Propaganda Detection

1. Introduction

1.1. Explanation of the task and challenges involved.

Propaganda has long been used to change people's views and influence political and social events. Nowadays, as more information is shared online, it's important to be able to spot propaganda in what we read.

Using Natural Language Processing (NLP), we can automatically detect and analyse propaganda in text. NLP helps computers understand human language, aiding in identifying deceptive tactics and protecting against misinformation. This work is essential for promoting honesty and accuracy in online conversations and information sharing.

The core task of this research is twofold: first, to determine whether a given sentence contains propaganda; and second, to classify text snippets into specific categories of propaganda techniques when they are known to contain such content. Achieving accuracy in these tasks is essential not only for academic research but also for practical applications in media monitoring, information verification, and the development of educational tools to enhance critical media literacy.

However, Propaganda Detection poses several challenges. Propaganda techniques often use subtle emotional appeals and psychological tactics that are hard to categorize. The complexity of language, including idioms, sarcasm, and double meanings, adds additional complexity.

1.2. Relevance of the problem.

The aim of this project is to utilize and investigate cutting-edge NLP methods to address these complex challenges in propaganda detection. By applying advanced computational linguistics techniques, this study seeks to enhance the automatic detection and classification of propaganda, providing insights that can benefit researchers, educators, and technologists alike in their efforts to ensure the reliability and integrity of information in the public domain. (Bird, Steven, Ewan Klein, and Edward Loper, 2009).

2. Methods

2.1. Task 1 (Binary Classification of Propaganda):

BoW, BoW with Hyperparameter Tuning and BERT methods were chosen for this task.

2.1.1 Bag of Words Classifiers with Hyperparameter Tuning

The Bag of Words (BoW) model is used to transform text documents into numerical feature vectors where each unique word in the text is represented by a column in the feature matrix, and each document or sentence is a row in the matrix. Upgraded method uses GridSearchCV for optimizing model parameters to improve text classification. The process starts by converting text into numerical features using CountVectorizer, which examines both single words (unigrams) and two-word combinations (bigrams). Features are then normalized with StandardScaler to ensure each has equal weight, enhancing logistic regression performance. (Manning, Christopher D., Prabhakar Raghavan, and Hinrich Schütze 2008; Mikolov, Tomas, et al. 2013). Steps include:

- Feature - extraction Using CountVectorizer to transform text into a frequency matrix of words and word pairs.
- Normalization - applying StandardScaler to standardize feature scales.
- Optimization - employing LogisticRegression with varied regularization strengths to find the optimal balance between bias and variance.
- Grid Search - systematically testing combinations of parameters with GridSearchCV to find the best settings.

2.1.2 BERT

BERT (Bidirectional Encoder Representations from Transformers) is a transformer-based machine learning technique for natural language processing pre-training. Utilized the BertForSequenceClassification model from the Hugging Face's transformers library. This model is pre-trained on a large corpus of text and then fine-tuned for sequence classification by adding a fully-connected layer on top of the transformer. (Devlin, Jacob, et al. 2019). The steps included:

- Tokenizing text data using BERT's tokenizer.
- Processing data through BERT to obtain contextualized word embeddings.
- Fine-tuning a sequence classification model using labelled data.

2.1.3 Sensibility and justification for Method Selection

- **Bag of Words**: Chosen for its simplicity and effectiveness in binary classification tasks. This method is also computationally less demanding compared to deep learning approaches, making it suitable for initial experiments or when resources are limited. Added hyperparameter tuning through GridSearchCV because it enhances models ability to perform optimally by finding the most effective settings for both the feature extractor and the classifier.
- **BERT**: BERT's ability to understand the context of words in text due to its bidirectional training is a significant improvement over BoW methods. Selected for its ability to understand the context within text. Given the complexity of propaganda detection, which often involves nuanced language and indirect expressions, BERT's deep contextualized embeddings offer a significant advantage in capturing such subtleties.
- Both methods provide complementary insightsthe BoW models offers simplicity and speed, while BERT provides depth and contextual understanding.

