Article-Level Fake News Detection With BERT-Derived Natural Language Processing Architectures

Stacy Irwin, Kenneth Pong
University of California, Berkeley School of Information
stacy.irwin@berkeley.edu, kpong@berkeley.edu

Abstract

In this paper, we evaluate the use of Transformer-based models in fake news detection. Specifically, we evaluate the DistilBERT, RoBERTa, and Longformer models. A drawback of using Transformer models for this purpose is that many news articles are longer than the sequence length limitation of typical Transformer models. Researchers have evaluated several techniques for classifying texts with long sequences, including truncation, hierarchical classification, and modification of attention heads to reduce time complexity. We evaluate using modified attention heads (Longformer) and combining truncation with prefixing the article title to the beginning of the article sequence. Our results indicate that use of article titles as a proxy summary is beneficial. On the other hand, the additional computational complexity of models that accept longer sequences makes the models difficult to train and appears to offset the benefits of evaluating a longer sequence. Based on this, our final model consists of two stages - the first for article classification, and the second for veracity classification. Using this combined approach, we realize slight improvements over a DistilBERT-based baseline model combining data from multiple sources.

Introduction

The rise of internet-based social media has made fake news easier and cheaper to broadly disseminate. Our objective is to create a product that can be trained on readily available data to offload some of the work of fake news detection onto algorithms. To accomplish this, we make use of variants of BERT-based language models to train a model against the FakeNewsNet corpus, which consists of articles from PolitiFact (PF) and GossipCop (GC).

To establish a baseline, we first fine-tuned a DistilBERT model to distinguish between fake and real news articles (Sanh et al., 2019). DistilBERT's architecture is similar to BERT's, but with half the layers. DistilBERT was trained using a technique called knowledge distillation, where training data was passed through both BERT and DistilBERT models and loss was calculated using the resulting probabilities from both BERT and DistilBERT.

We next attempted to train a Longformer model (Beltagy, Peters and Cohan, 2020) on the dataset. Due to DistilBERT's and BERT's sequence length limit, our model truncated articles longer than 512 tokens. Longformer is able to evaluate sequences of up to 4096 tokens. Longformer achieves this by using sparse attention heads -- most Longformer attention heads evaluate only a portion of the sequence, resulting in attention heads that can be processed in linear time instead of the quadratic time required for full attention heads. We discovered that Longformer was difficult to train and whatever performance improvement we could extract did not justify the additional tuning time and resources required.

To determine if the challenges we experienced when attempting to train Longformer were due to the longer sequences or some other difference between Longformer's and DistilBERT's architecture, we repeated the tuning process on a RoBERTa model (Liu et al., 2019), the transformer model upon which Longformer is based. Longformer was developed by replacing the quadratic attention heads on the RoBERTa model. RoBERTa was trained on a larger corpus than BERT, but was trained only on the masked language modeling task, not on next sentence prediction. Because many of the articles would be truncated, we prepended the article tokens with the article's title, separating the two sequences with a separator token. We had more success tuning the RoBERTa model than Longformer, however the RoBerta model performed poorly on datasets containing both GC and PF data. Our final model architecture used three transformer models: a DistilBERT model that categorized articles as either political or celebrity news; and two RoBERTa models, one trained on the GC dataset and the other trained on PF data. This model achieved a slight improvement over our baseline, as indicated by the F1 score.

Although a number of papers have employed the base BERT model for fake news detection, the use of the RoBERTa architecture for this task has limited precedent in the literature, with only one known paper by Slovikovskaya (2019) having employed it in this capacity, albeit on the FNC-1 dataset[1].

Literature Review/Background

Fake News Detection Tasks - Overview

Broadly speaking, there are several distinct approaches to fake news detection under research, as enumerated below by Zhou and Zafarani (2020):

1) Knowledge-based fake news detection, which focuses on knowledge extraction,
2) Style-based fake news detection, which looks at intentions and writing style,
3) Propagation-based fake news detection, which looks less at the content and more at the pattern of spread across a network, and
4) Source-based fake news detection, which evaluates the probability of a piece of content given the historical credibility of the publisher.

Our approach uses style-based fake news detection.

Fake News Detection Tasks - Algorithms

Presently, the state of the art consists of Bidirectional Encoder Representations from Transformers (BERT), as proposed by Devlin et al. (2019). BERT makes use of the encoder mechanism from Transformers, and consists of a pre-trained model with an additional output layer for fine-tuning the model for different tasks (e.g. sequence classification, question answering, multiple choice answering).

