FNC-1 Challenge: Fake News Detection with LSTM's and MLM's

Aleksander Ficek, Parth Shah

University of Waterlooo
acficek@uwaterloo.ca, phshah@uwaterloo.ca

Abstract

In this report we study ways in which recent machine learning techniques can be used to tackle the Fake News Challenge (FNC) competition, which presents an emerging problem of detecting false information in articles. Using the FNC-1 dataset we first train a gradient boosting classifier as a baseline model and attain a relative FNC score of 8761.25. Next, we implement multiple deep learning approaches to surpass this score starting with a Bi-directional LSTM. Upon creating the LSTM network, modifications were introduced to the model architecture as an unsuccessful attempt to improve performance such as using BERT as opposed to GloVe embeddings and introducing 1-dimension convolutional filters. We then fine-tune the transformer-based masked language models of BERT and RoBERTa for the fake news classification task. This results in a significantly higher FNC score of 10374.25 using RoBERTa. Finally, we introduce a data augmentation technique allowed within competition rules that involves cropping body texts in training data to make the task more challenging for the MLM's. 1.06% improvement over the non-data augmented approach. We preform analysis on this optimal model and provide suggestions for future exploration[1].

1 Introduction

Fake news is a prevalent problem in our current day and age. There is a plethora of information available online, with the ability for any user to upload content for mass viewing. This generates the potential for misinformed transmission of data.

In addition to this, with the rapid need for time efficiency, it has become habitual to simply scan through the face-value of information online. This creates a great deal of problems as headlines of news articles don't always correlate to the content that is being presented. Thus, readers are subject to a bias based on their instinctive intuition retained from the headlines.

The Fake News Challenge (FNC) is an organized competition that aims to tackle the issue of hoaxes and deliberate misinformation in news stories (Challenge n.d.). The intended goal is for competitors to design and optimize the detection of fake news using machine learning, natural language processing and artificial intelligence.

Two sets of texts are compared in order to perform stance detection. The stances are estimated based on a body text from a news article relative to a headline (Challenge n.d.). The output of the detection can suggest whether the body text agrees, disagrees, discusses, or poses to be unrelated to the associated headline. The competition provides the train and test datasets to be utilized for supervised learning. The scoring metric for the competition as evaluated on the test dataset is displayed in Figure 1 below.

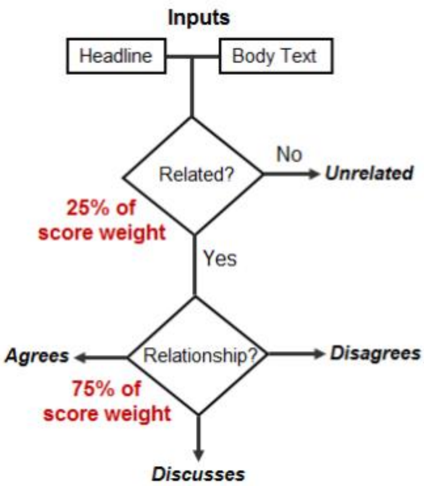

Figure 1: FNC Scoring flowchart (Challenge n.d.)

[1] Code can be found at https://github.com/ParthShahMechatronics/fnc-1-baseline

To take on this task, various natural language processing techniques were explored. An initial baseline model was provided that applied a gradient boosting classifier to generate stances. Neural network models were explored with a focus on LSTM based models and their performance was ranked according to the FNC scores calculated by the challenge. Through further investigation, it was found that transformer-based models had remarkable performance for the FNC challenge. Thus, the BERT and RoBERTa models were trained with modifications to the pre-processing of the dataset in order to obtain competitive results.

The contributions of this paper are as follows:

- We train and tune a Bi-LSTM architecture on the FNC-1 dataset
- We suggest modifications to the model that succeeded and failed such as changing pretrained word embeddings, number of layers and introducing 1D convolutional layers to the Bi-LSTM model.
- We train and tune a Masked Language Model (MLM) architecture on the FNC-1 dataset.
- We propose a new data augmentation technique permitted within the challenge's constraints to improve MLM performance.

2 Background

Fake news detection is a cumbersome task that requires various algorithms to be able to successfully classify misinformed texts. The application of deep learning methods is able to handle such highly computational tasks. The models to ingest these large datasets are built on neural networks, which encompass inputs that map to outputs through a series of interconnected layers. Natural Language Processing (NLP) combines the foundational concepts of these deep learning models to linguistics and can be utilized for text classification. Several text classification models have been developed to combat the identification of fake news.