2.2. Task 2 (Classification of Propaganda Techniques):

BERT and N-gram methods were chosen for this task.

2.2.1 BERT

BERT excels in understanding context and nuance in language, making it highly suitable for the subtle distinctions required in differentiating between various propaganda techniques. The implementation involves:

- Loading the pre-labelled training and validation data.
- Mapping textual labels to unique IDs for classification.
- Tokenizing the text using BERT’s tokenizer, which prepares the data for the model.
- Creating PyTorch datasets and dataloaders to handle the data during training.
- Setting up the BERT model for sequence classification with the number of unique labels.
- Training the model using Hugging Face’s Trainer with custom training arguments to optimize learning.
- Evaluating model performance using metrics like accuracy, precision, recall, and F1 score.

2.2.2 N-gram Model

N-gram models are a simple yet effective method for capturing the local context within text. By using combinations of words (bigrams, trigrams), this method can detect common phrases and expressions typical to specific propaganda techniques. (Jurafsky, Daniel, and James H. Martin, 2021). Steps included:

- Preprocessing the text to normalize and tokenize it.
- Utilizing CountVectorizer to transform the text into n-gram frequency features.
- Training a Naive Bayes classifier on these features, suitable for handling the frequency-based input.
- Evaluating the model with standard classification metrics.

2.2.3 Sensibility and justification for Method Selection

- **BERT:** Highly sensible for the task due to its state-of-the-art performance in various NLP applications, especially in understanding context, which is crucial for correctly identifying specific propaganda techniques.

Chosen for its advanced capabilities in handling context and sequence learning, essential for the subtle differentiation required in this task.

- **N-gram**: Offers a straightforward approach to capture local text patterns, sensible for identifying repeated phrases typical in propaganda. It is less computationally intensive compared to deep learning methods, providing a good balance between performance and computational efficiency. Selected for its effectiveness in feature extraction from text data, useful in identifying specific linguistic patterns associated with different propaganda techniques.

These methods cover a spectrum from traditional to advanced deep learning, providing a comprehensive approach to the task. Each has its strengths in dealing with different aspects of the text data.

3. Hyper-Parameter Settings

3.1. Task 1:

3.1.1 Bag of Words Classifiers with Hyperparameter Tuning

Fixed Parameters:

- CountVectorizer: Transforms text to numerical data by counting word occurrences. Configured to consider stopwords and perform tokenization to enhance model interpretability and performance.
- StandardScaler: Ensures all features contribute equally to the logistic regression model by normalizing feature scales, important because logistic regression is sensitive to the scale of input features.

Optimized Parameters:

- N-gram Range: Examines both unigrams (single words) and bigrams (pairs of words) to capture more context within the data, which could be crucial for identifying propagandistic content.
- Regularization Strength (C): Varies from 0.01 to 100 to fine-tune the logistic regression model’s complexity, balancing between underfitting and overfitting.

Rationale:

- CountVectorizer with N-grams: Helps in capturing local context, increasing the chances of detecting propaganda based on commonly used phrases or word pairings.
- Regularization (C): Adjusts the model's tendency to overfit, ensuring it generalizes well to new data. A broader range of C values allows the model to explore various degrees of regularization, optimizing performance on the validation set.

3.1.2 BERT

Fixed Parameters:

- Model Architecture (BertForSequenceClassification): Pre-trained BERT model adapted for sequence classification tasks.
- Tokenizer (BertTokenizer): Specific to BERT, ensuring consistency with how the model was originally trained.

Optimized Parameters:

- Number of Training Epochs: Set to 3 based on preliminary tests to balance training time and model accuracy.
- Batch Size: Training and evaluation batch sizes were tuned to optimize GPU usage.
- Learning Rate and Warmup Steps: These were adjusted to ensure the model converges to a good solution without overshooting or getting stuck.

Rationale:

- Epochs and Batch Size: These settings ensure efficient use of computational resources while allowing the model enough iterations to learn from the data.
- Learning Rate and Warmup: Critical for stabilizing the model's training dynamics in the initial phases.