As for the algorithms used in the task of fake news detection, BERT in general has some precedent in being used for that task. For instance, Singhal et al. (2019) also apply BERT to fake news detection, albeit to Twitter and Weibo, while Nakamura et al. (2019) use BERT on data extracted from Reddit. Jwa et al. (2019) also employed BERT in conjunction with Cable News Network (CNN) and Daily Mail corpuses during pre-training, and evaluated against the FNC-1 dataset. In terms of adjacent tasks, BERT has also been used to detect spreaders of fake news (Baruah et al., 2019).

Fake News Detection Tasks - Datasets

Having chosen the proper algorithm for our purposes, we turned to the availability of datasets fit for the identified purpose. Our understanding of the dataset topology in this problem space largely derives from the work of Oshikawa et al. (2018), who compiled a survey on existing datasets that could be used for fake news detection. As per his work, we considered four different datasets:

1) **CREDBANK** (Mitra and Gilbert, 2015), a repository consisting of 60 million tweets spanning 1,049 events. We eliminated this dataset because individual tweets were not categorized as fake or real.
2) **FEVER** (Thorne et al., 2018), a three-class repository of 185,445 claims. We eliminated this dataset because, as per Schuster et al. (2019), the fake examples in FEVER tend to include idiosyncrasies that affect FEVER-trained models and hamper generalizability.
3) **LIAR** (Wang, 2017), a repository of 12,836 claims made on PolitiFact along with corresponding six-class veracity ratings. We eliminated this dataset because the claim strings are short, hindering our ability to classify the records
4) **FakeNewsNet** (Shu et al., 2019), a 23,921 article repository consisting of full-length PF and GC articles with a binary (true/fake) classification. We chose to focus our analysis on FakeNewsNet.

Although we decided to focus on FakeNewsNet (FNN), this dataset does have its own limitations - most notably, the domains from which the training data are drawn are restricted to fake news from celebrity and political spaces. Also, exploratory data analysis (see corresponding section) shows that this repository is significantly biased in composition towards GC-based examples over PF, and that a significant class imbalance exists in GC articles.

Methodology

For all models mentioned in this paper, we used the HuggingFace implementation[2] and fine-tuned the

[1] https://github.com/FakeNewsChallenge/fnc-1

[2] https://huggingface.com

models using PyTorch. The datasets were split into training (60%), validation (20%) and test (20%) subsets. The validation dataset was used for hyperparameter tuning while the test dataset was reserved for final evaluation. We used a cross entropy loss function and we chose the AdamW optimizer, created by Loshchilov and Hutter (2019), over the standard Adam optimizer because it yields lower training losses and generalizes better than Adam. Unless otherwise stated, all models were trained for ten epochs.

Metric Choice

The dataset's imbalance makes it easy to get a high accuracy by predicting the majority (real) class. Therefore, we primarily use recall, precision, and F1 scores to compare models. For purposes of calculating precision, recall, and F1, detection of a fake article is considered a positive result. The choice of these metrics is consistent with other papers employing BERT for fake news-based tasks, such as Levi et al. (2019) and Hou and Chen (2019).

Baseline Model Architecture Choice

We chose a DistilBERT model for the baseline due to its compact size and ease of training. While there was considered to be a benefit to using the base BERT model, DistilBERT reduces the model size by 40% compared to BERT and is 60% faster (Sanh et al., 2019). For the initial run, we used the combined FakeNewsNet data (GC and PF concatenated into a single dataset) as an input to train a DistilBERT model for sequence classification.

Post-Baseline Model Architecture Choice

Initially, we had hoped to make use of the Longformer variant of the BERT model to take advantage of the 4,096 token limit. As the EDA demonstrated that the standard token limit is only sufficient for about two-thirds of GC articles and one half of PF articles, it was believed that the sequence length would improve performance over the baseline. By contrast, Longformer's 4,096 token limit is able to fully capture 98% of all GC articles, and 89% of all PF articles. However, results from the PF-only run of Longformer were merely comparable to the equivalent model trained on DistilBERT, with a ten-epoch Longformer model trained on the validation dataset yielding precision of 0.921, recall of 0.817, and an F1 of 0.865, a little lower than the equivalent DistilBERT model's results of 0.929 precision, 0.942 recall, and 0.935 F1. This indicated that the longer sequence length did not necessarily result in better predictions. We believe this may be attributable in part to the standard inverted pyramid structure of articles, which dictates that the most important information is placed at the beginning of the article, with less important information following it.

