Towards Understanding Linguistic Features for Unreliable News Classification

A0206068H, A0206224U, A0149818N,
A0201642W, A0199359E, A0200112M
Group 16
Mentored by Ou Longshen
{jonathantyt, yeohuaizhe, e0014722}@u.nus.edu
{e0415451, fanmin.jian, seantaysl}@u.nus.edu

Abstract

With the widespread use of the Internet and social media as sources of information, the reliability of news has become a prominent global issue. Unreliable news has the potential to impact all communities, from misinformation to political radicalism and even violent extremism. In this work, we conduct a series of document classification studies on the Labeled Unreliable News (LUN) dataset, using methods including logistic regression, simple neural networks, and transformer models. We apply a simple neural network to the combination of TF-IDF and transformer embedding vectors, without additional fine-tuning on the latter to achieve a macro-F1 score of 0.797 on the LUN test set, surpassing the performance of prior works on LUN. Finally, we provide additional ablation and explainability studies toward a better understanding of linguistic features for unreliable news classification. The project code is available at this GitHub repository[1].

1 Introduction

The term "Fake News" has become mainstream in modern language, most notably catalyzed by the 2016 US elections during which websites published falsified claims and biased stories to capitalize on social media profitability (Allcott and Gentzkow, 2017). While the term has risen in popularity since then, unreliable news has been around since the 1890s. Unreliable news can be extremely dangerous as it has the potential to allow malicious entities to push harmful agendas by shaping the beliefs and opinions of their audiences (Schackmuth, 2018). This is exacerbated by the proliferation of the internet and social media, which has resulted in global sharing of (mis)information at unprecedented scale and speed. In light of this, manual verification of news content is simply no longer feasible.

Governments and researchers have instead attempted to detect fake news through automatic fact-checking or leveraging machine learning models. While the former can be effective, a complete collection of verifiable facts are both expensive and often impractical to maintain. Fortunately, machine learning models are often designed to make predictions on text-based inputs only, which allows them to pick up on linguistic features that can differentiate between reliable and unreliable news. Therefore, this work presents a study on the language and semantic structure of news media by applying machine learning methods to the Labeled Unreliable News (LUN) dataset (Rashkin et al., 2017), in which text samples are to be classified as one of reliable, hoax, satire or propaganda. We demonstrate a method that performs competitively on LUN and use it to compare the key features between trusted and unreliable news extracts.

2 Related Work

2.1 Text Classification

Text classification has many applications, including sentiment analysis, document tagging and fake news identification. It can be formulated as a supervised learning problem, in which machine learning algorithms learn inherent associations between text and their assigned labels (Minaee et al., 2021). The text classification problem is commonly formalized in Equation 1 below.

$$y = argmax(z) = argmax(f(x)) \quad (1)$$

Specifically, given some text x as input, a model represented by function $f(x)$ is applied to obtain a vector of probabilities $z \in \mathbb{R}^n$, where n is the number of classes. The argmax of z is then taken as the final prediction y.

2.2 Automated Fake News Detection

In most literature, fake news detection is formulated as a binary classification problem in which news is either real or fake (Oshikawa et al., 2020). However, natural language is often complex, and fake news is intentionally blended with true information for various purposes. Therefore, researchers turn to multi-class classification where labels are assumed to be independently distributed (Wang, 2017). Similarly, we focus on multi-class classification for a more robust detection of fake news using the LUN dataset (Rashkin et al., 2017).

Some other methods utilise a repository of known facts to supplement text features in unreliable news classification, which can be used to directly refute inaccurate statements. For example, CompareNet (Hu et al.,

[1] https://github.com/supperted825/LUN-TextClassification

2021) uses a graph neural model for sentence embedding and constructing a directed entity-relationship (ER) graph from the text inputs. The latter is then compared to a knowledge base in the form of another ER graph, and the sentence embeddings and comparison features are then used as inputs to a classifier. While we reference the performance of CompareNet on the LUN test set, we do not directly compare it with our methods. This is because CompareNet involves the use of external knowledge and is arguably more focused on knowledge representation than text classification.

2.3 Classical Machine Learning Methods

We consider classical machine learning to be methods that do not utilize neural networks. Popular techniques include generative methods such as Naive Bayes (Hand and Yu, 2001) and discriminative methods such as logistic regression (LR) or maximum entropy (Berkson, 1944). Tree-based methods such as random forest (Hassan et al., 2017) and XGBoost classifier (Rao et al., 2021) have also been used. These methods often use some form of text embeddings, such as TF-IDF, as input.

For the LUN dataset, Rashkin et al., 2017 found that the LR paired with TF-IDF and Linguistic Inquiry and Word Count (LIWC) features significantly outperformed the simple baseline of random classification. While classical machine learning methods remain widely used, they rely on the user for extensive feature engineering to achieve good performance in practice (Minaee et al., 2021). Hence, we consider such classical methods, particularly logistic regression, as the baseline.

2.4 Text Embedding

Since most machine learning models require structured numerical inputs, text embedding is a critical component of text-preprocessing. The simplest is the bag-of-words (BoW) model (Luhn, 1958), which produces a vector of counts for each n-gram in each document, where the range of n is an adjustable parameter. Term frequency-inverse document frequency (TF-IDF) (Salton and Buckley, 1988) improves upon BoW by normalizing the frequency of each word against how often it appears across different documents to suppress the weighting of frequent but unimportant terms.

BoW-like methods ignore ordered dependencies between terms. While dependencies can be modeled via higher order n-grams, this has limited effectiveness due to increasing sparsity from the rarity of such higher order terms. Thus, only short-range dependencies modeled by only up to trigrams are typically used in practice.

Vector space models such as Word2Vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014) have also gained popularity in recent years, which utilize self-supervised contrastive learning to represent the meaning of words as high dimensional vectors. This has also naturally been extended to the paragraph vector (Le and Mikolov, 2014) for use with entire documents. Our work on vector space models is in Appendix B.

2.5 Deep Learning Methods

To overcome the limitations of BoW methods, several deep learning methods, most notably the recurrent neural networks (RNNs) such as long-short-term memory (LSTM) and gated recurrent unit (GRU) have been shown to be capable of representing long-term dependencies across text sequences (Li et al., 2022). This is done with the addition of a hidden state that is recurrently passed through the network with each new input, acting as a contextual representation of previous words.

Later, Vaswani et al., 2017 introduced the Transformer architecture, which features a global attention mechanism to allow dependencies between words in sequences to be modeled effectively regardless of how far they are from one another. BERT (Devlin et al., 2019) is one such transformer-based language model that is trained using two objectives: masked token prediction and next sentence prediction. More significantly, BERT can be fine-tuned by adding additional layers and training on downstream natural language processing tasks, achieving good performance on question answering, named entity recognition, and text classification. Given the promising performance of the BERT models on a variety of NLP tasks, we chose to primarily work with the BERT-based models within this study.