3.2.Task 2:

3.2.1 BERT

Hyperparameters:

- Number of epochs: This controls how many times the model will see the entire dataset. Fixed at 3 to balance between training time and learning enough patterns.
- Batch size: Size of the data subsets the model sees before updating the weights. Fixed at 8 for training to manage GPU memory usage effectively.
- Warmup steps: Number of steps to perform learning rate warming up. Fixed at 500 to avoid very rapid updates at the beginning of training.
- Weight decay: Regularization parameter to prevent overfitting. Set to 0.01 to add a small amount of additional penalty on large weights.
- Learning rate strategy: Utilized a scheduler that adjusts the learning rate based on the training progress.

Parameters Explored:

- Learning rate: explored through different runs to find an optimal setting that balances convergence speed and model stability.

Rationale:

- These settings were chosen to provide a balance between efficiency and performance, considering the complexity of BERT and the computational resources typically available.

3.2.2 N-gram Model with Naive Bayes Classifier

Hyperparameters:

- Ngram_range: Determines the range of n-gram sizes to use. Explored both unigrams (1,1) and bigrams (1,2) to see if including the sequence of words improves the model.
- Alpha in Naive Bayes: Smoothing parameter, not specifically mentioned but often explored to handle the problem of zero probability in unseen data.

Fixed Parameters:

- Vectorizer settings such as stop words filtering were set to remove common English words that might not contribute much information to model training.

Rationale:

- Using both unigrams and bigrams allows the model to capture not only the presence of specific words but also the local word order, providing more context to the model.

4. Evaluation

4.1.Task 1:

4.1.1 Bag of Words Classifiers with Hyperparameter Tuning

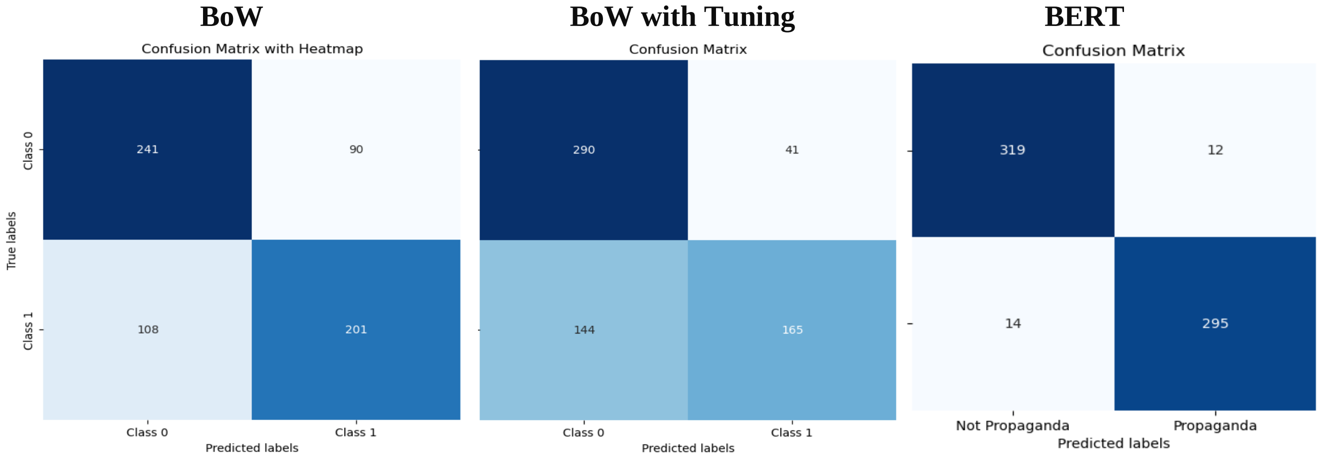
This approach optimized a logistic regression classifier that is trained on text data converted into numerical features using CountVectorizer. The hyperparameter tuning aimed to enhance the classifier's performance by finding the ideal balance between underfitting and overfitting, considering both single words and bigrams to better capture the context within the text but the result was not significantly different.

