weights, we train only small LoRA adapter layers (rank 8) inserted into the model's attention projections, while the rest of the model stays frozen. This lets us fine-tune an 8B model efficiently without sacrificing too much performance. The source code of our methodology can be accessed from https://github.com/Minal-Arunashalam/propaganda-detector

3 System/Methodology Evaluation

3.1 Experimental Setup

For our system, we trained and evaluated our three models on the same train/validation/test splits of the processed CSV dataset that we created after our sentence-level processing of the SemEval 2020 Task 11 data. This ensured that we had a consistent comparison of how each model performed on the same data. For our evaluation metrics, we focused on multi-label classification performance measured through micro and macro-averaged F1 scores. For each model, our evaluation script runs the model once over the entire test set and collects the raw logits (raw prediction values) and the truth label vectors.

These logits are then passed through a sigmoid to obtain per-label probabilities in [0,1], which we convert into binary predictions by thresholding. Because multi-label classification is highly sensitive to the choice of threshold, we go through a set of threshold values: for RoBERTa and DeBERTa we evaluate thresholds from 0.1 to 0.9 in increments of 0.1, and for LLaMA we use a grid of smaller thresholds (0.005–0.30) to account for different logit scales induced by quantization and LoRA.

For each threshold, we compute micro-averaged and macro-averaged precision, recall, and F1 using scikit-learn, treating each label in each example as an independent binary decision. Micro-averaged metrics tell us how good the model is overall, weighting every (example, label) decision equally, so frequent labels dominate. Macro-averaged metrics tell us how balanced the model is across labels, by averaging performance per label so rare and frequent techniques count the same, better reflecting how well the system handles rare techniques.

3.2 Experimental Results

Table 1 displays the training and validation loss obtained for each model after their final training epochs. RoBERTa achieved the lowest training loss and a mid-range validation loss, while DeBERTa showed a slightly higher loss on both metrics, consistent with their similar overall behavior. Interestingly, LLaMA-3.1-8B obtained the lowest validation loss despite having by far the highest training loss, but only slightly, suggesting that the model did not overfit but also did not fully learn the task.

Table 2 presents the best performance metrics for each model on the test set, including micro and macro precision, recall, and F1 scores. The reported threshold for each model corresponds to the minimum prediction score required for a label to be classified as positive, reflecting the point at which each model achieves its strongest overall performance.

On the test set, RoBERTa achieves a micro F1 of 0.3790 with micro precision 0.3844 and micro recall 0.3737 at a threshold of 0.70, while DeBERTa achieves a very similar micro F1 of 0.3780, trading slightly lower precision (0.3698) for slightly higher recall (0.3866) at a threshold of 0.80. RoBERTa had higher macro precision (0.2948) but lower macro recall (0.1938), while DeBERTa reaches higher macro recall (0.2349) and a slightly higher macro F1 (0.2049 vs. 0.2001). Across multiple runs (not shown in the table), these small differences fluctuated, with RoBERTa sometimes leading and DeBERTa sometimes leading, suggesting that their overall performance is effectively the same. In contrast, the QLoRA fine-tuned LLaMA model performs substantially worse, with micro F1 of 0.1665 and macro F1 of 0.0142 at a threshold of 0.30, indicating weak precision and recall for both common and rare propaganda techniques. Looking at the results, the micro and macro scores indicate that RoBERTa and DeBERTa learn the frequent classes to a similar degree, and neither encoder model clearly dominates the other.

Table 1: Final training and validation loss by model.

Model	**Training Loss**	**Validation Loss**
RoBERTa-base	**0.1224**	0.3793
DeBERTa-v3-base	0.1401	0.4003
LLaMA-3.1-8B	0.3777	**0.3048**

Table 2: Test set metrics at the best threshold for each model.

RoBERTa-base	
Best Threshold	0.70
Micro Precision	**0.3844**
Micro Recall	0.3737
Micro F1	**0.3790**
Macro Precision	**0.2948**
Macro Recall	0.1938
Macro F1	0.2001
DeBERTa-v3-base	
Best Threshold	0.80
Micro Precision	0.3698
Micro Recall	**0.3866**
Micro F1	0.3780
Macro Precision	0.2340
Macro Recall	**0.2349**
Macro F1	**0.2049**
LLaMA	
Best Threshold	0.30
Micro Precision	0.1106
Micro Recall	0.3369
Micro F1	0.1665
Macro Precision	0.0079
Macro Recall	0.0714
Macro F1	0.0142

Imbalance Issue: The dataset was quite unbalanced for the 14 propaganda types it covers, as shown in Table 3. The low macro F1 across both encoders highlights the severity of class imbalance in this dataset and confirms that rare techniques remain difficult to model. The disparity between the largest and smallest classes, respectively is over 1000 samples, and as such models may have found it easier to consistently report those classes as not present. This kind of trivial solution was definitely impacting performance. As such, we implemented a loss function taking these imbalances into account, which showed some promise. Selective sampling, and more balanced training data as a whole would improve performance further.

