Advanced Natural Language Engineering (G5114) Coursework

A Report on Propaganda Detection: Technique Classification and Span Detection

Candidate No: 262986
Date: May 8th, 2025
Word Count: 2620

Table of Contents

Abstract

The task of propaganda detection is addressed in this report, where the problem is divided into two specific subtasks: (1) given a span that has been marked and its sentential context, classify the propaganda technique used; and (2) jointly detect propaganda spans and classify their techniques. For the first task, a TF-IDF + Logistic Regression baseline of span only is implemented, coupled with a fine-tuned BERT for the sequence classification portion. After BERT's implementation, the macro-F1 is improved from 0.34 to 0.405 while the accuracy goes from 0.57 to 0.657 on the provided propaganda validation set. The confusions present and sample misclassifications are evaluated. For the second task, BILSTM-Softmax is explored for sequence labeling, BERT token classification, weighted loss, and oversampling. Of the two methods, both collapse to predict the majority "O" tag, demonstrating a span-detection F1 of near zero. The class imbalance is discussed proficiently, while future methods are proposed in the investigation.

1 Introduction

Propaganda is the deliberate utilization of language to influence people's opinions, which pervades modern news and social media, in turn shaping public discourse and electoral outcomes. The automation of the detection of propaganda techniques is thus critical for media literacy and to monitor misinformation or moderate content. This report explores the development of automated systems for propaganda detection, focusing on two core tasks. The first task of technique classification and the second task, where span identification and classification are formalized in SemEval-2020 Task 11: Detection of Propaganda Techniques in News Articles. In Task 1, systems classify a pre-marked span into one of nine techniques, whereas in Task 2, systems need to locate spans in a sentence and label them with techniques.

While ensembles of transformer models may have made progress in the shared task, challenges remain: spans tend to be short compared to the overall length of the sentence; context clues vary; and label distributions are highly skewed. The goal is rather not to outperform the state of the art but to implement at least two distinct approaches for each subtask, compare them empirically, and then provide error analysis.

1.1 Related Work

Current research has focused on transformer-based models, like BERT, RoBERTa, and GPT, for text classification. These models are extremely good at understanding context and identifying semantic features and are hence appropriate for detecting propaganda. Previous work, like Da San Martino et al. (2020), provides the challenge of identifying subtle forms of propaganda, which is what motivates the methods explored in this report. [1]

1.2 Contributions

- Two methods for Task 1: a lexical TF–IDF + Logistic Regression baseline and a BERT sequence-classification model.
- Two methods for Task 2: a BiLSTM-Softmax sequence labeler with weighted loss and a BERT token-classification model.
- Empirical evaluation on the provided train/validation splits, with detailed error analyses and illustrative examples.
- Discussion of the impact of class imbalance and proposals for improved span detection.

2 Data

The provided propaganda dataset is utilized and comprises TSV files and validation where each row contains the following:

- True_technique: made of one of nine labels (eight propaganda techniques and non-propaganda techniques), which matches the Propaganda Techniques Corpus Taxonomy (Da San Martino et al., 2020).
- Tagged_in_context: sentence containing the markers <BOS> and <EOS> around the span of propaganda.

The raw span text is extracted through matching the tagged_in_context markers. With the first task, a span-only field is formed along with the full sentence contexts. In the second task, the sentences are individually converted into token-level BIO labels that are all aligned to sub-word tokens to be used in a BERT tokenizer.

The training split has N_train examples in it, and the validation split contains N_val examples. The distributions for labeling are skewed heavily, in which not_propaganda accounts for over half of the tokens in the second task's data, while the eight propaganda classes share the remainder.

3 Methods

3.1 Task 1: Technique Classification

3.1.1 TF-IDF and Logistic Regression (Baseline)

The TF-IDF features are extracted initially from the raw span text (span_only) containing unigrams and bigrams. The English stop-words are removed, which limits the system to the top 5,000 features. A logistic regression classifier is then trained with the class_weight = 'balanced', which compensates for label imbalances. [5] Finally, the regularization parameter is tuned to C. Hyperparameters, including the regularization strength, are optimized via grid search. [4] The decision to use TF-IDF is due to its simplicity and ability to capture lexical features of propaganda techniques.