4.1.2 BERT

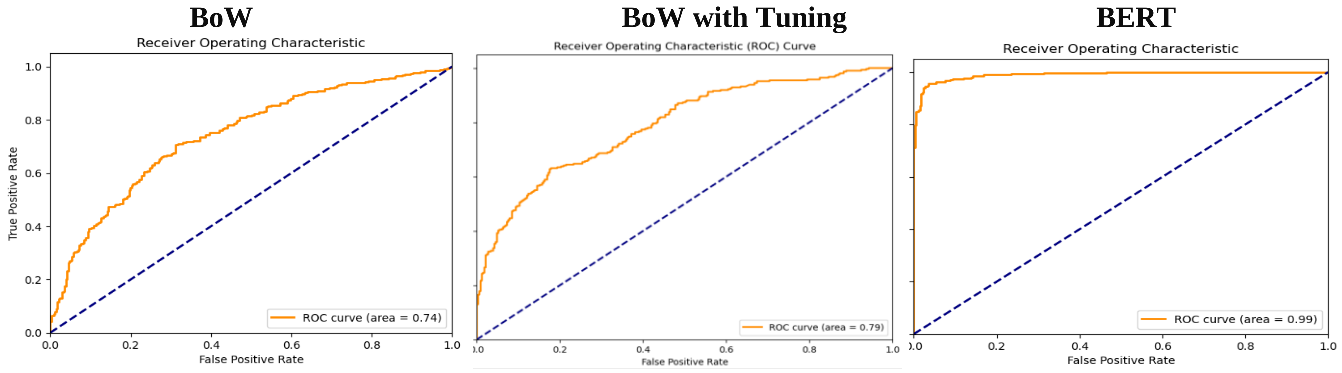
The BERT model benefits from a deep understanding of language context and nuances due to its architecture and pre-training on a large corpus. Fine-tuning this model on the specific task of propaganda detection allows it to leverage its pre-trained knowledge for better performance in identifying subtle linguistic features associated with propaganda.

4.1.3 Graphical Representation

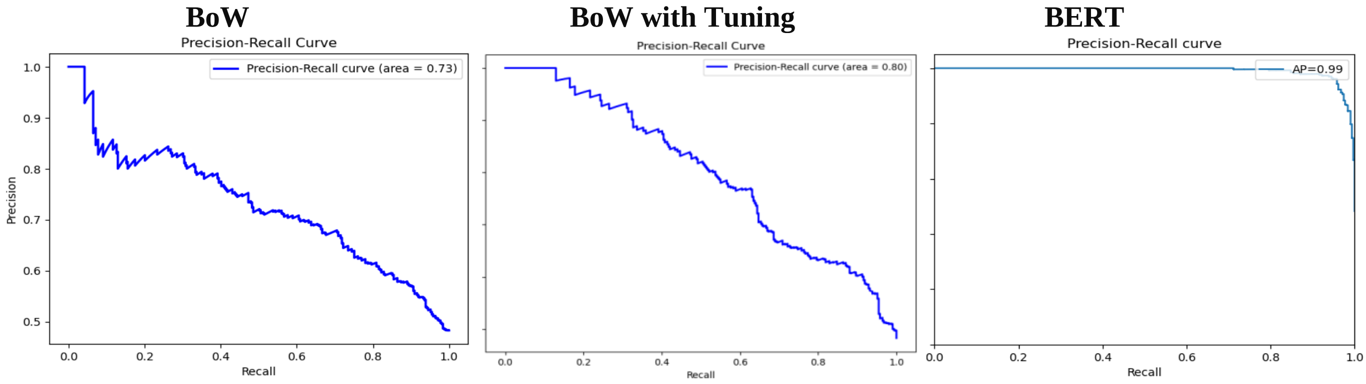
1. **Confusion Matrix Comparison**: Displays the counts of true positives, true negatives, false positives, and false negatives, providing a quick overview of model performance. BERT outperforms both BoW models.



2. **ROC Curve Comparison**: Shows the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. The area under the curve (AUC) gives a single scalar value to evaluate the overall performance of the model, where a higher value indicates better discrimination capabilities. BERT shows better performance than both BoW models.