3.3 Analysis

The results are fairly surprising compared to the expectation that LLaMA would easily outperform the smaller BERT-based models. Multiple factors may have caused this disparity. A major one is the effect of dataset size. The SemEval 2020 Task 11 dataset is relatively small, meaning that it was harder for it to have a meaningful impact on larger models like LLaMA-3.1-8B used in this task. The model doesn't appear to benefit from their greater complexity in this task, and in fact may have been its own handicap, making further training computationally expensive and unfeasible for more epochs as well as the need for techniques such as LoRA with 4-bit quantization and a small batch size. DeBERTa and the even smaller RoBERTa were able to generalize better from the small SemEval Dataset after receiving full 5 epochs of training. However, their performance is still not that great, showing that sentence-level propaganda detection on this small, highly imbalanced dataset remains difficult regardless of model.

The imbalances in the dataset clearly come into play when considering the macro-averaged precision, recall, and F1 scores, which remained significantly lower than the micro-averaged equivalents for every model. Techniques that were greatly underrepresented such as Bandwagon and Thought-Terminating Clichés had much lower performance than the greatly overrepresented Loaded Language technique. It is interesting to note that Loaded Language as a technique is also more easily identifiable to any reader due to its overall obviousness, using words with strong connotations. Such techniques may also be more easily identified by models as well while other more nuanced techniques, which were also underrepresented in the dataset, clearly required more information to be reliably identified.

LLaMA's weaker outcomes compared to its size and expected capabilities raise an important question about architectural suitability for this type of task. Decoder-only LLMs are extremely powerful for generative tasks, but multi-label sentence classification is a more structured prediction problem that typically benefits from encoder architectures. Encoders like RoBERTa and DeBERTa build strong sentence-level representations optimized for downstream classification, while decoder models distribute information differently across layers and often require more data, more compute, or different training strategies to adapt effectively. When combined with QLoRA's constraint of only training small adapter modules while keeping the base model frozen, LLaMA likely did not have enough representational flexibility to specialize on such a small and highly imbalanced dataset.

Table 3: Propaganda technique counts in the dataset.

Propaganda Technique	Count
Appeal to Authority	174
Appeal to Fear/Prejudice	335
Bandwagon, Reductio ad Hitlerum	81
Black-and-White Fallacy	131
Causal Oversimplification	250
Doubt	613
Exaggeration, Minimisation	456
Flag-Waving	234
Loaded Language	1889
Name Calling, Labeling	941
Repetition	577
Slogans	128
Thought-Terminating Clichés	81
Whataboutism, Straw Men, Red Herring	115

Overall, these results suggest that sentence-level propaganda detection on the SemEval dataset is a challenging task for all models tested. While RoBERTa and DeBERTa perform better than LLaMA, even they show clear limitations, especially on rare labels. The outcome reinforces two themes: encoder architectures are currently better suited for this task under realistic resource constraints, and meaningful gains may require either more training data or modeling approaches that incorporate additional context beyond a single sentence.

4 Limitations and Future Work

4.1 Limitations

Dataset imbalance remains a challenge. Some propaganda techniques are relatively rare, as shown in Table 3, making it difficult for the models to learn those patterns. Although we tried to address this by using class-weighted BCE (pos_weight) and oversampling in the training loader, our evaluation still shows that rare techniques receive noticeably lower F1 scores.

Some propaganda techniques involve cross sentence dependencies, meaning a larger context window would be needed for optimal performance. Currently the models only receive singular sentences, so the lack of contextual information from surrounding sentences or the overall document may have been inhibiting performance. This was also made difficult to accomplish due to the smaller context windows allowable on the smaller BERT-based models.

One major limitation for LLaMA was that our LLaMA-3.1-8B fine-tuning was limited by computational constraints on Google Colab. We were only able to train the model for two epochs with a batch size of two and relied on 4-bit QLoRA fine-tuning. Access to more powerful computational resources for longer periods of time would have allowed for full-parameter fine-tuning and a higher bit quantization, potentially resulting in much better results.

4.2 Future Work

There is much room for progress. At the base level, substantial improvements could be made to the dataset, expanding upon it with more examples to not only provide more data for training, but to deal with the existing imbalances. More research is needed with both encoder and decoder architectures to determine if there really is an advantage for using encoders for this kind of task or if other training methods could help overcome the struggles LLaMA faced. Beyond just access to more computational resources for more epochs of training, retrieval-augmented generation (RAG) or other nuanced prompting techniques may improve LLM performance for this task. Exploration with other LLMs such as the GPT and Gemini families would be more comprehensive, due to their overall capabilities beyond that of LLaMA.

5 Team Work Clarification

This project was completed entirely as a team effort with all members contributing to the project in all of its phases. We dealt with difficulties such as computational usage limits on Google Colab, model access restrictions on Huggingface through a joint effort, with each team member pitching in at different places to get us past these obstacles. Overall, the work was evenly split.

References

Giovanni Da San Martino, Alberto Barrón-Cedeño, and Preslav Nakov. 2020. Semeval-2020 task 11: Detection of propaganda techniques in news articles. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation (SemEval-2020)*, pages 1377–1414.

Giovanni Da San Martino, Seunghak Yu, Alberto Barrón-Cedeño, Nikolay Petrov, and Preslav Nakov. 2019. Fine-grained analysis of propaganda in news articles. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5636–5646.

Shehel Yoosuf and Yang Yang. 2019. Fine-grained propaganda detection with fine-tuned bert. In *Proceedings of the 2019 Conference on EMNLP Workshop BlackboxNLP*, pages 608–612.