2.1.1 Recurrent Neural Networks

Recurrent Neural Networks (RNNs) are a popular method that process sequential data for text analytics. In RNNs the input of the current step is fed from the output of the previous step (Pal 2021). Long Short-Term Memory (LSTM) are an artificial RNN that can learn long-term dependencies. This suggests that the model is able to preserve the memory of the words surrounding it to provide useful context using a set of additional gates. Previous work demonstrates that a Bi-directional LSTM model that simultaneously steps through the input sequence in both directions can achieve high performance in detecting fake news after training on two publicly obtained datasets (Bahad, Saxena and Kamal 2020).

2.1.2 Transformers

Transformers in NLP are unprecedented architectures that are able to solve sequence to sequence tasks similar to RNNs (Magnone 2022). However, these architectures harness self-attention mechanisms, as seen in Figure 2 below, to be able to focus on inputs with higher significance (Joshi 2019). This helps increase the range of the data the model can look at with the ability to correlate different positions of an input sequence. Transformers gain an advantage over RNNs as they can expand their range of scope using attention and they encode data in parallel enhancing GPU performance to speed up the time required for training (Magnone 2022).

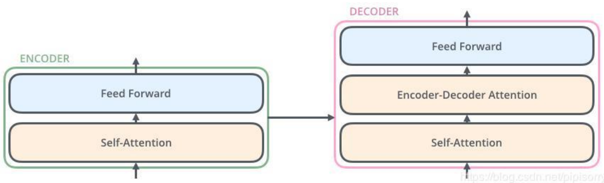

Figure 2: Summary of transformer model architecture (ProgrammerSought n.d.)

One of the widely known transformer-based architecture is called BERT (Bidirectional Encoder Representations from Transformers). BERT is built to be able to train on deep bi-directional representations such that it can compute unlabeled contexts on the left and right side of each token during training (Vidhya 2020). BERT poses great benefits for text classification as it has been pre-trained over an intensively large corpus making it more informed on the language structure. The two objectives of pre-training BERT are masked language modelling and next sentence prediction. Masked language modeling is performed by masking 15% of the tokens and then feeding in the word sequences to the model (Horev 2018). The model is then used to predict the original values of the masked words based on the context of the sequence. Next sequence prediction is used to make the model learn the next sequence of a sentence. Two input pairs of sentences are received with a 50% change of the latter sentence in the pair being the original subsequent (Horev 2018). The BERT model is trained with both the mentioned objectives in place to minimize their combined loss function.

An Extension of BERT is the Robustly Optimized BERT Pretraining Approach (RoBERTa) where the model is able to predict intentionally hidden sections in text (Meta AI 2018). RoBERTa is built by modifying specific hyperparameters for training BERT such as removing the next sentence pretraining objective, increasing training batch sizes and incrementing learning rates which in turn improve its mask modeling language objective (Facebook AI 2019). This makes the RoBERTa a lot more effective for text classification purposes as the model will be able to generalize more.

3 Approach

In this section we describe the approaches taken to achieve the best performing model for the FNC-1 Fake News Challenge. We first implemented a gradient tree boosting baseline model. We then implemented two main approaches, using an LSTM based model and a Masked Language Model (MLM) while introducing significant modifications to both to achieve an optimal relative FNC-1 score.

3.1 Baseline Model

A gradient tree boosting classifier is used in the baseline model to determine the stances of the headline and body pairs of text. The implementation also applied pre-processing techniques by removing stop words, lowercasing, tokenizing, and lemmatizing the data. From the training dataset, 20% of the stances are split off for validation that are used for k-fold cross validation. In addition to this, n-gram overlap, and indicator features are created to support the model. The baseline model achieved 79.56% accuracy on dev set and 75.09% accuracy on test set, with an FNC score of 8748.75 on the test set. The subsequent models in this report aim to surpass the benchmarks of this baseline.

3.2 LSTM Model

We implemented an LSTM model for the FNC-1 task. For this we utilized several libraries for preprocessing, model formulation and training mainly `Tensorflow`, `Keras` and `Scitkit-learn`. Preprocessing the data consisted of parsing into a `Pandas Dataframe`, filtering out punctuation and lowercasing all heading and body text and tokenization.