When we attempted to train Longformer on the full 60% training GC data, the model failed to learn after five epochs, each of which took several hours. The failure to learn was evidenced by the output logits, which were nearly identical for all evaluation cases and produced the same prediction for all validation article inputs. Numerous experiments were conducted with the parameters by testing different loss weights, learning rates, and values of epsilon (a parameter for adjusting weight update sizes for stability), none of which successfully allowed Longformer to learn on the full GC dataset after five epochs.

We suspect that the difficulty was due to the larger data set associated with GC, coupled with the memory-intensive nature of Longformer necessitating the use of small batch sizes (<=8), yielding gradients that would take longer to converge. These issues are likely compounded by the similarity of common words in real and fake GC articles. Given more time and computational resources, we would be able to test our hypothesis regarding the technical causes - however, lacking evidence of better performance from the PF model results and in the interests of time, we decided to switch to the RoBERTa model.

Topic Choosing Neural Network

Due to the difficulty we experienced with training the RoBERTa model on a combined GC and PF dataset, we decided to first use a DistilBERT model to classify input articles as either political (PF) or celebrity (GC) news. After this, the articles would be sent to either a PF-trained or GC-trained RoBERTa model, depending on the classification assigned by the DistilBERT topic chooser model. We assumed that a model specializing in fake news from one domain would perform better than a model that combined data from both domains.

The DistilBERT topic chooser model performed satisfactorily, achieving a precision of 0.826, recall of 0.832, and an F1 score of 0.829 after five epochs. Performance on the development data began to degrade after five epochs, perhaps due to overfitting. The results indicated that an approach where the full dataset is submitted to two sequential transformer models, where the first model splits the data into political and gossip categories and the second model detects fake articles, could be successful.

Preprocessing

Data Retrieval

Prior to training any models, we needed to acquire a full copy of the dataset. As mentioned, we used the FakeNewsNet dataset, which consists of articles that were rated by GC and PF's sites. Because 90% of GC articles were rated as fake, the developers supplemented the dataset with articles from *E! Online*, all of which are considered to be true.

The developers of FakeNewsNet did not provide the text of the articles directly. Instead, they provided a dataset of article URLs and Python modules from their GitHub repository[3] to download the articles from the source websites. Many of the articles were no longer available at the reported URL, which caused the text of some articles to consist of an error message, leading us to filter out these error messages by eliminating all articles with fewer than 500 characters in the article body. After this removal, we were left with 664 PF and 17,647 GC articles.

For the RoBERTa and topic chooser models, our training data consisted of both the article title and body text. We hypothesized that the title would serve as a proxy article summary and could compensate for truncating the article to meet the limits on sequence length. Many of the titles contained the article's source (e.g., New York Times, PBS News Hour) in addition to a descriptive headline. We decided to strip the sources from the title because those source names can easily be faked. As well, we prefer for the model to concentrate on the article's style and content. Removing the sources was straightforward because the sources were separated from the rest of the title with a dash or pipe. The title was separated from the body text with the appropriate separator token for the particular model in use.

[3] https://github.com/KaiDMML/FakeNewsNet

Post-Baseline Fine Tuning

We used DistilBERT models to evaluate the impact of including the article title. The result was mixed, with a slight increase in precision at the expense of recall. Still, given the importance of the title, it was decided to retain this change.

Next we experimented with different weighting schemes. Due to the imbalance of the data set and given that predictions of fake articles are more important, we experimented with increasing the weights on losses for fake articles. To test these out, we tried four different sets of weights - unweighted, [1,2], [1,4], and [1,8]. Based on these tests, we observed that [1,4] yielded the best performance. The weights significantly increased recall over the first baseline, albeit at the expense of precision. Nevertheless, we believed it was more important to weigh the false predictions more heavily. That, coupled with the overall improvement in performance in terms of F1, led us to start using loss weights for subsequent models.

Results and Discussion

Exploratory Data Analysis

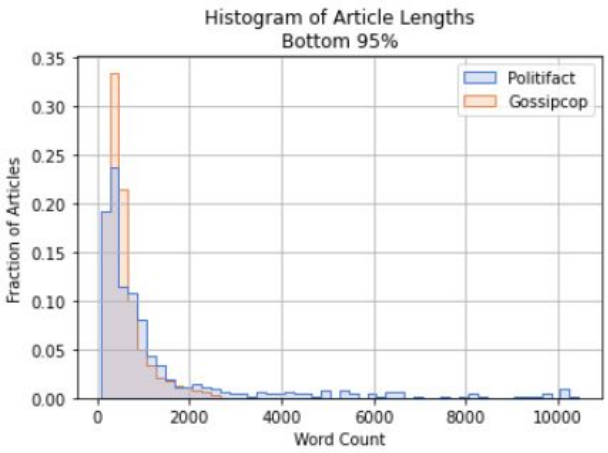
Table 1 shows that the GC portion of the dataset is imbalanced with more real articles than fake, whereas the PF portion is more evenly split.

