CS4248 Final Report - Lableled Unreliable News

A0194551B, A0205182N, A0201185R, A0194494N, A0218236H
Group 15
Mentored by HAI Ye
{jiahe.li, e0425105, yuanxing_z, tang_yuxuan, kelvin.soo}@u.nus.edu

Abstract

This paper presents an investigation of document classification and misinformation detection to predict the reliability of news documents. Natural Language Processing (NLP) techniques were utilized, including traditional machine-learning methods and advanced deep learning models to create a robust news classifier. The Labeled Unreliable News (LUN) dataset was used to evaluate the proposed approaches. This study aims to improve the accuracy of news documents and mitigate the spread of false information. The findings contribute to the ongoing effort to combat misinformation by providing a reliable and practical framework for reliability prediction in the area of NLP.

1 Introduction

1.1 Motivation

The rise of social media and online news sources has led to a proliferation of misinformation. The problem has become particularly acute in recent years with the increasing influence of social media platforms in shaping public opinion. The dissemination of unreliable news, such as the spread of conspiracy theories and the manipulation of public opinion, has been linked to a range of negative consequences, such as political instability, erosion of public confidence in the media, social divide, and public health risk. For example, during the COVID-19 pandemic, the spread of false information about the virus has been linked to public health risks. Misinformation has led to confusion and distrust in public health guidance, for instance, the outlandish claims of unverified medications that have miraculous healing properties. Therefore, classifying the reliability of news documents can help prevent the spread of false information and promote accuracy in reporting.

1.2 Problem Statement

This project aims to address the issue of unreliable news in digital media by developing a robust news classifier using machine-learning and deep-learning techniques in the field of Natural Language Processing (NLP). The proposed approaches include traditional machine learning methods such as logistic regression, support vector machines, deep learning methods like long-short-term memory and transformer models, as well as more advanced prompt-tuning. Our study evaluates the proposed approaches on an open-source dataset called Labeled Unreliable News (LUN) comprising 48K news articles for training and 3K articles for testing. The news articles will be classified as satire, hoax, propaganda, or reliable news based on the context and semantic understanding of the trained models. The study aims to improve the reliability of news consumption and alleviate the dissemination of false information while providing team members with practical exposure to the field of NLP.

1.3 Key Contributions

Our project introduced cutting-edge tools such as transformer models and prompt-based learning on top of established techniques as mentioned. Experiments were designed to compare and contrast the performances given by different models and different feature combinations. In the end, we hope to enhance the reliability of news assumptions and halt the propagation of incorrect information.

2 Literature Review

2.1 Traditional Machine Learning Methods for Fake News Classification

Several traditional machine learning techniques have been employed in fake news classification research, including logistic regression, decision trees, random forests, and support vector machines

(SVM) (Vogel and Meghana, 2020). In traditional machine learning methods, feature extraction plays a crucial role. Faustini and Covoes (2020) extracted custom features, Word2Vec, DCDistance, and bag-of-words, achieving an accuracy of 79%, indicating the limitations of hand-crafted features in traditional machine learning methods. Hakak et al. (2021) summarised the performance of various models, showing that deep learning models generally outperform traditional machine learning methods, making them a suitable baseline for comparison.

2.2 Deep Learning Approaches

2.2.1 Recurrent Neural Networks (RNNs) and LSTM

Recurrent neural networks (RNN), especially long-short-term memory (LSTM) networks, have shown better performance in fake news classification compared to traditional machine learning methods (Hakak et al., 2021). However, they are no longer the state-of-the-art, and their lack of explainability remains a concern (Guo et al., 2019).

2.2.2 Transformers and Pre-trained Language Models

Transformers have revolutionized the NLP field, leading to a surge of pre-trained language models like BERT, RoBERTa, and GPT (Qasim et al., 2022). These models have achieved state-of-the-art performance in fake news classification tasks. For example, Jwa et al. (2019) used BERT for the first time in fake news detection with a headline-body dataset, achieving better F1 scores than other state-of-the-art methods. Kula et al. (2021a) proposed a hybrid architecture that mixes BERT with RNNs, achieving competitive results. They also utilized a combination of CNNs with BERT, achieving an accuracy of 98.90% on test data (Kula et al., 2021b).