3.2.1 Initial Model

The initial model consisted of two sets of Bi-LSTM's, one to intercept heading input and the other to intercept body input. Each LSTM was first equipped with an embedding layer with loaded pretrained embeddings. In the first iteration of the model, GloVe embeddings were selected and set as untrainable (Pennington, Socher and Manning 2014). Once each heading and body input would be passed through its respective embedding layer, the output of the embedding layer would then pass into a Bi-directional LSTM layer of size based on the embedding layer output. To fuse the two Bi-LSTM layers, the models were then flattened and concatenated together. Once the layers are fused a number of final fully connected layers are used to narrow the output size until a final softmax layer is used to classify the outputs.

3.2.2 MLM Word Embeddings

In an attempt to improve the performance of the model, BERT and RoBERTa pretrained embeddings were introduced. These MLM based embeddings replaced the original GloVe embeddings and hence the input embedding size was modified accordingly.

3.2.3 1D Convolutional Layers

The complexity of the model was also adjusted by adding more LSTM and hidden layers until an optimal structure was achieved. To further improve the ability of the model 1D convolutional layers were implemented between embedding and Bi-LSTM layers as seen in Figure 3. These followed by max pooling layers would receive embedding layer output, perform the convolution operation based on tuned filter and kernel size and pooled together to be passed on next to the Bi-LSTM model.

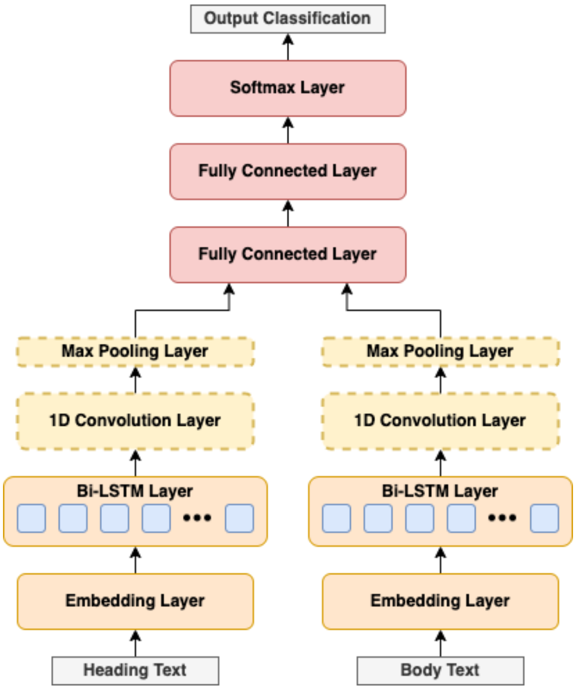

Figure 3: LSTM Model with 1D Convolution Layers

3.3 MLM Model

We then implemented a transformer-based model using the `HuggingFace` library wrapper `SimpleTransformers`. This allowed us easy access to the BERT and RoBERTa models to be used during training along with simpler usage in fitting the models for the fake news detection task. The preprocessing steps completed were similar to that done in the LSTM model trained for the FNC-1 task but with filtering and tokenization steps being done by the `SimpleTransformers` library.

3.3.1 Initial Model

The two popular MLM's of BERT and RoBERTa were selected for implementation for this task. They were modified to fit the fake news detection task so the models could receive both headline and body texts as input and perform adequate classification. More specifically, the classification wrapper used for the transformers was a sentence-pair classifier that directly suits the needs for this task. Within the `SimpleTransformers` wrapper, the MLM's succeed in this task by appending heading and body texts together with start, end and separation tokens being included to indicate heading from body to the model. Cross entropy by default is implemented and is used as the loss function which helps drive classification and improvement via backpropagation. The model's previous values from initial training as an MLM were preserved and were updated through classification on the FNC-1 task. This transfer learning approach was used to benefit from the unsupervised training and corpus the models were previously trained on while saving time and GPU computation. The initial model was used as the backbone for fine tuning on the FNC-1 dataset.

3.3.2 Data Augmentation

Due to the robustness and built-in complexity of BERT and RoBERTa models, we concluded that fundamental changes to the model's architecture would not lead to any tangible benefit. We theorized that modern MLM's like BERT and RoBERTa had enough complexity in their architecture to perform better on the FNC-1 task and would benefit noticeably from a larger sized corpus. Instead of modifying the model, the data fed into the model itself was investigated to determine if it could be modified within the rules of the competition to improve model performance. This was ultimately done by modifying the input body length and cropping the body of each labelled entry to make the task more difficult for the MLM. The MLM would receive alternating instances for each epoch of complete and incomplete bodies by degrees of 25%, 50% and 75% being cropped out at a time.

4 Experiments

4.1 Dataset

The FNC-1 dataset used for model training and evaluation consists of heading and body pairs classified as agree, disagree, discuss or unrelated. Bodies are reused for different classifications with other headlines, so it was important to split into training and validation datasets such that bodies are exclusive to one split set.