2.6 BERT-based Models

Other works such as DistilBERT (Sanh et al., 2019) and RoBERTa (Zhuang et al., 2021) have improved the original BERT model through knowledge distillation and domain aware pretraining respectively. DeBERTa (He et al., 2020) introduced disentangled attention and an enhanced mask decoder to encode the absolute position of tokens, and the most recent DeBERTaV3 (He et al., 2021) uses the adversarial replaced token detection pretraining objective from ELECTRA (Clark et al., 2020) for more effective pretraining.

2.7 Bag-of-Words & Transformers

BoW methods like TF-IDF identify keywords within a document, while transformer-based methods like DeBERTaV3 conduct holistic sequence modeling to represent semantic and linguistic structure. Thus, prior attempts to combine the two are not uncommon.

For example, Chen et al., 2020 found that incorporating TF-IDF weighting into the BERT mask layer significantly improves the performance of various downstream tasks. Alternatively, Lim and Tayyar Madabushi, 2020 found that incorporating TF-IDF representations with the embedded BERT features using simple concatenation improves the performance of a variety of sub-tasks. In both examples, the authors state that introducing some form of corpus-level representation in the form of TF-IDF while using BERT improves the downstream performance of the model. Hence, we also aim to build on such prior works to improve the performance of our model for unreliable news classification, by providing a larger variety of signals in the input features.

3 Corpus Analysis

3.1 Dataset Exploration

The LUN dataset comprises 48,652 unique fake news extracts divided into 4 classes: satire, hoax, propaganda, and reliable, with the number of samples in each class shown in Table 1. The number of samples in each class is imbalanced, with propaganda being the most common and hoax being the least. However, as there are still nearly 7,000 hoax samples, this does not yet represent an immediate need for over or undersampling. The accompanying test set includes 3,000 balanced samples with 750 news extracts per class.

Class	No. Samples
Propaganda	17,870
Satire	13,911
Reliable	9,932
Hoax	6,939

Table 1: No. Samples per News Class

We estimated the number of tokens in the documents of each class by applying word punctuation tokenization to the dataset, shown in Figure 1. This is referred to this as an estimate because transformers use a BPE-like (Byte Pair Encoding) tokenizer which produces a different number of tokens. Our analysis revealed that document lengths for each class follow a clear unimodal distribution of varying lengths, with the exception of satire which is bimodal. We also observed that each class has a number of outlier documents with a higher number of tokens, the most obvious of which occurs in the 'propaganda' class. Finally, the distributions of each class between the training and test sets is quite different, even if the difference in dataset size is accounted for. This is an interesting observation that will be further discussed in later sections of this report.

3.2 Bag of Words Analysis

We conducted a bag of words analysis using term frequency and TF-IDF on the LUN dataset. For each class, we calculated and summed together the count vectors and TF-IDF vectors of the documents to form a representative vector, from which the top 5000 terms (unigrams and bigrams) are extracted. Figure 2 illustrates the proportion of overlap between the top terms of each class. Each is shown to be fairly similar to the others, as there is a fairly large overlap of between 60% to 69% in top TF-IDF terms. Thus, TF-IDF features alone may not be sufficient to achieve good classification performance.

The top terms of each class also reveal that the documents in the LUN dataset are particularly focused towards western political news. For example, "obama", "trump", "president", "clinton" are top terms in the hoax documents, while "government", "police", "war", "state" are top terms in propaganda. This indicates that the training set may be highly specialized in the political domain and has the potential to hinder the model's performance on general forms of fake news.

3.3 Punctuation Analysis

We calculated the frequency of various punctuations to further gather insights about our dataset. Since we are interested in primarily distinguishing reliable from unreliable news extracts, we take the top 10 punctuation from the reliable class and plot their normalized counts across the different classes as shown in Figure 3. Full stops and commas are among the top few punctuations which is expected in English sentence structures. Reliable news seem to use the most hyphens relatively. Another interesting observation is the common usage of apostrophe in satire and reliable classes, but very little to no usage of apostrophe in hoax and propoganda classes. The hoax class has the largest usage of question marks compared to the other classes.

Further dataset exploration is in Appendix C.

4 Methodology

4.1 Dataset Cleaning

The original training dataset is first divided 80:20 into stratified train and validation splits, which we refer to as training and validation sets respectively. We remove samples with estimated lengths of less than 10 tokens, since they do not provide sufficient signal to be classified correctly. On the other hand, some samples are much longer and will be truncated when preprocessed with transformer tokenizers. To better utilize training data, we preemptively divide long samples in our training set into multiple short documents with the same class. We ensure that each document is coherent by splitting only on complete sentences. The algorithm for document splitting is detailed in Appendix A.

We explicitly do not clean the text by removing abbreviations and spelling errors, as these may be informative features for differentiating between reliable and unreliable news. Furthermore, we apply transformer-based methods which are trained in a self-supervised fashion on unprocessed corpora. Thus, cleaning is often not required and can even be harmful to model performance.

4.2 Dataset Augmentation

To regularize the training process and improve the generalization capabilities of our trained models, we apply a series of augmentations with the *nlpaug* package (Ma, 2019) to each sample in the training set: random word swap, random word delete, random sequence crop, and synonym imputation. Each augmentation is applied with a probability of 0.3, and affects a minimum of 1 and a maximum of 10 words in the sample. This is applied once to each sample to yield an additional sample per original training sample, and increases the size of our train set to 75,656 unique samples.