	Real	Fake	Total
Politifact (PF)	560 (58.8%)	392 (41.2%)	952
Gossipcop (GC)	15,373 (75.8%)	4,891 (24.2%)	20,264
Total	15,933	5,283	21,216

Table 1: Number of Articles in FNN Dataset

To assess the potential consequences of BERT and RoBERTa's token limits, we assessed the word count distribution of the GC/PF datasets (see Tables A1 and A2 in Appendix B). What we found was that a 512 token limit fully captures only 67% of GC articles, and 48% of PF articles. The article length distributions have long tails, with the longest articles being several

times longer than even the 99th percentile length values in all cases except for real PF articles.



Graph 1:Article Length Frequency Distribution

In terms of the length of the articles between the two sources, PF's articles were nearly three times as long, averaging 1,810 words compared to GC's articles, which averaged 670 words. In both cases, the average length of the articles exceeds the token length limit of conventional BERT models. In terms of variations in average word length between fake and real articles, Gossipcop's article length did not vary much, at 684 words for real articles and 603 for fake articles. However, PF's data showed significant disparity with fake articles averaging 475 words and true articles averaging a whopping 2,763 words.

As for the frequency of specific words by data source, there is considerable overlap between the set of most common words for fake and real GC articles (see Graphs A1 through A4 in Appendix B). Meanwhile, PF articles show significantly different common word frequency profiles, with real PF articles most frequently containing "going", "think" and "people". Of particular interest is the fact that the most common words in fake PF articles were "Trump" and "said". This suggests that most PFarticles flagged as fake are heavily weighted towards assessments of Donald Trump's statements and actions, which hints at limitations in the model's ability to generalize on political articles not centered on American politics.

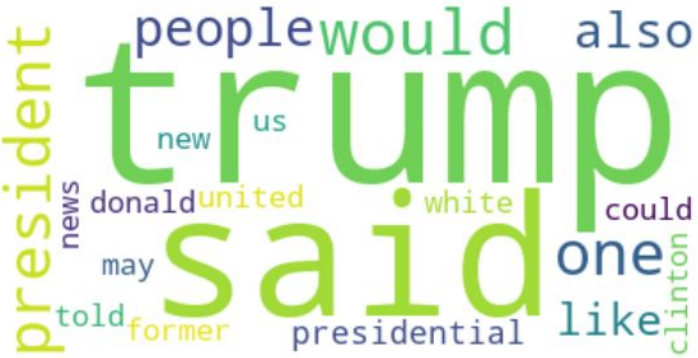


Figure 1: Wordcloud of Most Common Words in Fake PolitiFact articles

Baseline Models

	Precision	Recall	F1
1. Distilbert	0.7211	0.6226	0.6682
2. Distilbert, titles	0.7753	0.6344	0.6978
3. DistilBERT, titles and weights	0.6507	0.7172	0.6824

Table 2: Performance of DistilBERT models on test data

For the first baseline model, we used DistilBERT, trained over ten epochs without custom losses or titles in the training sequence. The baseline model achieved an F1 score of 0.684 (see Table 2, Row 1). Inclusion of titles improved all metrics and brought the F1 score up to 0.6978 (see Table 3, Row 2). Finally, the incorporation of the aforementioned loss weights reduced the final baseline model's F1 score to 0.6824 (see Table 2, Row 3). However this model did have the highest recall of three DistilBERT models. This suggests that the choice of weighting may depend on the intended application and the relative cost of a false positive compared to a false negative.We did verify that the DistilBERT model does not score well when asked to perform inferences on subject domains not included in the training data. To test this, the GC and PF data were kept separate, then a DistilBERT model using only GC data was trained and scored its performance in validating against PF data. The results suffered significantly compared to the combined GC/PF model, with a resulting precision of 0.45, recall of 0.51, and F1 of 0.48. This outcome was expected, as it is intuitive that a model trained on one domain (celebrity gossip) does not necessarily generalize well to another domain (political discussions). This shows how the efficacy of models produced as part of this process may be restricted to the domains from which they are trained.