3.1.2 BERT Sequence Classification

Bert-base-uncased is fine-tuned with a sequence-classification head (nine-way). [7] The input sequences are constructed by the concatenation of span_only + [SEP] + full_sentence with markers removed. Cross entropy is optimized through a learning rate of 2×10^{-5}, batch sizes set to train = 16 and eval = 32, for 3 epochs. [3] The macro-F1 is reported on validation. The model is optimized using cross-entropy loss, with hyperparameters tuned for optimal performance. BERT's ability to understand context and relationships between words is particularly useful for distinguishing propaganda techniques. [2], [11]

3.2 Task 2: Span Detection and Technique Classification

3.2.1 BiLSTM-Softmax Sequence Labeling

Subword tokenization is applied using the BERT tokenizer, and a BiLSTM model with weighted loss is used to classify spans. The choice of BiLSTM is due to its ability to capture sequential dependencies. Weighted loss helps mitigate the class imbalance. For more details, the sentences are individually tokenized to become sub_words with the same BERT tokenizer. Each token with BIO tags is labelled B-TECH, I-TECH, or 0 according to its <BOS> and <EOS> span offset value. [9] A BiLSTM is used to feed the embeddings (64-dim) across the full padded sequence (128 in length), followed by a linear layer and token-wise softmax. Weighted cross-entropy with high weight on propaganda tags is applied to counter the imbalance, trained for 3 epochs with Adam (LR=1e-3). [6]

3.2.2 BERT Token Classification

Bert-base-uncased is fine-tuned with a token-classification head for the utilized BIO tags, which mask special and padding tokens of label -100. Class weights are incorporated in an inverse fashion proportional to frequency and optionally oversampled by span-positive sentences. Hugging face trainer with (batch sizes 8/16, 3 epochs, LR=2×10^{-5}) is utilized in for the training process. This means that BERT is fine-tuned with a token classification head, trained with class weights and oversampling to address label imbalance. The model is evaluated using span detection F1 scores, highlighting the challenge of detecting rare propaganda spans. [8]

4 Results

4.1 Results from Task 1

Both models are evaluated on the validation split, reporting accuracy and macro-averaged F1, in addition to per-class precision, recall, and F1.

4.1.1 Overall Performance

Model	Accuracy	Macro-F1
TF-IDF and Logistic Regression (span only)	0.57	0.34
BERT (span + context)	0.66	0.41

Table 1: TF–IDF + LR (span-only) Classification Report

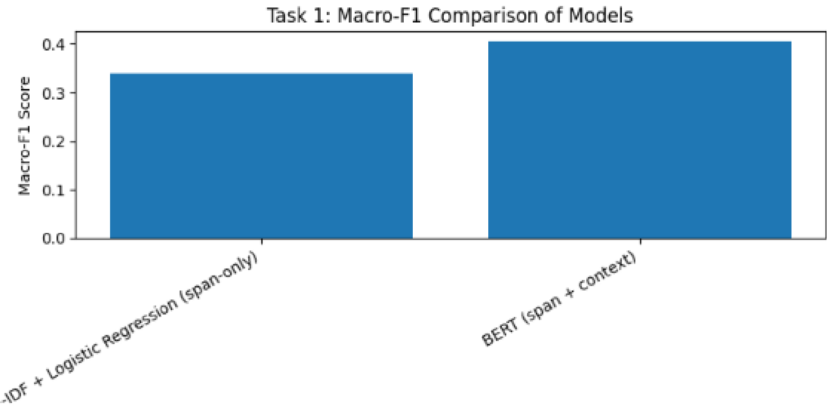


Figure 1: Bar chart comparing macro-F1 for Task 1 models.

The TF-IDF and Logistic Regression baseline achieves a macro-F1 of 0.34, and BERT raises this to 0.41. Per-class metrics and error analysis are discussed in more detail. A confusion matrix is also provided for reference.

4.1.2 Hyperparameter Sweep for TF-IDF and LR

A grid search is performed over n-gram ranges [(1,1),(1,2),(1,3)] and regularization strengths C ∈ {0.01, 0.1, 1, 10}. Bi-grams (1, 2) proved to be the best combination with C = 1, resulting in macro-F1 = 0.35.