The bodies of text in the dataset were calculated to be on average 2283 characters long in the training set. This is relevant because the data augmentation technique reduces this by cropping each body length by a desired percentage.

4.2 Model Configurations

Each overarching model was tuned by iteratively selecting more appropriate hyperparameters. Models were trained in Google Colab using the GPU compute provided.

The LSTM models were typically trained for 30 epochs, but this was reduced if any of the specific models were observed to begin overfitting. LSTM layer sizes of 80 were found optimal for the task while hidden layers at the end of the model were tuned for each specific version of the LSTM being tested. Embedding layer size was selected based on the input embedding layer with 50 for GloVe embeddings and 768 for BERT embeddings. Additionally, Adam optimizer was used and remaining hyperparameters such as learning rate, batch size, heading max length and body max length were tuned. 1 dimensional convolutional layers with filter sizes of 32 and kernel sizes of 8 were used when testing their addition to the model along with max pooling layers with pool size of 2. The training time for the LSTM models took about 14 minutes for all 30 epochs.

The MLM models were fine tuned for 4 to 5 epochs at learning rates of either 0.00001 or 0.00003. The batch size used for the model was selected based on the learning rate as the two hyperparameters are inversely proportional. The max sequence length of the model was also tuned and the optimal selection of 512 was used for the model. The training time for the MLM models took about 3 hours for 4 epochs.

4.3 Results

4.3.1 Interim Results

LSTM and MLM models contained a variety of changes that were first evaluated using the validation set, a portion of data held out from received training stances and bodies. It was also determined that relative FNC score was a better metric to determine model suitability compared to accuracy as it is the metric being evaluated as part of the competition.

Figure 4 summarizes the performance of the four main architectural changes to the LSTM model. It can be seen that introducing 1D convolutional layers decreases the performance of the model from 78.49% to 76.35% when using GloVe embeddings. Additionally, using BERT embeddings did not improve performance either for with or without 1D convolutional layer settings. The final validation score of 78.49% was still less than the validation scores previously achieved with the baseline gradient boosting tree classifier model.

BERT, RoBERTa and RoBERTa with augmented data were tuned on the validation set and resulted in the parameters described in the model configurations section. The different degrees of 25, 50 and 75% body text cropping were experimented with, and it was concluded that cropping of 50% led to optimal performance.

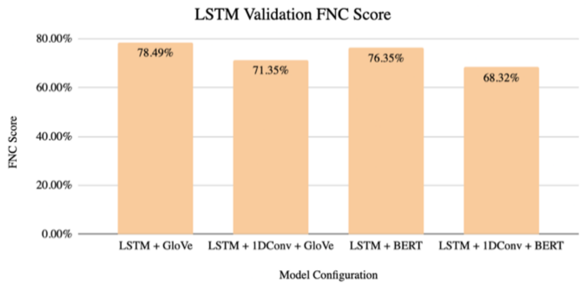

Figure 4: LSTM Model Validation FNC Performance

4.3.2 Final Results

Results of each model on the competition test set were calculated and can be seen in Table 1. Figure 5 visualizes the differences in performance in each model where the benefit of using MLM's instead of LSTM's is clear. BERT, RoBERTa and RoBERTa with augmentation all exceeded the performance of both the baseline and LSTM. Although both BERT and RoBERTa noticed significant improvements it can be concluded that RoBERTa is better suited for this type of multi-class classification. It can also be seen than the data augmentation technique used proved beneficial as it improved overall test score by 1.06% over the standard RoBERTa model fine-tuned on FNC-1 fake news detection.

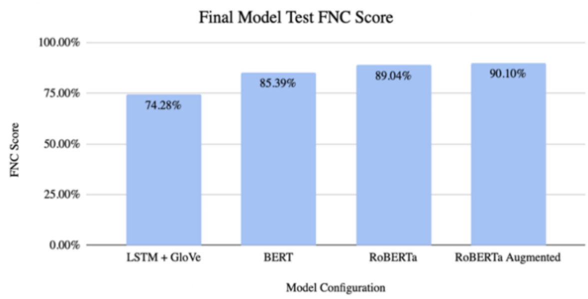

Figure 5:Final Model Test FNC Performance

Model	Relative FNC Score	
	Value	Percentage
Baseline	8761.25	75.20%
LSTM + GloVe	8654.50	74.28%
BERT	9949.00	85.39%
RoBERTa	10374.25	89.04%
RoBERTa + Aug.	**10498.00**	**90.10%**

Table 1: Final Model Test FNC Score

4.3.3 Additional Analysis

Doing more detailed analysis of the best performing RoBERTa model trained with 50% body cropping can be seen in the confusion matrix seen in Table 2. The table shows that disagree classifications have highest absolute value of false negative classifications with 663. It can also be seen that the discuss classification has the highest absolute value of false positives of 596.