Post-Baseline Models

	Precision	Recall	F1
1. RoBERTa, GC (dev data)	0.7454	0.6624	0.7015
2. RoBERTa, PF (dev data)	0.9221	1.000	0.9595
3. RoBERTa, GC and PF, data sorted by DistilBERT topic chooser model	0.7338	0.6817	0.7068

Table 3: Performance of RoBERTa models on test data

While we experienced difficulty training a RoBERTa model on a combined GC and PF dataset, we found that RoBERTa models were trainable if the datasets were kept separate. We achieved an F1 score of 0.702 on GC data and 0.960 on PF data (Table 4, Rows 1 and 2).The greater accuracy of models trained on PF over GC was a recurring theme throughout this project, and may be due in part to the significantly different common word profiles observed or due to the difference in average article length for real and fake articles.

Finally, having selected our models and parameters, we retrieved the pre-reserved test articles in order to evaluate the performance of the topic chooser model on unseen data. In the end, the two-stage neural network slightly edged out the best baseline model (with titles and custom loss weights) in terms of F1, with a score of 0.707 (see Table 3, Row 3) compared to the best DistilBERT model's score of 0.698. Interestingly, the baseline model performed better in recall (0.717), but at the cost of significantly more false positives (358 real articles flagged as fake for the baseline compared to 230 for the RoBERTa model).

Error Analysis

Of the errors that were observed in the final topic chooser model run on test data, we counted a total of 526 erroneous predictions, 512 of which belonged to GC data. Although the PF errors were too few to analyze in any meaningful fashion, the median distance between real and fake article probabilities reported for the GC errors was 0.78 (see Graphs A5 and A6 in Appendix B). This indicates that the large majority of erroneous predictions were the result of highly confident, albeit incorrect predictions. This appears to be at least partially the result of the similar word profiles between fake and real GC articles, which suggests that further gains in predictive accuracy may be contingent on a language model's ability to extract salient features given similar interclass vocabulary composition.

Conclusion

In exploring the state of the art in language models, we have tapped DistilBERT and RoBERTa to produce models capable of detecting fake news based on articles from GossipCop and PolitiFact data. We were unable to demonstrate that analyzing longer text sequences with the Longformer model resulted in better system performance, but it is possible that better results could be obtained with more powerful hardware that supports larger batch sizes during training. We also discovered that RoBERTa models made little progress during training compared to the simpler DistilBERT model on datasets with both celebrity and political news. A contributing factor could be that the RoBERTa models required more memory and were limited to a smaller batch size during fine-tuning (16 vs 32 for DistilBERT). Our final model consisted of a two-step neural network with a DistilBERT model categorizing the article by topic and a topic-specific RoBERTa model predicting whether the article was fake or real. This architecture slightly exceeded the baseline in terms of F1 when evaluated on the previously-unseen test dataset. These results were achieved despite there being two possibilities for error, first during topic selection and then during evaluation of veracity.

References

Baruah, Arup, Kaushik Amar Das, Ferdous Ahmed, and Kuntal Dey (2020). "Automatic Detection of Fake News Spreaders Using BERT". http://ceur-ws.org/Vol-2696/paper_237.pdf.

Beltagy, Iz, Matthew E. Peters, and Arman Cohan (2020). "Longformer: The Long-Document Transformer". https://arxiv.org/pdf/2004.05150.pdf.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee and Kristina Toutanova (2019). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". https://arxiv.org/abs/1810.04805.

Hou, Wenjun, and Ying Chen (2019). "Sentence-Level Propaganda Detection Using BERT

withContext-Dependent Input Pairs". https://www.aclweb.org/anthology/D19-5010/.

Jwa, Heejung, Dongsuk Oh, Kinam Park, Jang Mook Kang and Heuiseok Lim (2019). "exBAKE: Automatic Fake News Detection ModelBased on Bidirectional Encoder Representations from Transformers (BERT)". https://www.mdpi.com/2076-3417/9/19/4062/htm.

Levi, Or, Pedram Hosseini, Mona, Diab, and David A. Broniatowski (2019). "Identifying Nuances in Fake News vs. Satire:Using Semantic and Linguistic Cues". https://www.aclweb.org/anthology/D19-5004/.

Liu, Yinhan, 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". https://arxiv.org/abs/1907.11692.

Loshchilov, Ilya, and Frank Hutter (2019). "Decoupled Weight Decay Regularization". https://arxiv.org/abs/1711.05101.

Mitra, Tanushee and Eric Gilbert (2015). "CREDBANK: A Large-Scale Social Media Corpus With Associated Credibility Annotations". https://www.aaai.org/ocs/index.php/ICWSM/ICWSM15/paper/view/10582/10509.

Nakamura, Kai, Sharon Levy, and William Yang Wang (2019). "r/Fakeddit:A New Multimodal Benchmark Dataset forFine-grained Fake News Detection". https://arxiv.org/abs/1911.03854.