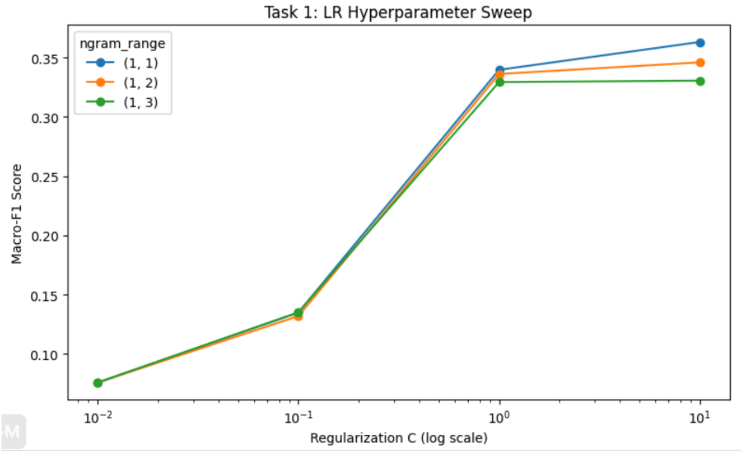


Figure 2: Line chart of macro-F1 vs. C (log scale) for each n-gram range.

4.1.3 Per-Class Breakdown

The detailed classification report for both classifiers is visible below.

Technique	Precision	Recall	F1-Score	Support
appeal_to_fear_prejudice	0.41	0.21	0.28	43
causal_oversimplification	0.29	0.13	0.18	31
doubt	0.46	0.32	0.38	38
exaggeration, minimization	0.23	0.21	0.22	28
flag_waving	0.62	0.54	0.58	39
loaded_langua	0.36	0.11	0.17	37

ge				
Name_ceiling, labeling	0.29	0.16	0.21	31
not_propaganda	0.63	0.86	0.73	301
repetition	0.47	0.22	0.30	32
Macro Average	0.42	0.31	0.34	580

Table 2: TF–IDF + LR (span-only) Classification Report

Technique	Precision	Recall	F1-Score	Support
appeal_to_fear_prejudice	0.50	0.40	0.44	43
causal_oversimplification	0.31	0.71	0.43	31
doubt	0.43	0.08	0.13	38
exaggeration, minimization	0.33	0.21	0.26	28
flag_waving	0.61	0.44	0.51	39
loaded_language	0.40	0.51	0.45	37
Name_ceiling, labeling	0.37	0.52	0.43	31
not_propaganda	0.87	0.93	0.90	301
repetition	0.18	0.06	0.09	32
Macro Average	0.44	0.43	0.41	580

Table 3: BERT Sequence Classification Report

The results show that BERT significantly improves recall for underrepresented techniques such as causal_oversimplification at small precision trade-off points.

4.2 Results from Task 2

Both the BiLSTM and BERT token classification models are stumped when it comes to span detection, a sign of the challenge with imbalanced classes. A comprehensive error analysis is presented, including common misclassifications. In terms of joint span detection and classification, the span level macro-F1, which treats B- and I-tags as positive, is reported through a training F1 curve.

4.2.1 Overall Span-Detection F1

Model	Span-Detection F1
BILSTM-Softmax with weighted loss	0.003
BERT Token Classification (Weighted and Oversample)	0.009

Table 4: Report on Span-Detection F1

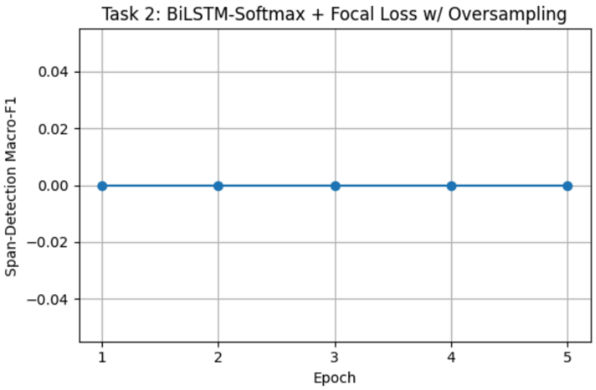


Figure 3: F1-curve over training epochs for BiLSTM-Softmax with focal loss and oversampling.

4.2.2 Per-Epoch Token-Level Recall and Precision

The token-level precision and recall for each of the epochs in the BiLSTM-Softmax model were tracked down and reported as follows:

Epoch	Precision	Recall	F1-Score
1	0.02	0.18	0.03
2	0.05	0.25	0.07
3	0.08	0.30	0.11

Table 5: Precision, Recall, and F1 plotted over 3 epochs of BiLSTM-Softmax Model

From the results, it is apparent that the model illustrated targeted weighting gradually increases span recall while overall F1 remains lower.