2.3 Prompt-based Learning for Fake News Classification

Prompt-based learning with large-scale pre-trained models, such as BERT, RoBERTa, and GPT-3, has gained attention for its ability to perform well with minimal fine-tuning (El Vaigh et al., 2021). These models have shown potential for further performance improvements in fake news detection tasks (Jiang et al., 2022). In particular, the knowledgeable prompt learning (KPL) model achieves an average increase of 3.28% in the F1 score under low resource conditions compared to fine-tuning.

3 Corpus Analysis & Feature Engineering

3.1 Data Exploration

The texts in the LUN dataset were categorised into 4 labels–1 represents satire, 2 for hoax, 3 for propaganda, and 4 represents reliable news. The training dataset contained 48854 rows and 2 columns–the text and label columns. The test data set consisted of 3000 instances with text and label columns. There were a total of 202 duplicated train instances and they were removed, hence 48652 train instances were used. There were no missing values in the train and test data sets. From Figure 1, label 3 has the most number of instances at 17870, followed by label 1 (13911 instances), then label 4 (9932), and lastly label 2(6939). For the test set, the number of test instances in each label was 750.

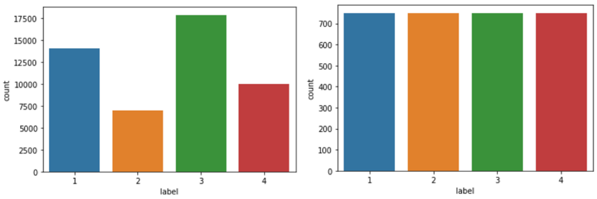


Figure 1: Countplot of train (left) and test (right) instances for each label

3.2 Feature Engineering on Raw Text

Engineered features include the log transformations applied to the characters count, sentences count, numerical values count, and stopwords count. In the calculation, we added one to the count to avoid numerical overflow due to those zero values. As shown in Figure 2, looking at the interquartile range, among the four classes, label 3 has the largest variations across all counts, with its median count in sentence, character, and stopwords being the highest. Label 2 has a smaller variation between samples and the smallest median count in numerical values. In addition, label 1 and label 2 have a similar median for character count and stopword count, while label 3 and label 4 are similar in numerical counts. Lastly, we have observed outliers that could potentially reduce the significance of using these variables for classification purposes. However, these outliers will not be removed as they can potentially occur in real-life situations. In all, these analyses helped us to better understand the data and enabled us to make effective use of the features for modeling later.

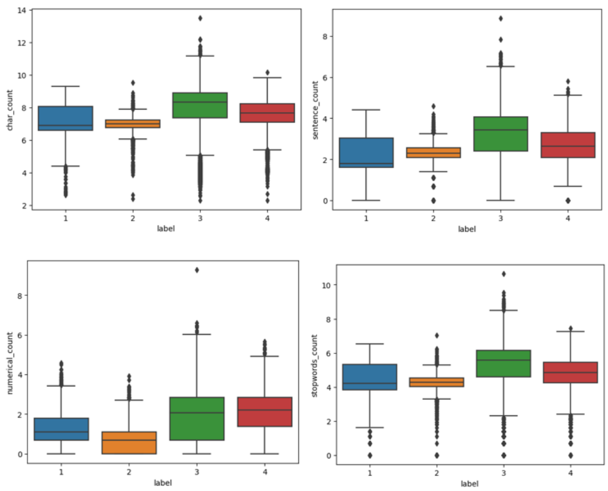


Figure 2: Boxplots of log transformations of character count, sentence count, numerical count and stopword count, from left to right, top to bottom

3.3 Data Cleaning and Further Feature Engineering

Preprocessing was done on the raw text data. The initial step in this process involved tokenization, which involved partitioning the input text into individual tokens using whitespace as a delimiter while converting the text to lowercase. Subsequently, tokens of insufficient or excessive length were filtered, with those comprising fewer than three characters being discarded and those exceeding 15 characters being truncated. Another essential aspect of our preprocessing involved the removal of stop words from the tokenized text. These common words, such as "and" or "the," often contribute little to the overall meaning and can be safely omitted from the analysis. In the final stage, alphanumeric filtering was employed to eliminate tokens containing non-alphanumeric characters, including punctuation marks and numbers. This step ensured that the text was free of extraneous elements, thereby streamlining the data for more effective NLP applications.