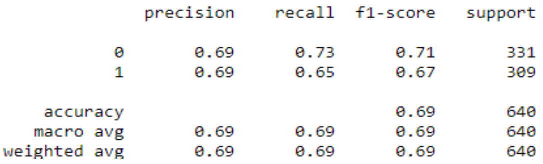


3. **Precision-Recall Curve Comparison**: Useful for showing the trade-off between precision and recall for different threshold settings. This curve is especially beneficial when the classes are imbalanced. The area under the curve (PR AUC) provides a measure of the model's ability to distinguish between classes at various threshold levels. BERT demonstrates a high recall, missing very few actual positives.



4. Evaluation Metrics Overview

BoW

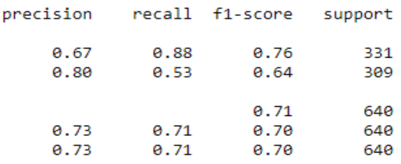


```
              precision    recall  f1-score   support

           0       0.69      0.73      0.71       331
           1       0.69      0.65      0.67       309

    accuracy                           0.69       640
   macro avg       0.69      0.69      0.69       640
weighted avg       0.69      0.69      0.69       640
```

BoW with Tuning

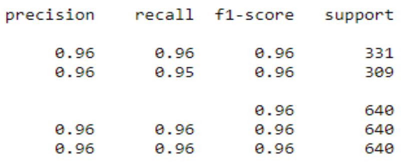


```
precision    recall  f1-score   support

    0.67      0.88      0.76       331
    0.80      0.53      0.64       309

                        0.71       640
    0.73      0.71      0.70       640
    0.73      0.71      0.70       640
```

BERT



```
precision    recall  f1-score   support

    0.96      0.96      0.96       331
    0.96      0.95      0.96       309

                        0.96       640
    0.96      0.96      0.96       640
    0.96      0.96      0.96       640
```

BERT significantly outperforms the other models across all metrics. It not only detects true positives and negatives more accurately but also minimizes errors significantly better than both versions of BoW. This shows the effectiveness of BERT in handling complex classification tasks. The improvements in BoW with Tuning over standard BoW also illustrate how parameter optimization can help enhance model performance, although not to the extent of more sophisticated models like BERT.

4.2.Task 2

4.2.1 BERT

The BERT model showed a high performance in terms of accuracy, precision, recall, and F1-score. It has been especially effective due to its deep contextual understanding gained from pre-training.

4.2.2 N-gram

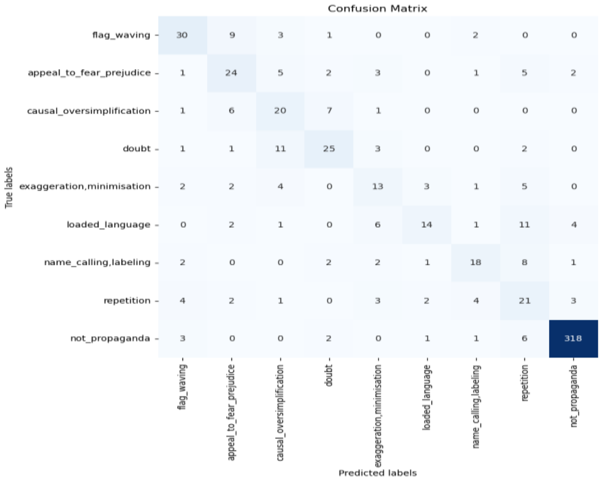
The N-gram model, using a Bag of Words approach with CountVectorizer, demonstrated moderate effectiveness. Its simplicity and lack of contextual awareness are reflected in the lower performance metrics compared to the BERT model.