Oshikawa, Ray, Jing Qian and William Yang Wang (2018). "A Survey on Natural Language Processing for Fake News Detection". https://arxiv.org/pdf/1811.00770.pdf.

Sanh, Victor, Lysandre Debut, Julien Chaumond and Thomas Wolf (2019). "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter". https://arxiv.org/abs/1910.01108.

Schuster, Tal, Darsh J Shah, Yun Jie Serene Yeo, Yun Jie Serene Yeo and Enrico Santus (2019). "Towards Debiasing Fact Verification Models". https://arxiv.org/abs/1908.05267.

Singhal, Shivangi, Rajiv Ratn Shah, Tanmoy Chakraborty, Ponnurangam Kumaraguru and Shin'ichi Satoh (2019). "SpotFake: A Multi-modal Framework for Fake News Detection". https://ieeexplore.ieee.org/document/8919302.

Shu, Kai, Deepak Mahudeswaran, Suhang Wang, Dongwon Lee and Huan Liu (2019). "FakeNewsNet: A Data Repository with News Content, Social Context and Spatialtemporal Information for Studying Fake News on Social Media". https://arxiv.org/abs/1809.01286.

Slovikovskaya, Valeriya (2019). "Transfer Learning from Transformers to Fake News Challenge Stance Detection (FNC-1) Task". https://arxiv.org/abs/1910.14353.

Thorne, James, Andreas Vlachos, Christos Christodoulopoulos and Arpit Mittal (2018). "FEVER: a large-scale dataset for Fact Extraction and VERification". https://arxiv.org/abs/1803.05355.

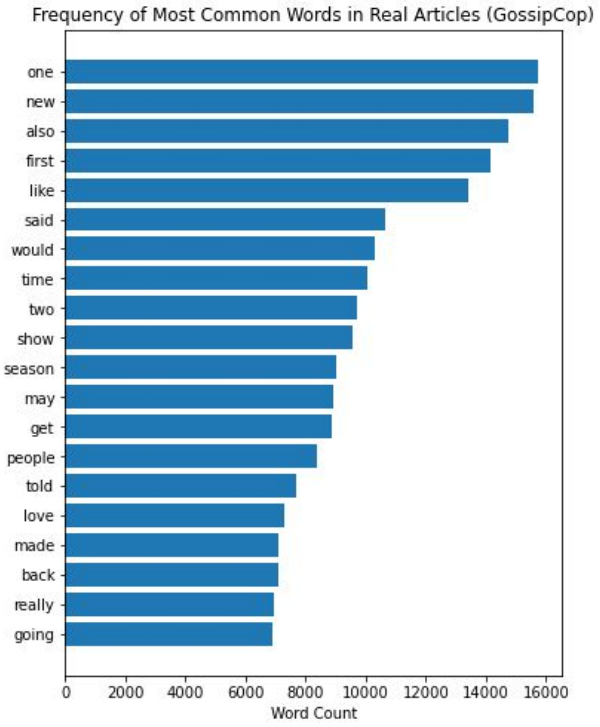
Wang, William Yang (2017). ""Liar, Liar Pants on Fire": A New Benchmark Dataset for Fake News Detection". https://www.aclweb.org/anthology/P17-2067.pdf.

Zhou, Xinyi, and Reza Zafarani (2020). "A Survey of Fake News:Fundamental Theories, Detection Methods, and Opportunities". https://arxiv.org/pdf/1812.00315.pdf.