Moreover, following the preprocessing stage, we employed the Term Frequency-Inverse Document Frequency (TF-IDF) approach to capture the word importance factors. In this process, tokens that appeared in more than 80% of the documents or fewer than 3 documents were excluded. Additionally, we imposed a constraint on the maximum number of tokens, limiting it to 1,000. This measure helped to refine the feature set, ensuring that only the most relevant and informative tokens were utilized.

Following this, pre-trained word embedding GloVe was also implemented. GloVe, or Global Vectors for Word Representation, is an unsupervised learning algorithm that employs co-occurrence statistics from large text corpora to generate continuous, dense vector representations for words, which effectively capture syntactic and semantic relationships among them. In the different models we used below, we used different methods to handle this feature set, either through aggregation or by retaining the original word vectors to keep the sequential information. We also defined several parameters for efficiency purposes: the vector dimensionality was set to 200, the vocabulary was limited to a maximum of 10,000 features, and the cleaned texts were restricted to a maximum length of 500 words to prevent excessively lengthy input.

Different combinations of features, including our hand-crafted features listed above, TF-IDF, and word embeddings were examined in our ablation studies, which are described in the next section.

4 Methodology

4.1 Evaluation

In the modeling step, we used several evaluation matrices to assess and compare the model performances. Firstly, accuracy is a straightforward assessment of a classifier's performance to analyze the proportion of correct classifications. Secondly, the F1 score is used as it combines the trade-off between precision and recall, which is particularly suitable for multiclass classification problems. We use the macro F1 score, as the training dataset exhibits a reasonable degree of balance between the 4 classes. Lastly, we presented the confusion matrix for a detailed analysis of the model's performance in each class, by displaying the counts of true positive, false positive, true negative, and false negative predictions.

4.2 Logistic Regression

Logistic Regression was explored together with the support vector machine as our baseline model. In particular, Logistic Regression has low computational complexity, which makes it a practical option for large-scale text classification tasks. In this task as well as the Support Vector Machine in the next subsection, the word embedding vectors were taken by averaging across an input document. For Logistic Regression, we used L2 regularisation to shrink the coefficients of less important features, making the model more robust and less prone to overfitting. An ablation study shown in Table 1 was performed. With only the word embedding vector, the test accuracy and macro F1 score are 0.6513 and 0.6382 respectively. Adding the hand-crafted features as well as the TF-IDF features improved the performance by about 10%, which shows their effectiveness.

Models	Test accuracy	Test macro f1 score
Using all features	0.7283	0.7227
Using only word embedding features	0.6513	0.6382

Table 1: Logistic regression ablation study

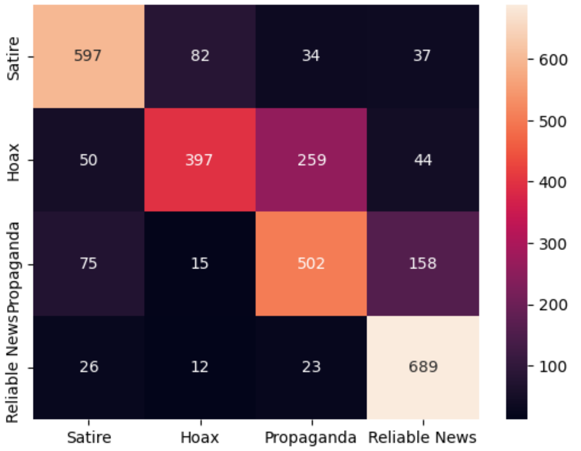


Figure 3: Confusion matrix of most optimal logistic regression model

4.3 Support Vector Machine

The main idea behind Support Vector Machine (SVM) is to find the optimal decision boundary that can separate the data points of different classes, in contrast with Logistic Regression whose objective is to minimize cross-entropy loss. Furthermore, we used the RBF kernel to learn nonlinear decision boundaries in a high-dimensional feature space apart from the linear boundary. Using all the features we have, variants of SVMs were built. To speed up the SVM training loop, we have reduced the dimensionality of the data using Singular Value Decomposition (SVD). The original input with 1204 features was projected onto a reduced feature space with 200 components.