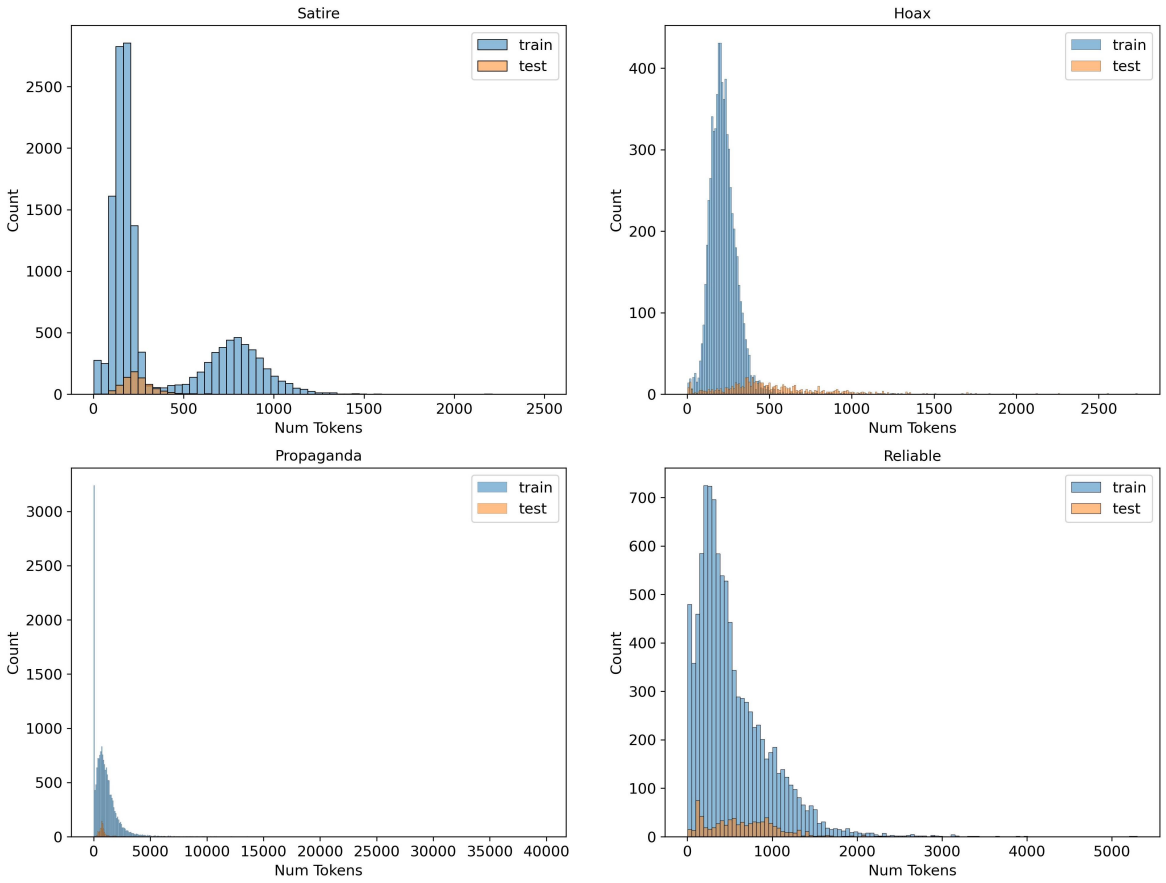

Figure 1: Histogram of Document Lengths for Each Class in Training & Test Set. A single training sample with length of 140,000 has been removed from the propaganda class to improve the quality of visualization.

Figure 2: Heat-map of Overlap Proportion between Top 5K TF-IDF Ranked Terms of Documents in Each Class

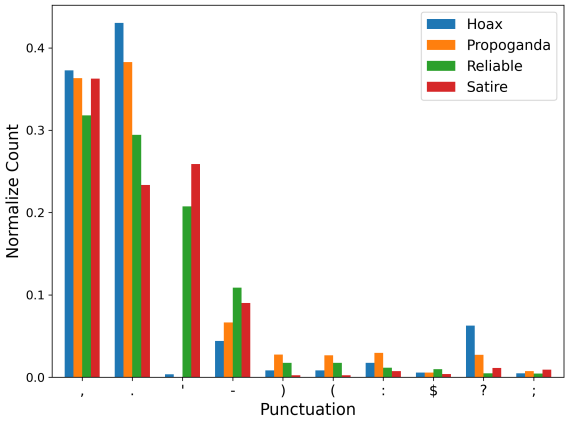

Figure 3: Normalized Count per Class for Top 10 Punctuation used in Reliable News Class

4.3 Baseline Methods

We identify logistic regression and a simple neural network (NN) as our baselines, using TF-IDF features as inputs. An n-gram range of (1,2) is used, and English stopwords are removed. Additionally, terms with a document frequency greater than 0.8 and those appearing in less than 10 documents are removed. A maximum number of 5096 features is used. To keep the baseline simple and easily reproducible, we did not conduct any additional manual feature engineering.

Logistic regression looks for linear relationships between input features of a sample and the probability of that sample belonging to a given class. For prediction, the softmax function is applied to the model's outputs to obtain the probability distribution over the target classes.

Finally, the argmax of that output is taken to be the predicted class of the particular sample.

The simple NN comprises two hidden layers, each having the same number of neurons as the number of TF-IDF features (5096). Then, the final classification layer of size 4 outputs logits to which softmax is once again applied to obtain a probability distribution over classes. ReLU activation and dropout of $p = 0.5$ is applied following each hidden layer.

4.4 DeBERTaV3

Since none of the prior methods on LUN utilized transformer methods, we chose to apply DeBERTaV3 to the unreliable news classification problem to benchmark the performance of state-of-the-art methods.

The DeBERTaV3 model is loaded as the backbone of our classification model, and we append 3 linear layers with the same dimensions as DeBERTaV3 output features for classification. During the forward pass, mean pooling with attention masking is conducted on the output layer of the transformer, before being passed through the remaining layers. Similarly, ReLU activation and dropout of p=0.5 is applied to each hidden layer.

4.5 DeBERTaV3 with TF-IDF

We build upon prior works to show that TF-IDF features and transformer embeddings represent distinct features from the text, and the direct concatenation of the two can be beneficial to unreliable news classification. We refer to this method as DeBERTa-TFIDF.

Similar to Section 4.4, transformer embeddings are first obtained by forward pass through the DeBERTaV3 model, followed by mean pooling and a linear layer. The embeddings are then concatenated with TF-IDF features before passing through 2 more hidden layers before the classification layer. The first linear layer shares the same hidden size as DeBERTaV3 (768), and the subsequent 2 layers are of the concatenated embedding dimension (768 + 5096). Activations and dropout are applied as done in the baseline methods, and TF-IDF vectorization parameters are kept the same as in Section 4.3. We illustrate this architecture in Figure 4.

The DeBERTa-TFIDF model can be trained in a variety of ways. For example, one can choose to train every parameter in the model, or freeze the DeBERTaV3 backbone and train only the linear layers of the classification head. Alternatively, every transformer layer except the last n layers can be frozen during fine-tuning. We discuss the effectiveness of such methods in Section 5.2.

4.6 Training & Evaluation

Since the problem posed is imbalanced multi-class classification, we monitor the accuracy, precision, recall and F1 scores for each class, and consider the respective macro-average of the above metrics to measure the overall performance of each model. In line with prior works (Rashkin et al., 2017; Hu et al., 2021), we use macro-F1 score as the primary evaluation metric.

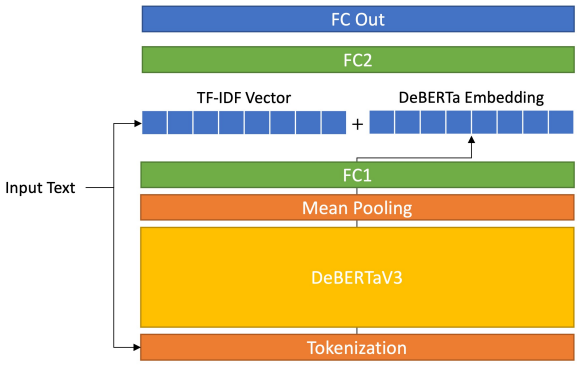

Figure 4: Illustration of the Proposed DeBERTa-TFIDF Architecture. Dropout layers are omitted for simplicity.