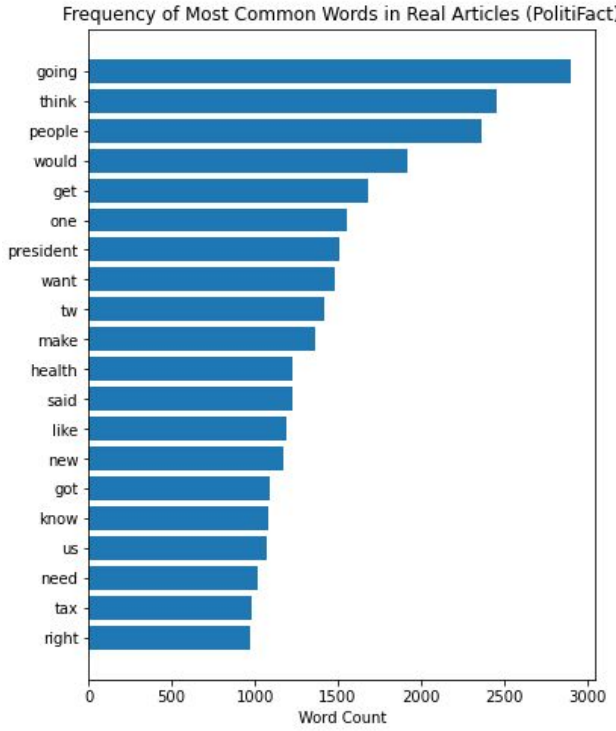
Appendix A - Model Results and Scores

Model Name	Dataset	Model	Task	Titles?	Weights	Epochs	Dataset	Precision	Recall	F1
Baseline	Both	distilbert-base-uncased	Fake / Real	No	No	10	Validation	0.738	0.639	0.685
Baseline with Titles	Both	distilbert-base-uncased	Fake / Real	Yes	No	10	Validation	0.759	0.613	0.678
Baseline with Titles, Weights	Both	distilbert-base-uncased	Fake / Real	Yes	Yes	10	Validation	0.664	0.755	0.707
RoBERTa GC	GossipCop	roberta-base	Fake / Real	Yes	Yes	10	Validation	0.745	0.662	0.701
RoBERTa PF	PolitiFact	roberta-base	Fake / Real	Yes	Yes	10	Validation	0.922	1.000	0.959
Topic Chooser	Both	distilbert-base-uncased	Political / Gossip	Yes	Yes	5	Validation	0.826	0.832	0.829
Longformer PF	PolitiFact	longformer-base-4096	Fake / Real	Yes	Yes	10	Validation	0.905	0.851	0.851
DistilBERT PF	PolitiFact	distilbert-base-uncased	Fake / Real	Yes	Yes	10	Validation	0.929	0.942	0.935
LongFormer GC	GossipCop	longformer-base-4096	Fake / Real	Yes	Yes	2	Validation	0.000	0.000	0.000
Baseline (Test Data)	Both	distilbert-base-uncased	Fake / Real	No	No	10	Test	0.721	0.623	0.668
Baseline with Titles (Test Data)	Both	distilbert-base-uncased	Fake / Real	Yes	No	10	Test	0.775	0.634	0.698
Baseline with Titles, Weights (Test Data)	Both	distilbert-base-uncased	Fake / Real	Yes	Yes	10	Test	0.651	0.717	0.682
RoBERTa GC for Topic Chooser	GossipCop (sorted)	roberta-base	Fake / Real	Yes	Yes	10	Test	0.727	0.667	0.696
RoBERTa PF for Topic Chooser	PolitiFact (sorted)	roberta-base	Fake / Real	Yes	Yes	10	Test	0.820	0.926	0.870
RoBERTa using Topic Chooser Inputs	Both (sorted)	roberta-base	Fake / Real	Yes	Yes	10	Test	0.734	0.682	0.707
LongFormer PF (Test Data)	PolitiFact	longformer-base-4096	Fake / Real	Yes	Yes	10	Test	0.915	0.982	0.947
RoBERTa PF (Test Data)	PolitiFact	roberta-base	Fake / Real	Yes	Yes	10	Test	0.887	1.000	0.940

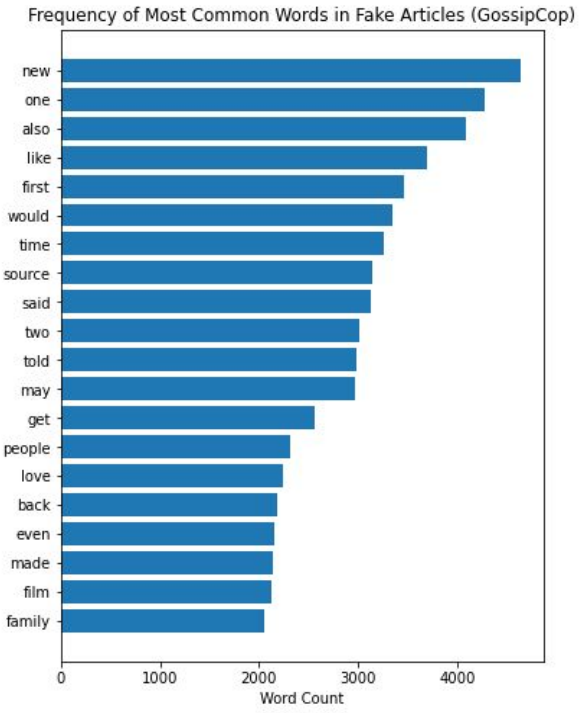
Appendix B - Word Frequency and Misclassification Graphs