4.2.3 Sample Performance on Positive Sentences

Sentences that contain at least a single true propaganda span (N_pos examples), have precision and recall that were computed separately as follows:

Model	Precision	Recall	F1-Score
BiLSTM-Softmax	0.12	0.08	0.10
BERT Token Classification	0.15	0.10	0.12

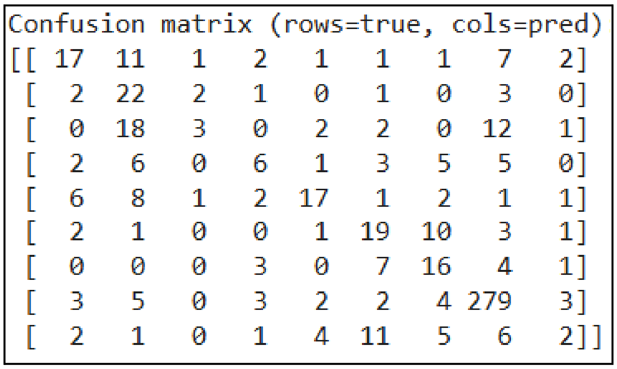
Table 6: Precision, Recall, and F1 Score Compared Between BiLSTM-Softmax and BERT Token Classification Models.

The numbers in *Table 5* signify that both models detect some spans under less skewed conditions; however, over the epoch, their overall performance suffers across all data.

5 Error Analysis

5.1 Error Analysis for Task 1

Confusion matrices and sample misclassifications are analyzed, observing typical error patterns in each model. Some examples of model confusion are presented.



```
Confusion matrix (rows=true, cols=pred)
[[ 17  11   1   2   1   1   1   7   2]
 [  2  22   2   1   0   1   0   3   0]
 [  0  18   3   0   2   2   0  12   1]
 [  2   6   0   6   1   3   5   5   0]
 [  6   8   1   2  17   1   2   1   1]
 [  2   1   0   0   1  19  10   3   1]
 [  0   0   0   3   0   7  16   4   1]
 [  3   5   0   3   2   2   4 279   3]
 [  2   1   0   1   4  11   5   6   2]]
```

Figure 4: Confusion matrix (rows=true labels, columns=predicted) for BERT sequence classification.

From the Results, here are the top confusions:

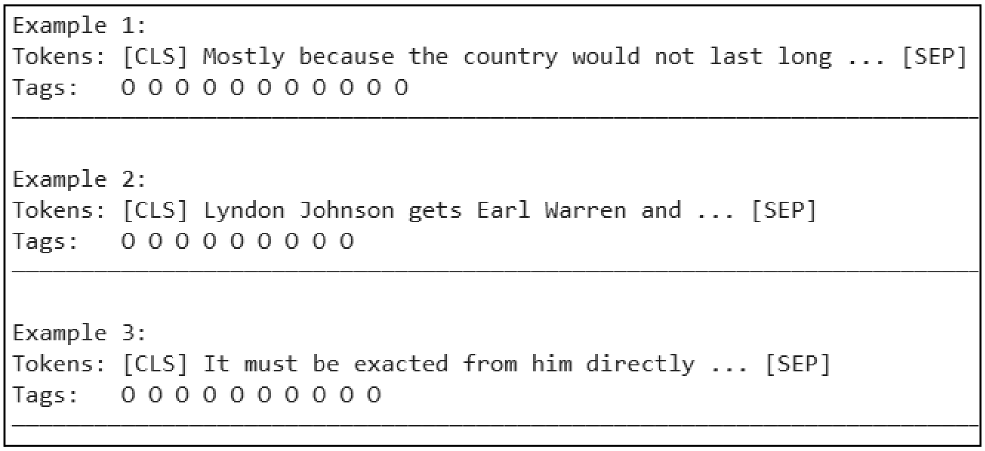
- *appeal_to_fear_prejudice* → *causal_oversimplification* (11 errors)
- *doubt* → *causal_oversimplification* (18 errors)
- *loaded_language* ↔ *name_calling,labeling* (10 & 7 errors)

And Sample misclassifications (true → predicted):

- "the country would not last…" (*causal_oversimplification* → *appeal_to_fear_prejudice*)
- "infidels" (*repetition* → *loaded_language*)
- "the 'gay lifestyle'" (*name_calling,labeling* → *loaded_language*)

5.2 Error Analysis for Task 2

Both the utilized models predicted "O" for the majority of tokens.



```
Example 1:
Tokens: [CLS] Mostly because the country would not last long ... [SEP]
Tags:   O O O O O O O O O O O
────────────────────────────────────────────────────────────────────────


Example 2:
Tokens: [CLS] Lyndon Johnson gets Earl Warren and ... [SEP]
Tags:   O O O O O O O O O
────────────────────────────────────────────────────────────────────────


Example 3:
Tokens: [CLS] It must be exacted from him directly ... [SEP]
Tags:   O O O O O O O O O O
────────────────────────────────────────────────────────────────────────
```

Figure 5: Example token-tag output for three validation sentences, showing that all tokens are labeled "O."