Models	Test accuracy	Test macro f1 score
SVM with linear kernel	0.7210	0.7174
SVM with radial basis (RBF) kernel	0.6843	0.6751

Table 2: SVM ablation study with all features utilised

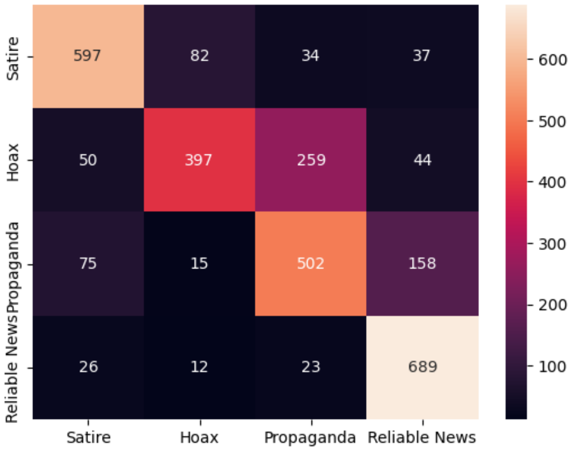


Figure 4: Confusion matrix of most optimal SVM model

Concerning Table 2, SVM with linear kernel performed better than SVM with RBF kernel on the test set, achieving a macro f1 score of 0.7158. This could be because our data are already linearly separable, given that the linear kernel version also performs better on the training set. In our discussion section later, we looked into our corpus to better explain this.

4.4 LSTM and Its Variants

LSTM was preferred over traditional RNNs due to its advanced architecture that includes memory cells and gating mechanisms, which enhance the ability to capture long-term dependencies in input data. In particular, LSTMs can selectively remember or discard information from past inputs, providing an advantage when processing noisy or irrelevant data. Furthermore, LSTMs are adept at mitigating the vanishing and exploding gradient issues commonly encountered in conventional RNNs. Here, to retain the sequential information, the word embeddings were kept at a word level instead of aggregated to a document level. Before putting in the model, we padded the vector inputs to uniform them. This is first fed into an embedding layer and one or two LSTM layers, after which the output from this is concatenated with the TFIDF features and handcrafted features. In the end, the resulting output was fed into the dense layers. The LSTM architecture used is shown in Figure 5.

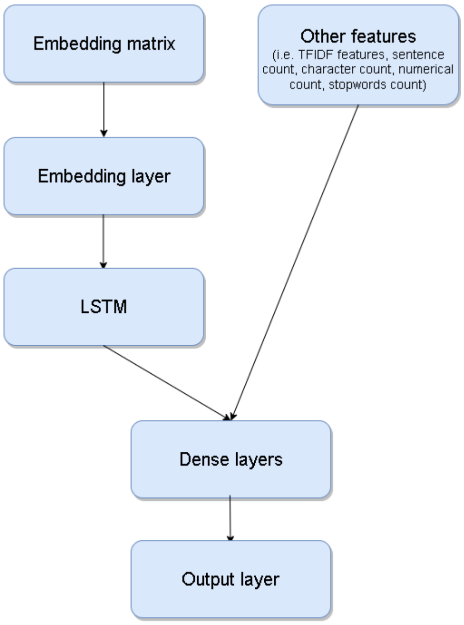

Embedding matrix
Other features
(i.e. TFIDF features, sentence count, character count, numerical count, stopwords count)
Embedding layer
LSTM
Dense layers
Output layer

Figure 5: LSTM architecture

Features	Layers	Test accuracy	Test macro f1 score
Using all features except TF-IDF	1 LSTM layer	0.6440	0.6246
	2 LSTM layers	0.6767	0.6643
Using only word embedding features	2 LSTM layers	0.6793	0.6626
Using all features	2 LSTM layers	0.6930	0.6838

Table 3: LSTM ablation study

As above, we also conducted ablation studies; specifically, apart from different combinations of feature sets, we added another model variant – bidirectional LSTM for comparison. From Table 3, When all features except TF-IDF were used, the model with 2 LSTM layers outperformed the one using only 1 LSTM layer. Moreover, adding TF-IDF features further improved the F1 score to 0.6838. This indicates that increasing the number of layers allows the learning of more intricate patterns in the input data, especially with a larger group of features fed in.