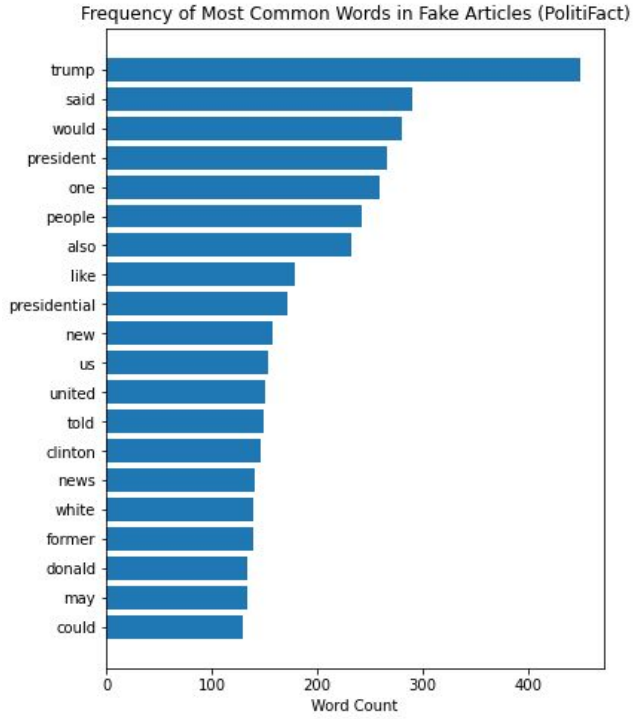
Graph A1: Most Common Words in Real GC Articles



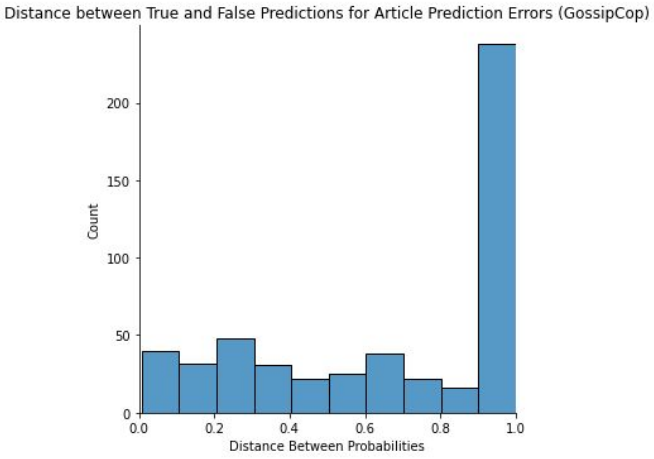
Graph A3: Most Common Words in Real PF Articles



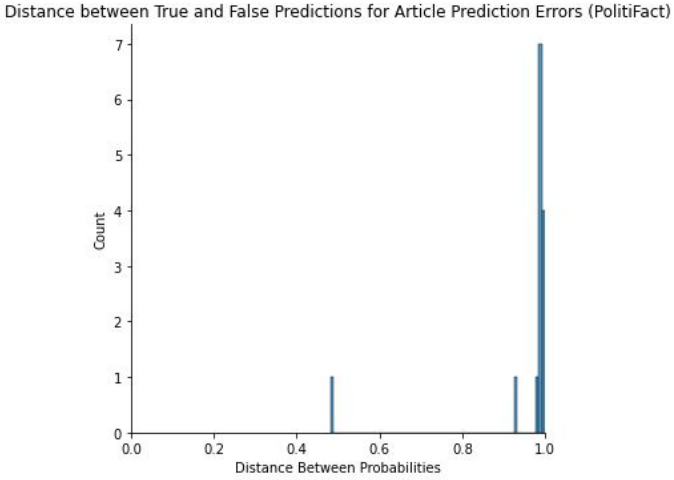
Graph A2: Most Common Words in Fake GC Articles



Graph A4: Most Common Words in Fake PF Articles



Graph A5: Distance Between Class Predictions for Misclassifications (GossipCop)



Graph A6: Distance Between Class Predictions for Misclassifications (PolitiFact)

	word_count								
	count	mean	std	min	50%	90%	95%	99%	max
dataset									
gossipcop	14110	664	1081	48	385	1257	2106	4997	17794
politifact	538	1678	3153	66	537	4666	8659	17061	18044

Table A1: Descriptive Statistics by Data Source

		word_count								
		count	mean	std	min	50%	90%	95%	99%	max
dataset	label									
gossipcop_train	fake	3459	603	920	54	380	996	1824	4896	14907
	real	10652	684	1127	48	388	1316	2175	5115	17794
politifact_train	fake	255	475	544	66	325	858	1095	2808	5555
	real	283	2763	4022	79	909	8613	9892	17345	18044

Table A2: Descriptive Statistics by Data Source and Veracity Label