4.2.3 Graphical Representation

Confusion Matrix Comparison :

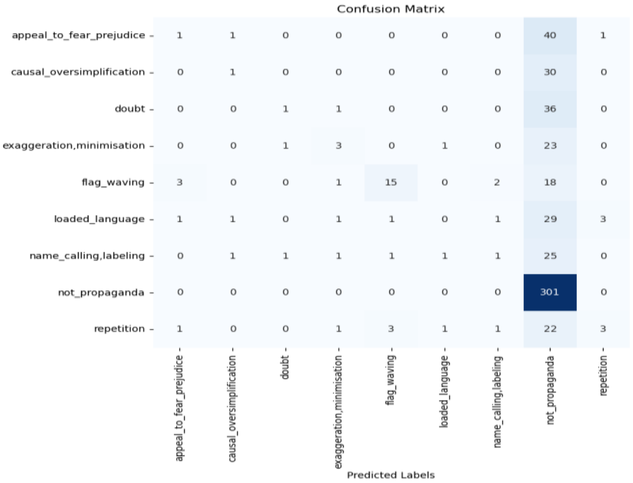
BERT predicted both classes reasonably well indicating that this model is better at detecting the presence of propaganda than N-gram model.

BERT



	flag_waving	appeal_to_fear_prejudice	causal_oversimplification	doubt	exaggeration,minimisation	loaded_language	name_calling,labeling	repetition	not_propaganda
flag_waving	30	9	3	1	0	0	2	0	0
appeal_to_fear_prejudice	1	24	5	2	3	0	1	5	2
causal_oversimplification	1	6	20	7	1	0	0	0	0
doubt	1	1	11	25	3	0	0	2	0
exaggeration,minimisation	2	2	4	0	13	3	1	5	0
loaded_language	0	2	1	0	6	14	1	11	4
name_calling,labeling	2	0	0	2	2	1	18	8	1
repetition	4	2	1	0	3	2	4	21	3
not_propaganda	3	0	0	2	0	1	1	6	318

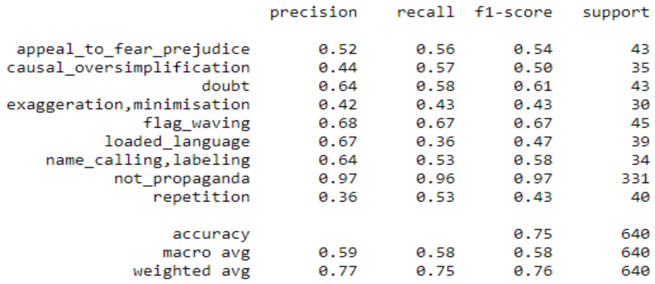
N-gram



	appeal_to_fear_prejudice	causal_oversimplification	doubt	exaggeration,minimisation	flag_waving	loaded_language	name_calling,labeling	not_propaganda	repetition
appeal_to_fear_prejudice	1	1	0	0	0	0	0	40	1
causal_oversimplification	0	1	0	0	0	0	0	30	0
doubt	0	0	1	1	0	0	0	36	0
exaggeration,minimisation	0	0	1	3	0	1	0	23	0
flag_waving	3	0	0	1	15	0	2	18	0
loaded_language	1	1	0	1	1	0	1	29	3
name_calling,labeling	0	1	1	1	1	1	1	25	0
not_propaganda	0	0	0	0	0	0	0	301	0
repetition	1	0	0	1	3	1	1	22	3

Evaluation Metrics:

BERT

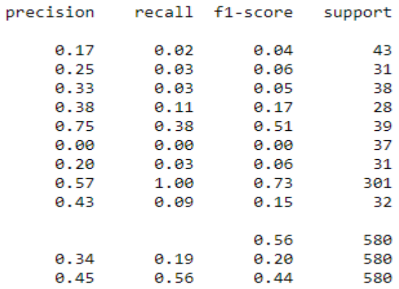


```
                           precision    recall  f1-score   support

    appeal_to_fear_prejudice       0.52      0.56      0.54        43
  causal_oversimplification       0.44      0.57      0.50        35
                      doubt       0.64      0.58      0.61        43
  exaggeration,minimisation       0.42      0.43      0.43        30
                flag_waving       0.68      0.67      0.67        45
            loaded_language       0.67      0.36      0.47        39
      name_calling,labeling       0.64      0.53      0.58        34
             not_propaganda       0.97      0.96      0.97       331
                 repetition       0.36      0.53      0.43        40

                   accuracy                           0.75       640
                  macro avg       0.59      0.58      0.58       640
               weighted avg       0.77      0.75      0.76       640
```