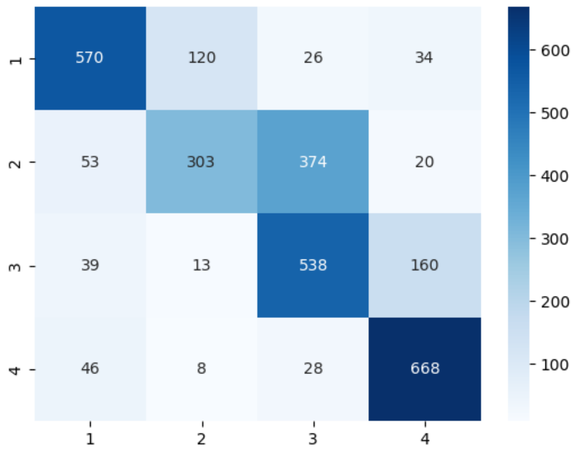


Figure 6: Confusion matrix of most optimal LSTM model

Bidirectional Long Short-Term Memory (BiLSTM) networks enhance the performance of standard LSTM by processing sequences in both forward and backward directions, thereby capturing long-range dependencies more effectively for improved context understanding. As shown in the ablation study in Table 4, we tried with 2 and 3 BiLSTM layers. The 3 BiLSTM layer variant outperformed the one with 2 BiLSTM layers across all three feature combinations. Upon considering all presented attributes, the tri-layered BiLSTM model attained a macro F1 score of 0.6957, which reinforces the idea that BiLSTM might be able to capture more nuanced relationships between the input features and lead to more accurate predictions. This demonstrates its superiority over the conventional LSTM model within the context of our experimental analysis.

4.5 Transfer Learning with BERT

Contextual word embedders like BERT (Devlin et al., 2018) have been shown to be more effective than traditional ones like GloVe in a variety of NLP tasks. The bidirectional transformer architecture of BERT enables context-dependent word representations, enhancing word sense disambiguation and comprehension of polysemy. The BERT models we used were pre-trained on vast amounts of text data. We further fine-tuned them on our training set, allowing for the transfer of general linguistic knowledge to the specific task of news classification. We expect that this transfer learning approach leads to improved model performance with limited labelled data. The

Features	Layers	Test accuracy	Test macro f1 score
Using all features except TF-IDF	2 BiLSTM layers	0.6893	0.6246
	3 BiLSTM layers	0.7007	0.6912
Using only word embedding features	3 BiLSTM layers	0.6853	0.6735
Using all features	3 BiLSTM layers	0.7070	0.6957

Table 4: BiLSTM ablation study

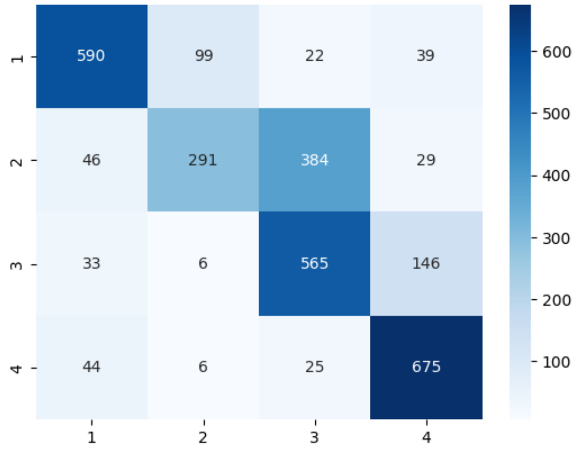


Figure 7: Confusion matrix of the most optimal BiLSTM model

architecture of BERT is shown in Figure 8.

We conducted experiments to evaluate the performance of three BERT models: `bert-base-cased`, `bert-base-uncased`, and `roberta-base`. The models were fine-tuned for 5 epochs, with `Adam` optimizer. At the end of each epoch, the models were evaluated in our evaluation set, which was randomly sampled from the training set with a 9:1 train-evaluation split ratio. The `bert-base-cased` model preserves the original casing of the text, whereas the `bert-base-uncased` model disregards the case information. `roberta-base` is a variant of BERT optimized with improved training techniques and larger amounts of data (Liu et al., 2019). The evaluation and test performance in terms of macro F1 score of these three models are listed in Table 5.

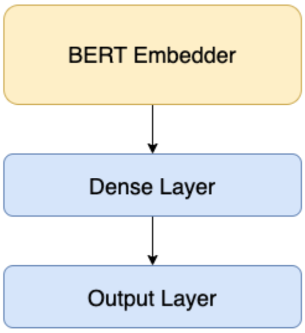


Figure 8: BERT Architecture

Model Name	Eval F1	Test F1
`bert-base-cased`	0.993	0.6062
`bert-base-uncased`	0.9965	0.5412
`roberta-base`	0.9968	0.5962

Table 5: BERT Evaluation and Test Macro F1

A large discrepancy in F1 score between evaluation and test set is observed. We further examine the confusion matrix. The confusion matrix of `roberta-base` experiment is as shown in Figure 9.