All neural network based methods are trained using cross-entropy loss and optimized with the AdamW optimizer for 5 epochs, at a learning rate of 5e-5. A batch size of 16 is used. The logistic regression is optimized using the L-BFGS solver with a regularization strength of 1.0, paired with the L2-penalty. The highest macro-F1 score achieved by a model across 5 epochs is taken to be the final score for that model.

4.7 Feature Importance

In this work, we employ the integrated gradient method (Sundararajan et al., 2017) to learn word importance during text classification of each class, using Captum (Kokhlikyan et al., 2020). Integrated gradients computes the attribution of each feature of a deep learning model based on the gradient of the model's output with respect to the input. The attribution score quantifies the contribution of a feature to the final prediction with respect to a baseline. A positive attribution indicates a feature that increases the prediction likelihood of a class and negative attribution indicates the opposite. Two inputs are required for integrated gradients: the original input comprising the text tokens from the news extract, and the baseline input, which is an empty extract.

Due to limited computation resources and time, only 20 samples were randomly sampled from each class in the test set for feature importance analysis at the dataset level. For every news extract, the attribution scores of the tokens were calculated, and the average score for each token within an extract was recorded. For this analysis, we removed punctuation and stopwords, which were initially found to have significant attribution scores but limited value for explainability. The top 20 words with positive attribution and top 20 words with negative attribution for each class are used to analyze word importance across classes.

5 Results & Discussion

5.1 Performance on LUN Test Set

Table 2 shows the performance of the logistic regression and simple NN baselines, along with DeBERTa-only

and DeBERTa-TFIDF methods. The latter two experiments are conducted twice, once with fine-tuning of the transformer backbone (FT), and once without.

Method	Prec.	Recall	F1
Rashkin et al., 2017	-	-	0.650
Hu et al., 2021	0.729	0.690	0.683
LogReg (w/o LIWC)	0.750	0.730	0.720
Simple NN	0.760	0.747	0.741
DeBERTa	0.737	0.719	0.695
DeBERTa (FT)	0.313	0.250	0.101
DeBERTa-TFIDF	0.807	0.802	0.797
DeBERTa-TFIDF (FT)	0.690	0.690	0.659

Table 2: Macro-Averaged Results on LUN Test Set

The best performance of 0.797 macro-F1 is given by the DeBERTa-TFIDF model without fine-tuning of the transformer backbone, which surpasses logistic regression and simple neural network baselines by a comfortable margin. This demonstrates that transformer-based embeddings are able to supplement TF-IDF features to improve text classification performance. At the same time, the DeBERTa model without TF-IDF and no fine-tuning performs poorer than the baselines, which suggests that TF-IDF features still contribute more than transformer-based embeddings to fake news classification performance on the LUN dataset.

Notably, our baseline methods already outperform prior works on LUN by Rashkin et al., 2017 and Hu et al., 2021, which achieve macro-F1 scores of 0.650 and 0.683 respectively. Our logistic regression baseline is essentially the same as the method used by Rashkin et al., 2017, except that it does not use LIWC features. Thus, we attribute our superior performance to effective data pre-processing and TF-IDF hyperparameter tuning.

5.2 Transformer Fine-tuning

Both methods that conduct fine-tuning on the DeBERTaV3 backbone perform significantly worse than their counterparts with no fine-tuning. This is a surprising result, as fine-tuning should allow the model to take advantage of domain specific knowledge to adjust the transformer's attention layers.

We hypothesized that the poor performance of backbone fine-tuning was the result of instabilities during learning due to the relatively large number of parameters in DeBERTa. This is apparent in DeBERTa (FT), as performance has significantly degenerated to a macro-F1 score of 0.101. To combat this, we explored various techniques such as lowering the backbone learning rate (LBLR), layerwise learning rate decay (LLRD) and partial fine-tuning of only the last transformer layer (Sun et al., 2019). Results are shown in Table 3.

LLRD (Eq. 2) is applied with a factor of 0.9 to each subsequent transformer layer, such that the layer closest to the input layer has the lowest learning rate, and the 12^{th} and final layer keeps the same learning rate of 5e-5.

Configuration	Val. F1	Test F1
DeBERTa-TFIDF (FT)	1.00	0.65
w. LBLR	1.00	0.76
w. LLRD	0.90	0.58
w. Partial FT	1.00	0.69

Table 3: Performance Results on LUN for Various DeBERTa-TFIDF Fine-tuning Configurations

LBLR is achieved by applying a learning rate of 5e-7 to the DeBERTaV3 backbone. For both methods, the learning rate of the classification head is kept at 5e-5.

$$lr_{n-1} = 0.9 * lr_n \quad (2)$$

LLRD appears to worsen performance, which we believe is caused by the connection introduced between the embedding layer and the output layer for the enhanced mask decoder in DeBERTa. Thus, using different learning rates across early and later layers result in conflicting gradient flows and disrupt the learning process. This is also supported by the weak performance of DeBERTa-TFIDF w. LLRD, even on the validation set.

Among all fine-tuning configurations, LBLR achieves the highest performance on the LUN test set with a macro-F1 score of 0.755. This is followed by the partially fine-tuned and then the base DeBERTa-TFIDF (FT) models. This trend shows that a greater degree of fine-tuning is in fact negatively correlated with better test set performance. Our other attempts to stabilize the fine-tuning process include further reduced learning rate and higher dropout layer probability. However, those methods did not yield improvements and are thus not further discussed in this report.

5.3 Dataset Drift

High training but low test performance is a common symptom of overfitting in machine learning models. Table 3 shows that fine-tuned models can perform well on unseen validation data, but not on the test set. Furthermore, we observed that the LUN dataset is noisy, with many indiscernible news extracts even for humans. Since our attempts to improve the performance of fully fine-tuned transformer models on the test set were unsuccessful, we conclude that the test set domain is significantly different from the training data, and conduct additional dataset analysis to corroborate this claim.

Previously, we observed significant differences in the number of tokens per sample between train and test sets for different classes in Figure 1. We further compare the difference between training and test set distributions by calculating the percentage of out-of-vocabulary (OOV) terms with respect to an external natural language lexicon. The Wasserstein Distance (WD) test was used to compare the probability distributions of the training and test dataset, using Evidently (EvidentlyAI, 2021). The WD test seeks to test whether two distributions in the form of samples are different, with the null hypothesis

that the distributions are the same and the alternative hypothesis that they are not.

The comparison of train and test token count distributions yields a WD test statistic of 0.15, while that of train and test %OOV distributions yields a WD test statistic of 0.2. Both test statistics exceed the critical value of 0.1, indicating statistically significant differences between training and testing sets. Figure 5 additionally shows a significant number of test documents lying outside the expected %OOV range (green band), which is conditioned on the distribution of the training set.