N-gram



```
precision    recall  f1-score   support

     0.17      0.02      0.04        43
     0.25      0.03      0.06        31
     0.33      0.03      0.05        38
     0.38      0.11      0.17        28
     0.75      0.38      0.51        39
     0.00      0.00      0.00        37
     0.20      0.03      0.06        31
     0.57      1.00      0.73       301
     0.43      0.09      0.15        32

                         0.56       580
     0.34      0.19      0.20       580
     0.45      0.56      0.44       580
```

BERT: Precision and Recall are balanced across classes, with overall high F1-scores, indicating a good balance between Precision and Recall. N-gram Model: Precision is generally lower, especially for propaganda, indicating many false positives. Recall is notably poor for propaganda, suggesting the model fails to detect most propaganda instances.

BERT outperforms the N-gram model in almost all metrics, showing better handling of context and semantics in text, which are crucial for nuanced tasks like detecting propaganda.

5. Analysis

5.1.Task 1

5.1.1 Bag of Words Classifiers with Hyperparameter Tuning

Error Patterns:

- Overfitting on Common Patterns: Despite the use of regularization, the model can overfit on frequent patterns in the training data, failing to generalize well on unseen, more complex texts or texts with different distributions.
- Misinterpretation of Context: Similar to the Bag of Words model, this logistic regression approach can struggle with contexts where the order and interaction of words are crucial for correct interpretation, as it still fundamentally relies on the frequency of word appearances rather than their sequential relationship.

Underperforming Types of Input:

- Phrases that don't translate directly based on the constituent words can confuse the model, leading to incorrect classifications.
- Texts where multiple interpretations are valid or the context outside a single sentence is needed for correct classification are particularly challenging.

5.1.2 BERT

Error Patterns:

- If not fine-tuned on a sufficiently large and representative dataset, BERT can overfit to the training data, failing to generalize well on unseen examples.
- When faced with data significantly different from the data seen during training, such as a different domain or a different style of writing, BERT models may not perform well.
- While better at context than BoW, BERT can still struggle with very subtle uses of language or highly indirect implications.

Underperforming Types of Input:

- Highly specialized texts that use jargon or are stylistically different from the training data can lead to lower performance.
- Texts with complex rhetorical devices or subtle propaganda techniques that require deep semantic understanding beyond local context cues.

5.1.3 Differences in Error Patterns

- BERT has significant advantage over BoW models due to ability to understand context and the relationships between words in a sentence. This leads to generally better performance on texts where context is key to interpretation.
- Generalization: BERT typically generalise better across different types of texts due to their pretraining on a vast corpus of data. In contrast, BoW models may only perform well on data similar to their training set.
- Error Sensitivity: Errors in BoW models are often due to vocabulary limitations and lack of contextual awareness, whereas errors in BERT can arise from inadequate fine-tuning or the subtleties in text that require broader world knowledge or inference capabilities.

5.2.Task 2

5.2.1 BERT

Error Patterns:

- There were few instances of incorrectly labelling non-propaganda as propaganda and vice versa.
- Complex Language Use: BERT may struggle with highly nuanced or complex sentence structures that deviate significantly from its training corpus.
- Rare Propaganda Techniques: If certain propaganda techniques are underrepresented in the training data, BERT may fail to recognize these effectively.
- Overfitting: In cases where training data is not diverse enough, BERT might overfit to the training examples and perform poorly on validation data that differs slightly in style or content.

Underperforming Types of Input:

- Texts with subtle linguistic cues that are crucial for understanding the intent, such as irony or sarcasm.
- Texts using less common propaganda techniques or those that are mentioned infrequently in the training set.