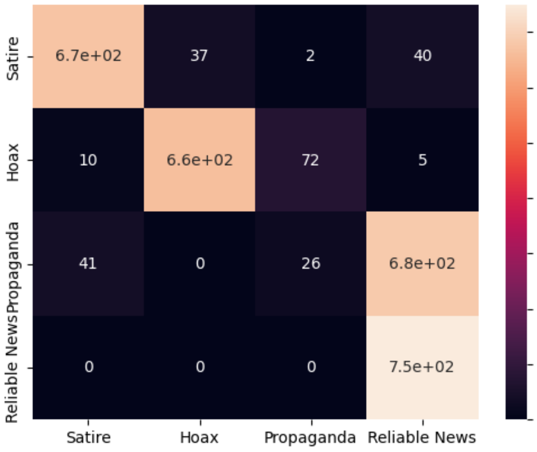


Figure 9: RoBERTa Confusion Matirx

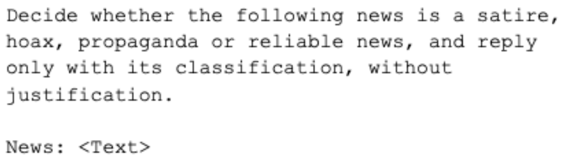
In Figure 9, it is clear that our BERT models struggle to differentiate between propaganda and reliable news within the test set. This substantial discrepancy can be attributed to the out-of-domain test sample and overfitting, as the models appear to capture exceedingly domain-specific features within the training set for propaganda.

To address the overfitting issue, we decreased the number of training epochs from 5 to 3, while increasing the dropout probability within the linear classifier from 0.1 to 0.3. The subsequent improvement in the F1 score from 0.5962 to 0.6401 provides evidence that overfitting may indeed be the underlying concern. However, due to constraints in time and computational resources, additional experiments aimed at mitigating overfitting and domain discrepancy were not pursued; instead, mitigation strategies will be suggested and expanded upon in this paper's discussion section.

4.6 Prompt-based Learning

In Section 4.5, we employed the widely adopted practice among NLP researchers of fine-tuning pretrained BERT models for our downstream news classification task. However, Radford et al. (2019) suggested that Large Language Models (LLMs) possess the capability to transfer their acquired knowledge to major NLP tasks without necessitating fine-tuning. This Prompt-based Learning approach, consequently, had become an increasingly prominent area of interest these years.

In this project, we adopted a zero-shot learning approach using the `gpt-3.5-turbo` model, which serves as the backbone for the current ChatGPT from OpenAI. To obtain the predicted labels from the `gpt-3.5-turbo` model, we employed the prompt in Figure 10 as part of our methodology. We substituted `<Text>` with each of the 3,000 test instances for every query. The resulting query responses were further processed and mapped to the four classes. Out of these 3,000 test instances, `gpt-3.5-turbo` declined to label 128 instances, deeming them nonnews. With manual inspection, we confirmed that some test instances did not resemble news articles. Therefore, we excluded these instances from the computation of our final test scores, which are presented in Table 6.



```
Decide whether the following news is a satire,
hoax, propaganda or reliable news, and reply
only with its classification, without
justification.

News: <Text>
```

Figure 10: Prompt for News Classification

Class	Precision	Recall	F1-score
Satire	0.88	0.98	0.93
Hoax	0.73	0.02	0.03
Propaganda	0.37	0.40	0.39
Reliable News	0.56	0.99	0.7
Accuracy			0.61
Macro avg			0.52

Table 6: Zero-shot Learning Test Results

Table 6 presents the test performance of `gpt-3.5-turbo` on 2,872 instances in a zero-shot learning setting. The results are reasonably satisfactory since no fine-tuning nor examples were included in the prompt. However, these results may be optimistic, due to the exclusion of 128 instances. Notably, `gpt-3.5-turbo` effectively differentiates satire news but struggles to distinguish hoax news from other classes.

5 Discussion

5.1 Confusion between Hoax and Propaganda

Upon examining the findings presented in the preceding section, it can be shown that the macro F1 scores attained by all the evaluated models did not exceed 0.75 within the test dataset. A detailed analysis reveals a prevalent shortcoming across the prediction outcomes with all the models examined: a considerable number of instances were erroneously classified as Propaganda when they should have been identified as Hoaxes. For the other three labels except for Hoax, on average the accuracy can achieve about 80%, while the overall performance is largely dragged behind due to the error in predicting Hoax.