Our analysis shows that there is an apparent dataset drift between the training and test set, and explains the different performance of the model on the validation set as compared to the test set. Notably, this explains why test set performance is worsened with more fine-tuning, as the model is increasingly optimised to a different underlying data distribution from the test set.

6 Ablation Studies

6.1 Classification Head Layers

We verify that the improved performance of DeBERTa-TFIDF comes from the combination of transformer and TF-IDF features, and not the increased size of layers in the classification head. To do so, we train a simple NN as done in the baseline, but now increase the size of hidden layers to 768 + 5096 to match DeBERTa-TFIDF. Results are shown in Table 4. The NN with increased parameters has similar if not worse performance than the baseline NN, which supports our claim that transformer embeddings can supplement TF-IDF features.

Method	Prec.	Recall	F1
Simple NN	0.760	0.747	0.741
+ Extra FC dim.	0.746	0.739	0.735
DeBERTa-TFIDF	0.807	0.802	0.797

Table 4: Comparison of simple NN performance with increased hidden layer size. Macro metrics are shown.

6.2 Error Analysis

Table 5 shows the test-set performances of the DeBERTa & DeBERTa-TFIDF models discussed in Section 5.1. It can be seen that the DeBERTa model already performs well on *satire* & *reliable*. Instead, the addition of TF-IDF features drastically improved the F1 score of the model on *hoax* by 0.348, while the performance on *propaganda* was improved by 0.053.

We explore our model's weaknesses with a qualitative observation of model predictions and token attributions using the integrated gradients method described in Section 4.7. Due to space limitations, we provide figures for sample-level token attributions in Appendix D.

Firstly, news extracts belonging to other categories are observed to often be misclassified as hoax. Upon a closer inspection, these samples often contain entities and words with political themes. This could suggest that political terms are skewing the distribution of the model predictions towards the hoax category. Intuitively, we would expect many satirical and reliable news articles to also contain political entities within the article, which indicates an over-generalization by the model on this heuristic as a decision boundary for identifying hoax. This could also explain why the hoax category is the poorest performing class for our model.

Class	DeBERTa	+TF-IDF
Satire	0.866	0.899
Hoax	0.331	0.679
Propaganda	0.653	0.706
Reliable	0.929	0.902
Macro Avg.	0.695	0.797

Table 5: Class-level F1 Scores for DeBERTa & DeBERTa-TFIDF models on LUN Test Set

Next, some samples have been misclassified as propaganda with high confidence. The words that positively contribute to misclassification often revolve around highly polarizing topics such as war, religion, and politics. Interestingly, a large proportion of such misclassified extracts belong to the hoax class. While both hoax and propaganda articles often contain political topics, it appears the model draws a decision boundary on the number of occurrences of named political entities.

Finally, we highlight the critical mistakes where the model misclassifies unreliable news as reliable. Political terms such as "minister" or "commission" are once again observed to contribute to the misclassification, whereas other terms such as "attack" or "alliance" lower these probabilities. Unlike hoax and propaganda, there are no immediately obvious general patterns in the misclassification of samples as reliable news. We hypothesize that this is mainly due to the diverse nature of reliable news articles which can range across many different topics. In view of this, it would not be surprising to observe that the model classifies extracts with a diverse range of topics, or uses seemingly random word attributions to classify a document as reliable.

6.3 Feature Importance

The top 40 most important tokens are obtained as described in Section 4.7, and the result for *hoax* is shown in Figure 6. Due to space limitations, the results for the other classes are shown in Appendix E.

1. **Hoax.** For hoax, as highlighted earlier, a heavy emphasis is placed on political entities and words that are commonly used to incite political discourse. For example, explicit names of previous presidential candidates like Obama and Bush were present. Strong words such as treason, war, and incompetent are potential phrases used by politicians to establish their rhetoric. Hence our model deems a politically driven extract to be of the *hoax* class.

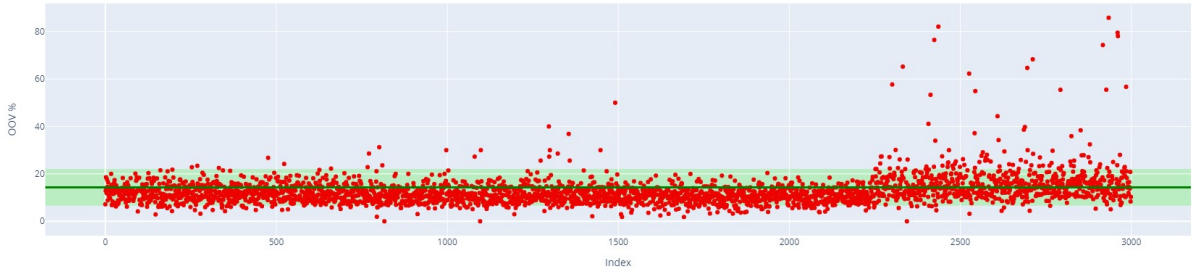

Figure 5: % Out of Vocabulary (OOV) Words Within each sample in the Test Set.

2. **Satire.** Satirical extracts are usually witty, ironic, and exaggerated. From the feature attribution scores, extreme or sarcastic words such as nothing, incredible and little have high attribution scores when predicting for satire. Words with negative attribution are nouns or verbs that are not applicable in the context of ridicule. Therefore, our model is correctly focusing on words that embody the characteristics of satirical news, which explains its relatively better performance in this class.

3. **Propaganda.** For propaganda, many words with high positive attribution are addressed to the general public (eg. children, people, world). This aligns well with an intuitive understanding of propaganda, as such articles are written to reach a wide audience. The model has likely identified that propaganda extracts write to communicate with a larger target audience in order to garner influence.

4. **Reliable.** A significant number of positively attributed words are related to money or economy, such as income, GDP, private and payroll. A number of reliable articles within the dataset consist of match scores for various sporting events, hence it is observed that days of the week such as "Wednesday" and potential match scores such as "23" and "25" place within the top 20 contributing terms.