5.2.2 N-gram

Error Patterns:

- N-gram Model shows imbalance in class distribution and poor performance in accurately classifying propaganda.
- F1-scores are consistently lower highlighting inefficacy in handling this dataset.
- Context Ignorance: As an N-gram model primarily captures frequency and proximity of word sequences, it misses the broader semantic context, which is often essential in understanding propaganda.
- Data Sparsity: Rare N-grams or those that appear in specific contexts can lead to overfitting, where the model performs well on seen data but poorly on unseen data.

Underperforming Types of Input:

- Sentences where the semantic meaning is not apparent through adjacent word analysis alone.
- Creative or unusual uses of language, where standard n-grams might not be predictive.

5.2.3 Differences in Error Patterns

- BERT outperforms N-gram model in terms of contextual understanding due to its attention mechanisms and deeper neural network architecture.
- BERT learns contextual features directly from text, while N-gram captures local word patterns. The feature learning in BERT is more adaptable and nuanced compared to the other models.
- BERT generalizes better than N-gram model due to its pre-training on a vast corpus.

6. Conclusions

6.1 Task 1

Summary of Findings and Their Implications

1. **Hyperparameter-Tuned BoW Classifiers:**

Effective for basic binary classification, focusing on word frequency without considering context. Suitable for preliminary analysis but limited in handling complex propaganda.

2. **BERT:**

Demonstrated great ability to understand deep linguistic contexts, making it highly effective for detailed propaganda detection.

Effectiveness and Implications

- BoW classifiers are quick and efficient for straightforward tasks but lack depth for nuanced analysis.
- Hyperparameter tuning makes BoW more adaptable and slightly more effective.
- BERT excels in detailed and context-sensitive tasks, ideal for complex analysis like content moderation.

6.2 Task 2

Summary of Findings and Their Implications

1. BERT:

- Achieved high metrics, showcasing strong performance in nuanced language tasks.
- Demonstrates that advanced NLP models like BERT are essential for complex text analysis tasks, indicating their utility in broader linguistic applications.

2. N-gram Model:

- Moderate effectiveness, capable in straightforward scenarios but limited in complex contexts.
- Suitable for preliminary text analysis and situations with limited resources, though it requires supplementation by more advanced models for depth.

Effectiveness and Implications

- BERT's deep learning capabilities allow for superior understanding of context, making it highly effective for detailed text analysis.
- N-gram’s simplicity serves well in resource-constrained environments but lacks the depth needed for more complex analysis, limiting its use to more general tasks or as a preliminary tool.

7. Future Work

Suggestions for Improvements and Further Research:

Hybrid Models: 1. Exploring combining BERT with N-gram or Word2Vec to balance contextual understanding with computational efficiency. 2. Utilizing ensemble methods to enhance accuracy and robustness.
Advanced Pre-training Techniques: 1. Expanding BERT's training with diverse datasets to better recognize complex propaganda. 2. Focusing on domain-specific pre-training for enhanced sensitivity to specific propaganda styles.
Exploration of New Model Architectures: 1. Investigating scalable and efficient alternatives like Sparse Transformers. 2. Developing compact models like DistilBERT for reduced resource use.
Improving Data Quality and Quantity: Enriching training datasets with rare examples and applying semi-supervised learning to leverage unlabelled data.
Interdisciplinary Approaches: 1. Integrating psychological and social insights to refine model understanding of propaganda. 2. Adapting models for diverse cultural contexts to improve global applicability.
Potential for Other Methods or Applications: Developing multilingual models for global propaganda detection and real-time systems for immediate disinformation flagging on social platforms.

8. References

- **Bird, Steven, Ewan Klein, and Edward Loper.** "Natural Language Processing with Python." O'Reilly Media, Inc., 2009.
- **Jurafsky, Daniel, and James H. Martin.** "Speech and Language Processing." 3rd ed., draft, 2021.
- **Mikolov, Tomas, et al.** "Distributed representations of words and phrases and their compositionality." 2013.
- **Devlin, Jacob, et al.** "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics, 2019.
- **Manning, Christopher D., Prabhakar Raghavan, and Hinrich Schütze.** "Introduction to Information Retrieval." Cambridge University Press, 2008.

9. Code Appendix

NLP Coursework code.ipynb