From a natural language perspective, both hoaxes and propaganda involve the manipulation of information and language to deceive, influence, or mislead audiences. These overlapped linguistic features, such as persuasive language, emotional appeals, and exaggeration, can make it difficult for the model to discern the differences. Additionally, the training dataset exhibits an imbalance, wherein the number of Propaganda samples surpasses that of Hoax samples by more than a factor of two. Consequently, the model may develop a bias toward propaganda, resulting in suboptimal differentiation performance between the two classes.

A potential way to mitigate this issue could be more targeted feature engineering. In research like Barrón-Cedeño et al. (2019) and Martino et al. (2019), which aim to classify propaganda, they made use of Readability Measures like Flesch Reading Ease, Flesch-Kincaid Grade Level, Gunning Fog Index, and Coleman-Liau Index for features. This can potentially provide more valuable insights into the linguistic, structural, and semantic aspects of the text, and helps to better differentiate Propaganda and Hoax.

5.2 Out-of-domain Test Sample

Rashkin et al. (2017) suggested that the test sample is out-of-domain, as it was drawn from sources distinct from those of the training set. The substantial discrepancy between the evaluation and test results in our experiments further highlights this domain discrepancy. These observations imply that our models may not have effectively generalized to the test sample, potentially overfitting the training sample by extracting highly domain-specific features, such as the writing styles of authors and publishers. Consequently, as evidenced by our confusion matrices, all of our models exhibited difficulty in distinguishing between Hoax and Propaganda, and between Propaganda and Reliable News.

A naive approach to address the out-of-domain test issue is to incorporate a more diverse range of sources into the training sample. By training on instances from various sources, classifiers may be better equipped to extract more general features as opposed to those derived from a limited set of similar sources. However, this approach may not be preferred, as labelling news articles can be a time-consuming and tedious task.

An alternative method for tackling domain mismatch and mitigating overfitting is to employ Domain Adaptation techniques. Domain Adaptation seeks to enhance model robustness and generalization across different domains by encouraging encoders to generate similar representations for data originating from distinct domains but sharing the same labels (Ajakan et al., 2014; Motiian et al., 2017). This approach has the potential to improve the model's performance on out-of-domain data, thereby offering a more effective solution to the identified challenges. Due to the limited time and computing resources, we were unable to explore this field further. However, we believe that Domain Adaptation is a promising avenue for addressing the challenges associated with out-of-domain test data in future research.

5.3 Non-news Instances in Test Sample

We have realized that there exist instances where the input data contains the non-news text. This issue might have potentially hindered the results of our news classification. To address this, we can filter the input data based on its content. This can be done by training a classification model to differentiate between news and non-news text. These non-news texts will then be removed from the training data during the classification of reliable news. Another approach to addressing non-news data is to treat it as a separate class in a multi-class classification model. In this case, the non-news data would be given its label, and the model would be trained to predict whether a given text is non-news or one of the four classes.

6 Conclusion

In this study, we investigated the effectiveness of various machine learning models for classifying news articles as satire, hoaxes, propaganda, or reliable news. Our approach involved exploring different models, including logistic regression, SVM, LSTM, BiLSTM, Transformers, and Prompt learning, and conducting ablation studies to compare their performance. We also identified several challenges associated with the dataset and proposed several approaches, including domain adaptation and refining the training data to include non-news as another class.

Going forward, we believe that continued research and experimentation in this area will lead to new and innovative ways to tackle the task. We also recognize the importance of interdisciplinary collaboration and feel that insights from fields such as linguistics, psychology, and sociology can be combined. Linguistics, for example, can enable us to understand how news is manipulated to create false or misleading narratives. Psychology and sociology can give us valuable perspectives by allowing us to look into the motivations and behaviour of individuals who create and spread hoaxes and propaganda. By incorporating insights from related fields, we can develop more nuanced and effective approaches to address this task. We wish that our findings and suggestions will inspire further research and progress in this critical field.

References

Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, and Mario Marchand. 2014. Domain-adversarial neural networks. *arXiv preprint arXiv:1412.4446*.