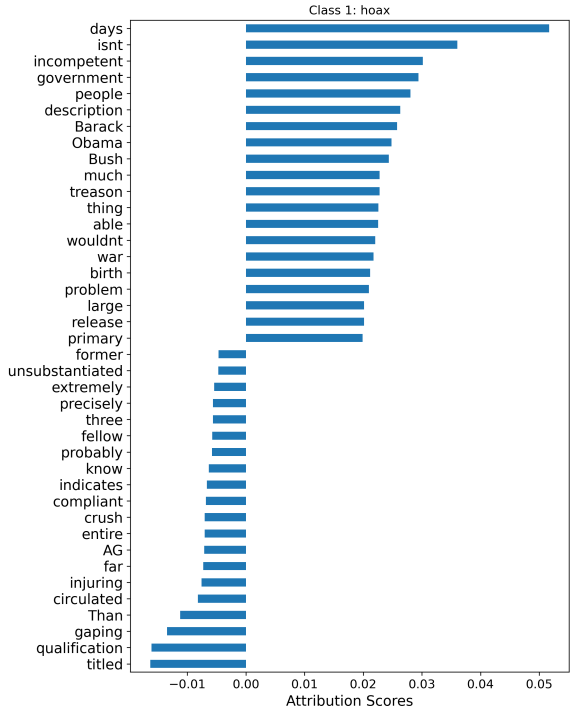

Figure 6: Top 20 Positively Attributed Tokens & Top 20 Negatively Attributed Tokens for the Hoax News Class

7 Conclusion

In this work, we have conducted text classification of unreliable news on the LUN dataset by applying a combination of TF-IDF BoW features and sequence modelling-based transformer embeddings. By using both embeddings obtained from TF-IDF and DeBERTaV3, the resulting DeBERTa-TFIDF method yields an improved test set macro F1 score of 0.797, showing that both features play complementary roles in achieving good fake news classification performance.

In addition, we provide in-depth analysis on the differences in data distribution between training and test sets of the LUN dataset, to explain why the fine-tuning of the transformer backbone fails to achieve good performance on the test set. Namely, the distribution of training data is too significantly different from the test data, such that further optimisation on the training set leads to poorer test set performance.

We then conclude our report with error analysis and explainability studies to understand the weaknesses of our method. While the model struggles with certain samples, it can identify characteristics of each news category that align well with human intuition to make good predictions on the LUN dataset. Furthermore, our work highlights the importance of a carefully crafted dataset for general fake news classification, since models tend to focus on domain-specific keywords for predictions.

Our work has primarily focused on the LUN dataset, which is heavily dominated by content surrounding the 2016 US elections. Future research may apply our approaches to a more general news corpus to yield more general insights towards general fake news classification. In addition, they may explore splitting reliable news into distinct categories to better explain the model's decision boundary between unreliable and reliable news.

Acknowledgements

We would like to offer our appreciation to our project mentor, Mr. Ou Longshen for his guidance and inputs throughout the duration of our work on this project, as well as to the rest of the CS4248 teaching team for an excellent and well taught course.

Statement of Independent 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 declare that this particular document did not use any AI tools for proofreading and was constructed and edited purely by manual work.

Signed, A0206068H, A0206224U, A0149818N, A0201642W, A0199359E, A0200112M

References

Hunt Allcott and Matthew Gentzkow. 2017. Social media and fake news in the 2016 election. *The Journal of Economic Perspectives*, 31(2):211–235.

Joseph Berkson. 1944. Application of the logistic function to bio-assay. *Journal of the American Statistical Association*, 39(227):357–365.

Weilong Chen, Xin Yuan, Sai Zhang, Jiehui Wu, Yanru Zhang, and Yan Wang. 2020. Ferryman at SemEval-2020 task 3: Bert with TFIDF-weighting for predicting the effect of context in word similarity. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 281–285, Barcelona (online). International Committee for Computational Linguistics.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: pre-training text encoders as discriminators rather than generators. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

EvidentlyAI. 2021. Evaluate and monitor ML models from validation to production. https://github.com/evidentlyai/evidently.

David J. Hand and Keming Yu. 2001. Idiot's bayes: Not so stupid after all? *International Statistical Review / Revue Internationale de Statistique*, 69(3):385–398.

Naeemul Hassan, Fatma Arslan, Chengkai Li, and Mark Tremayne. 2017. Toward automated fact-checking: Detecting check-worthy factual claims by claimbuster. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17, page 1803–1812, New York, NY, USA. Association for Computing Machinery.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing. *CoRR*, abs/2111.09543.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced BERT with disentangled attention. *CoRR*, abs/2006.03654.

Linmei Hu, Tianchi Yang, Luhao Zhang, Wanjun Zhong, Duyu Tang, Chuan Shi, Nan Duan, and Ming Zhou. 2021. Compare to the knowledge: Graph neural fake news detection with external knowledge. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 754–763, Online. Association for Computational Linguistics.

Narine Kokhlikyan, Vivek Miglani, Miguel Martin, Edward Wang, Bilal Alsallakh, Jonathan Reynolds, Alexander Melnikov, Natalia Kliushkina, Carlos Araya, Siqi Yan, and Orion Reblitz-Richardson. 2020. Captum: A unified and generic model interpretability library for pytorch. *CoRR*, abs/2009.07896.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1188–1196, Bejing, China. PMLR.

Qian Li, Hao Peng, Jianxin Li, Congying Xia, Renyu Yang, Lichao Sun, Philip S. Yu, and Lifang He. 2022. A survey on text classification: From traditional to deep learning. *ACM Trans. Intell. Syst. Technol.*, 13(2).

Wah Lim and Harish Tayyar Madabushi. 2020. Uob at semeval-2020 task 12: Boosting bert with corpus level information.

H. P. Luhn. 1958. The automatic creation of literature abstracts. *IBM Journal of Research and Development*, 2(2):159–165.

Edward Ma. 2019. Nlp augmentation. https://github.com/makcedward/nlpaug.

Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*.

Shervin Minaee, Nal Kalchbrenner, Erik Cambria, Narjes Nikzad, Meysam Chenaghlu, and Jianfeng Gao. 2021. Deep learning–based text classification: A comprehensive review. *ACM Comput. Surv.*, 54(3).

Ray Oshikawa, Jing Qian, and William Yang Wang. 2020. A survey on natural language processing for fake news detection. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 6086–6093, Marseille, France. European Language Resources Association.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

V. Chandra Shekhar Rao, Pulyala Radhika, Niranjan Polala, and Siripuri Kiran. 2021. Logistic regression versus xgboost: Machine learning for counterfeit news detection. In *2021 Second International Conference on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE)*, pages 1–6.

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, Copenhagen, Denmark. Association for Computational Linguistics.

Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513–523.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108.

Alex Schackmuth. 2018. *Extremism, fake news and hate: Effects of social media in the post-truth era*. Ph.D. thesis, DePaul University.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to fine-tune bert for text classification? In *Chinese Computational Linguistics*, pages 194–206, Cham. Springer International Publishing.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. *CoRR*, abs/1703.01365.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

William Yang Wang. 2017. "liar, liar pants on fire": A new benchmark dataset for fake news detection. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 422–426, Vancouver, Canada. Association for Computational Linguistics.

Liu Zhuang, Lin Wayne, Shi Ya, and Zhao Jun. 2021. A robustly optimized BERT pre-training approach with post-training. In *Proceedings of the 20th Chinese National Conference on Computational Linguistics*, pages 1218–1227, Huhhot, China. Chinese Information Processing Society of China.