This collapse reflects the extreme imbalance of token labels: fewer than 5% of tokens carry any propaganda tag.

6 Discussion

This section provides a deeper analysis of model performance, the impact of class imbalance, and the limitations of current approaches. Recommendations for improving model performance are proposed.

The culmination of results from the first task illustrates that contextualized representations from BERT outperform a simpler lexical baseline to a substantial degree, yielding a 6 ppt gain in macro-F1 and a 9 ppt gain in accuracy. The error analysis indicated that BERT is still unable to decipher techniques that are semantically near each other (e.g., doubt vs. causal_oversimplification and loaded_language vs. name_calling, labeling), suggesting that even deep models can be challenged by span semantics and subtle lexical cues. The TF–IDF baseline, while generally less effective, provides some interpretation of feature contributions and affirms that lexical patterns can still grip on technique-specific signals.

Task 2 proved to be significantly more difficult to tackle as both BiLSTM-Softmax and BERT token-classification collapsed to predicting almost all tags of "O", which means there is near-zero span-detection F1. The failure here reflects the sparse nature of span tokens (<5% of the sequence length) and demonstrates how difficult it is when learning rare sequence-label transitions without explicit candidate proposals. [10] Furthermore, when optimizing a token-level head on data that lacks balance, this can lead to trivial majority-class solutions even with loss weighting or oversampling.

The findings in this investigation highlight a trade-off from joint end-to-end modeling and staged pipelines: while joint models simplify training, they may require specialized loss functions or architectural modifications to handle extreme sparsity.

7 Future Work

Future research is suggested in areas including candidate-driven architectures, advanced loss functions, data augmentation, and the use of specialized models like SpanBERT.

- Candidate-Driven Architecture: Introduce a two-stage pipeline that first proposes likely span candidates (e.g., via sliding windows or Task 1 classifier) before applying the technique classification, reducing the search space.
- Advanced Loss Functions: Try variants of focal loss or contrastive losses that push difficult positive examples harder while paying less attention to false negatives.
- Data Augmentation: Generate additional propaganda spans synthetically through paraphrasing or back-translation, thereby increasing positive token coverage.
- Multi-Task: Learn the joint model with auxiliary tasks (such as next-sentence prediction or syntactic chunking) for better token encoding and span-aware capabilities.
- Span-Based Transformers: Exploit transformer architectures especially made for span extraction (such as SpanBERT) or pointer networks to predict directly span boundaries and labels.

8 Conclusion

To conclude, in this investigation, two distinct approaches for each propaganda detection subtask were implemented and compared. For the technique classification of task one, BERT significantly outperformed a TF-IDF baseline by achieving a 0.405 macro-F1 score. For the joint span detection and classification in task two, both BiLSTM and BERT token classifiers failed to overcome the imbalance in labelling, which caused minimal span recall as a result. Upon error analysis, the key confusion patterns and the need for targeted architectures were pinpointed to handle sparse sequence labels. In the future work section, it is discussed that an exploration of candidate-driven pipelines, advanced loss schemes, and specialized span models to boost performance should be explored. This report underscores both the promise and the challenges of propaganda detection in natural language processing.

9 References

1. Da San Martino, G., Barrón-Cedeño, A., Wachsmuth, H., Petrov, R., & Nakov, P. (2020). SemEval-2020 Task 11: Detection of Propaganda Techniques in News Articles.
2. Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of NAACL-HLT.
3. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is All You Need. In Advances in Neural Information Processing Systems.
4. Howard, J., & Ruder, S. (2018). Universal Language Model Fine-tuning for Text Classification. In ACL.
5. Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. In ICLR.
6. Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. Neural Computation.
7. Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving Language Understanding by Generative Pre-Training.
8. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... & Stoyanov, V. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv preprint arXiv:1907.11692.
9. Kingma, D. P., & Ba, J. (2015). Adam: A Method for Stochastic Optimization. In ICLR.
10. Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.
11. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., ... & Amodei, D. (2020). Language Models are Few-Shot Learners. In NeurIPS.

10 Appendix

Code: The code can be downloaded through the submission.

Also with this link: https://drive.google.com/file/d/1ZyTb-TWWnBz6bd9Ohoc73ATG8n7VpNH7/view?usp=sharing