Alberto Barrón-Cedeño, Israa Jaradat, Giovanni Da San Martino, and Preslav Nakov. 2019. Proppy: Organizing the news based on their propagandistic content. *Inf. Process. Manag.*, 56:1849–1864.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Cheikh Brahim El Vaigh, Thomas Girault, Cyrielle Mallart, and Duc Hau Nguyen. 2021. Detecting fake news conspiracies with multitask and prompt-based learning. In *MediaEval 2021-MediaEval Multimedia Evaluation benchmark. Workshop*, pages 1–3.

Pedro Henrique Arruda Faustini and Thiago Ferreira Covoes. 2020. Fake news detection in multiple platforms and languages. *Expert Systems with Applications*, 158:113503.

Tian Guo, Tao Lin, and Nino Antulov-Fantulin. 2019. Exploring interpretable LSTM neural networks over multi-variable data. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2494–2504. PMLR.

Saqib Hakak, Mamoun Alazab, Suleman Khan, Thippa Reddy Gadekallu, Praveen Kumar Reddy Maddikunta, and Wazir Zada Khan. 2021. An ensemble machine learning approach through effective feature extraction to classify fake news. *Future Generation Computer Systems*, 117:47–58.

Gongyao Jiang, Shuang Liu, Yu Zhao, Yueheng Sun, and Meishan Zhang. 2022. Fake news detection via knowledgeable prompt learning. *Information Processing & Management*, 59(5):103029.

Heejung Jwa, Dongsuk Oh, Kinam Park, Jang Mook Kang, and Heuiseok Lim. 2019. exbake: Automatic fake news detection model based on bidirectional encoder representations from transformers (bert). *Applied Sciences*, 9(19):4062.

Sebastian Kula, Michał Choraś, and Rafał Kozik. 2021a. Application of the bert-based architecture in fake news detection. In *13th International Conference on Computational Intelligence in Security for Information Systems (CISIS 2020) 12*, pages 239–249. Springer.

Sebastian Kula, Rafał Kozik, and Michał Choraś. 2021b. Implementation of the bert-derived architectures to tackle disinformation challenges. *Neural Computing and Applications*, pages 1–13.

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. *arXiv preprint arXiv:1907.11692*.

Giovanni Da San Martino, Seunghak Yu, Alberto Barrón-Cedeño, Rostislav Petrov, and Preslav Nakov. 2019. Fine-grained analysis of propaganda in news article. In *Conference on Empirical Methods in Natural Language Processing*.

Saeid Motiian, Marco Piccirilli, Donald A Adjeroh, and Gianfranco Doretto. 2017. Unified deep supervised domain adaptation and generalization. In *Proceedings of the IEEE international conference on computer vision*, pages 5715–5725.

Rukhma Qasim, Waqas Haider Bangyal, Mohammed A Alqarni, Abdulwahab Ali Almazroi, et al. 2022. A fine-tuned bert-based transfer learning approach for text classification. *Journal of healthcare engineering*, 2022.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Hannah Rashkin, Eunsol Choi, Jin Yea Jang, Svitlana Volkova, and Yejin Choi. 2017. Truth of varying shades: Analyzing language in fake news and political fact-checking. In *Proceedings of the 2017 conference on empirical methods in natural language processing*, pages 2931–2937.

Inna Vogel and Meghana Meghana. 2020. Detecting fake news spreaders on twitter from a multilingual perspective. In *2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 599–606. IEEE.

Acknowledgements

We would like to thank our professor A/P Kan Min-Yen, Dr Christian von der Weth for their efforts in designing CS4248 course materials, our tutors, and our PhD mentor Hai Ye.
We also would like to acknowledge our efforts for this project.

Statement of Independent Work

1A. Declaration of Original Work. By entering our Student IDs below, we certify that we completed our assignment independently of all others (except where sanctioned during in-class sessions), obeying the class policy outlined in the introductory lecture. In particular, we are allowed to discuss the problems and solutions in this assignment, but have waited at least 30 minutes by doing other activities unrelated to class before attempting to complete or modify our answers as per the class policy.

We have documented our use of AI tools (if applicable) in a following table, as suggested in the NUS AI Tools policy[1]. This particular document did not use any AI Tools to proofcheck and was constructed and edited purely by manual work.

Signed, [A0194551B, A0205182N, A0201185R, A0194494N, A0218236H]

[1] https://libguides.nus.edu.sg/new2nus/acadintegrity, tab “AI Tools: Guidelines on Use in Academic Work”