A Document Splitting Algorithm

Algorithm 1 Document Splitting Procedure

$X, Y \leftarrow array(), array()$
$maxlen \leftarrow 512$
$samplelen \leftarrow 0$
for $sentence \in text$ **do**
  $tokens \leftarrow tokenize(sentence)$
  if $samplelength < maxlen$ **then**
    $X.insert(tokens)$
    $samplelen \leftarrow samplelen + len(tokens)$
  else
    $Y.insert(X)$
    $X \leftarrow array(tokens)$
    $samplelen \leftarrow len(tokens)$
  end if
end for
if $samplelen \geq 10$ **then** $Y.insert(X)$
end if

B Word Vector Experiments

Combining TF-IDF and word vectorization techniques like GloVe, FastText, and Word2Vec is a popular approach for improving the performance of various NLP tasks, such as text classification and sentiment analysis. The basic idea behind this approach is to leverage the strengths of both techniques to capture both the importance and semantic meaning of words in a document. Pretrained Word2Vec/GloVe embeddings were used and the FastText model was trained on the training corpus. The resulting combination is then used as an input for classification using logistic regression.

In addition to word-level vectorization techniques, document-level vectorization techniques can also be combined with TF-IDF to further enhance the performance of text classification. Doc2Vec and Universal Sentence Encoder (USE) are examples of such techniques that encode complete sentences and documents into fixed-length vector representation.

Method	Prec.	Recall	F1
TF-IDF	0.75	0.73	0.72
Word2Vec	0.70	0.69	0.68
GloVe	0.69	0.68	0.67
FastText	0.63	0.64	0.63
Doc2Vec	0.64	0.65	0.64
USE	0.77	0.76	0.75

Table 6: Macro-Averaged Results of Word Vectorization Techniques Combined with TF-IDF.

From the results shown in Table 6, the performance of this approach does not yield as good a result as expected. The approach showed similar results for all methods, with the best performance on *reliable* and *satire* and poor performance on *hoax* and *propaganda*. The good performance for *reliable* could be attributed to the use of straightforward language. The good performance for *satire* could be attributed to similarity in writing style and language used with *reliable* headlines. *Satire* headlines are often written in a news-like format and use similar word and sentence structure as *reliable* news, albeit with a humorous or ironic twist.

The poor performance for *propaganda* and *hoax* could be due to the usage of language that is intentionally misleading and ambiguous, making it difficult to generate meaningful word embeddings from the text. Another possible factor is the usage of domain-specific language, such as emotional, sensational and manipulative language that is not commonly found in the corpora used to train the Word2Vec/GloVe models, which we explain in detail in the main body of this report.

C Additional Data Exploration

C.1 Reading Complexity

Further Analysis was conducted on the text using the Flesch Reading Ease Test. This test uses the average length of the sentences and the average number of syllables per word to give a score on a scale up to 121. Normally, scores will fall within a range of 10 to 100. Extremely complex texts can generate a negative score.

From the results shown in Figure 7, *satire* follows most closely to a normal distribution, with more variance in the complexity and reading ease of its sentences between each text compared to the other classes. This difference could be due to the fact that *satire* often employs a range of literary devices, such as irony, sarcasm, and hyperbole to ridicule and criticize individuals, institutions, or the status quo. Thus, a wide range of sentence structures and vocabulary are often used to create a nuanced and multilayered effect, leading to greater variance in sentence complexity.

In contrast, *hoax* and *propaganda* are designed to deceive or manipulate. Hence, they often rely on simple and straightforward language in forming sentences. Similarly, *reliable* text, such as news articles, prioritize clarity and accuracy and thus will use more standardized sentence structures and vocabulary.

C.2 Sentiment Analysis

Sentiment analysis was conducted using the TextBlob library, which provides more insights into the opinions that various authors holds towards the events described within their text. Each text is assigned a polarity score from -1 to 1, where a score of -1 suggests a negative sentiment while a score of 1 reflects a positive sentiment.

From the results shown in Figure 8, all classes follow some form of unimodal distribution, except for propaganda. The propaganda class is observed to have a more positive kurtosis compared to the others, with the mean centering more towards the positive sentiment. This could be a result of the nature of propaganda articles, which are trying to promote an agenda in a positive light. In contrast, a reliable news article will try to present facts and will do so in a neutral manner.

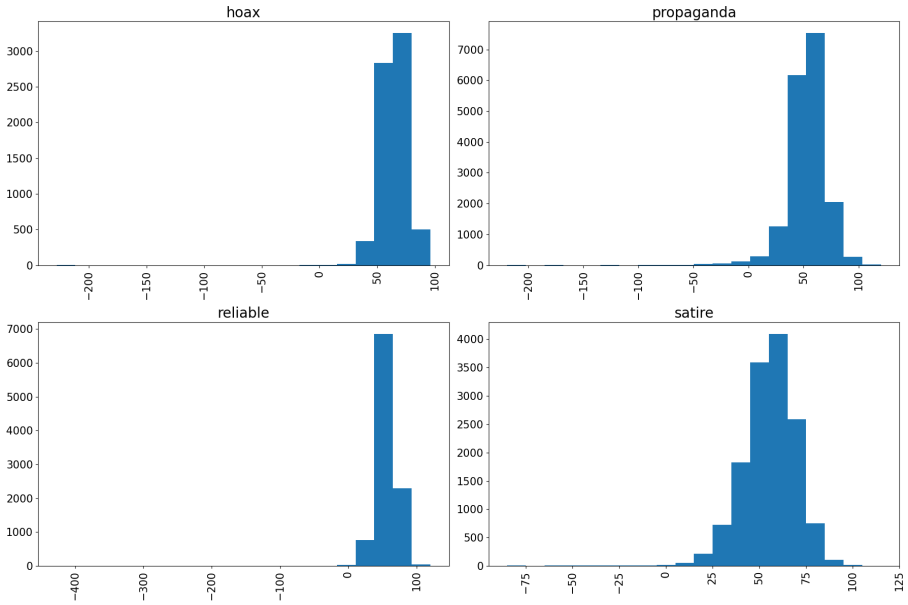

Figure 7: Flesch Reading Score Per Class

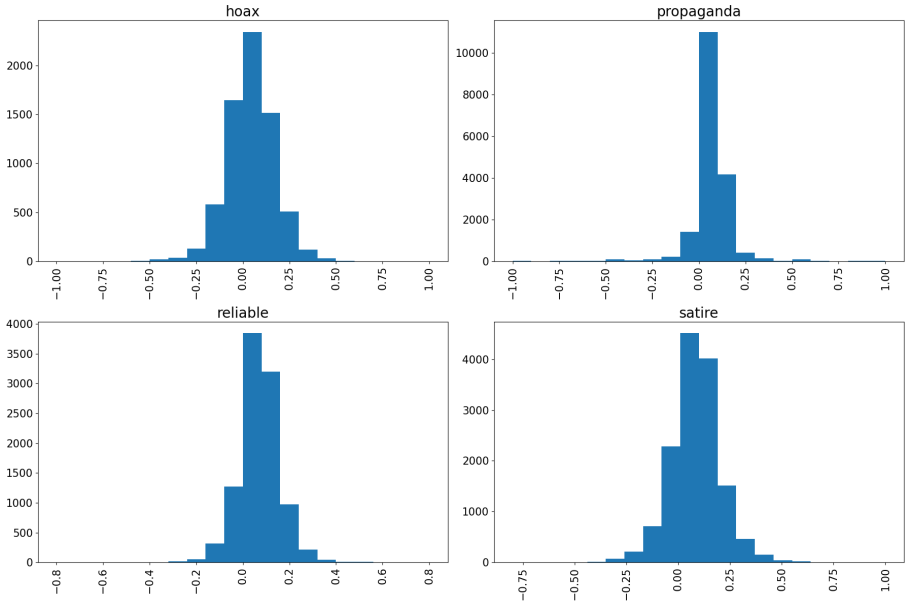

Figure 8: Polarity Score Per Class

D Sample-Level Token Attributions

Text	Real	Predicted
[CLS]A new poll released Wednesday revealed that people rank President Barack Obama as the worst President since the Second World War, and also blame him for starting the Second World War. While the respondents slammed the President for his handling of the economy, Iraq, and a host of other issues, his perceived role as the primary cause of the Second World War was the biggest drag on his numbers . Even more troubling, when compared to the three leaders of the Axis powers during that war , PresidentObama polled at the bottom of the list , finishing far behind Emperor Hiro hito of Japan . Fair or not , the American people hold the President responsible for starting the Second World War , Davis Logs don , a political - science professor at the University of Minnesota , said . If the President hasn t gotten his version of the story out , there s only one person to blame for that : Barack Obama . In other poll results , the most popular President in the survey was Ronald Reagan , widely credited with ending the Second World War . [SEP]	Satire	Hoax
[CLS] Just hours after Senator Ted Cruz (R - Texas) told CNN that he had no choice but to sign up for Obamacare , President Barack Obama signed an executive order making Cruz ineligible for coverage under the Affordable Care Act . Clearly , the hardship of receiving Obamacare was causing Ted a great deal of pain , the President said . This should take care of that . Obama acknowledged that the executive order , which makes Cruz the only American expressly forbidden from signing up for Obamacare , was an extraordinary measure , but added , I felt it was a necessary humanitarian gesture to protect Ted from the law he hates . Even as he signed the order , the President said that he was torn about barring Cruz from coverage , stating , He s definitely someone who would benefit from seeing a doctor . In an official statement released later in the day , Cruz blasted the executive order and accused Obama of distorting his position on Obamacare : I never said I didnt want to have it . I said I didnt want everyone else in the country to have it . [SEP]	Satire	Hoax
[CLS] Chattanooga Marine Hero Welcome d Home By This AMAZING Patriot Turn out ! Local WBS TV is reporting : A police motorcade escorted the body of Lance Cpl . Skip Wells from Hartsfield - Jackson International Airport along Interstate 75 to Cobb County , where his funeral will be held. Heather Henderson brought her two boys to join the hundreds of people who lined up on the Canton Connector Bridge Thursday. I m from Cleveland , Tennessee , which is close to Chattanooga , so the entire incident was close to home, Henderson told Channel 2s Ross Cavitt. [SEP]	Hoax	Reliable

Figure 9: Sample Error Analysis. Bright green highlighting indicates a higher positive attribution of the predicted class to that word, dull green indicates lower positive attribution, while orange indicates negative attribution.

E Class-Level Token Attributions

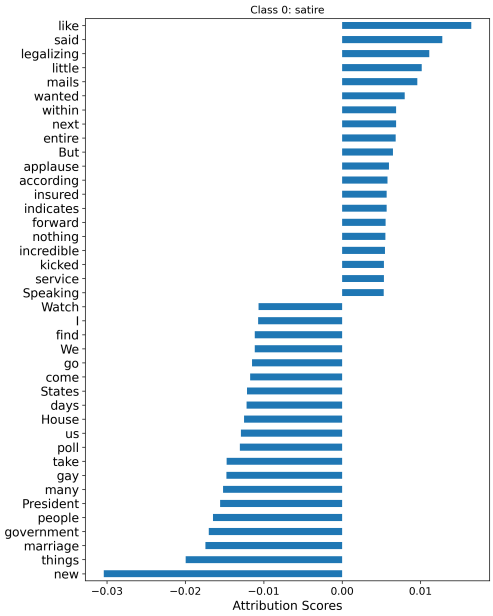

Figure 10: Satire Top 20 Words for Negative and Positive Attribution Scores.

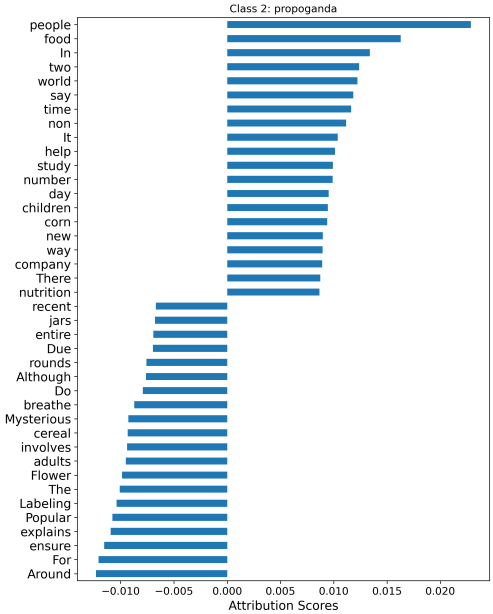

Figure 11: Propaganda Top 20 Words for Negative and Positive Attribution Scores.

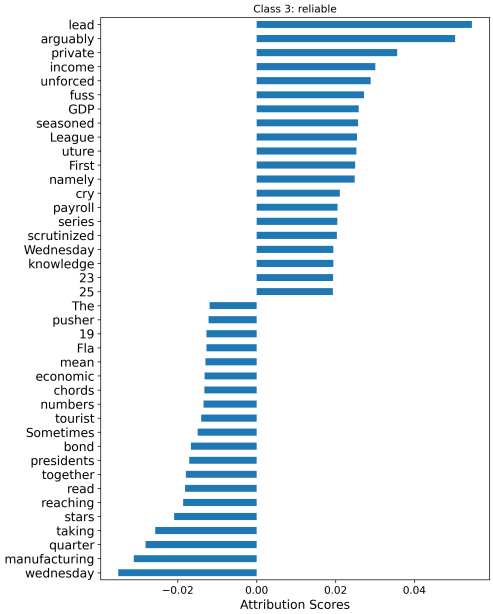

Figure 12: Reliable Top 20 Words for Negative and Positive Attribution Scores.