	Agree	Disagree	Discuss	Unrelated
Agree	1430	189	381	15
Disagree	35	316	107	7
Discuss	416	164	3868	83
Unrelated	22	28	108	18244

Table 2: RoBERTa with Augmentation Confusion Matrix

Percentage precision, recall and F1 results for the same model are also presented in Table 3. The model performs to a high degree of success on unrelated classifications but very poorly on disagree classifications. Since the disagree classification is not common in the dataset the overall performance is still decent with a macro average F1 score of 78% and weighted average F1 score of 94%.

Metric	Precision	Recall	F1	Support
Agree	71%	75%	73%	1903
Disagree	68%	45%	54%	697
Discuss	85%	87%	86%	4464
Unrelated	99%	99%	99%	18349
Macro Avg	81%	77%	78%	25413
Weight Avg	94%	94%	94%	25413

Table 3: RoBERTa with Augmentation Metric Results

5 Conclusion

We can take away many conclusions from the experiments conducted and final results. It is clear that MLM's perform noticeably better than LSTM and convolution type approaches for the task of fake news detection. This is reasonable to assume when compared to MLM's since models like BERT and RoBERTa have been trained on massive corpuses and have great language understanding capabilities built-in before fine-tuning. Although, it is important to try modifications to LSTM models like changing word embeddings and experimenting with convolutional layers there is a gap in performance that cannot be achieved without more sophisticated additions.

Additionally, data augmentation can be very beneficial for models in the FNC-1 task competition. When paired with MLM's, data augmentation can allow the model's innate complexity to perform better. Specifically introducing forced cropping on text bodies should be experimented with more in the future and we are leaving ideas like cropping headings to future works. Although the final RoBERTa and augmented RoBERTa models were tuned, we were limited by lack of GPU computation resources to do a larger hyperparameter search and attempt training with no backbone. Additionally, variations to the augmentation technique should be investigated such as randomly cropping a new portion of the training set prior to each epoch similar to how dropout randomly removes nodes in a layer. Other data augmentation techniques outside of the rules of competition should also be investigated as potential areas of improvement for the model if targeting fake news detection outside of the FNC-1 competition.

6 References

Bahad, Pritika, Preeti Saxena, and Raj Kamal. 2020. "Fake news detection using bi-directional LSTM-recurrent neural network." Procedia Computer Science. February 27. https://www.sciencedirect.com/science/article/pii/S1877050920300806.

Challenge, Fake News. n.d. http://www.fakenewschallenge.org/.

Facebook AI. 2019. "RoBERTa: A Robustly Optimized BERT Pretraining Approach." July 26. https://arxiv.org/pdf/1907.11692.pdf.

Horev, Rani. 2018. "Bert explained: State of the art language model for NLP." Towards Data Science. November 17. https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270.

Joshi, Prateek. 2019. "How do Transformers Work in NLP? A Guide to the Latest State-of-the-Art Models." Analytics Vidha. June 19. https://www.analyticsvidhya.com/blog/2019/06/understanding-transformers-nlp-state-of-the-art-models/.

Magnone, Sal. 2022. TRANSFORMERS IN NLP: HOW DOES IT WORK. March 25. https://proxet.com/blog/transformers-in-nlp-how-does-it-work/.

Meta AI. 2018. Roberta: An optimized method for pretraining self-supervised NLP systems. https://ai.facebook.com/blog/roberta-an-optimized-method-for-pretraining-self-supervised-nlp-systems/.

Pal, Jayanta Kumar. 2021. Fake news detection using LSTM neural networks. January 15. https://jayant017.medium.com/fake-news-detection-using-lstm-neural-networks-5bfb158be55e.

Pennington, Jeffrey, Richard Socher, and Christopher Manning. 2014. "GloVe: Global Vectors for Word Representation." Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Doha: ACL. 1532-1543.

ProgrammerSought. n.d. Deep learning----NLP-transformer model . https://www.programmersought.com/article/4970540041/.

Vidhya, Analytics. 2020. "What is Bert: Bert for text classification." June 14. https://www.analyticsvidhya.com/blog/2019/09/demystifying-bert-groundbreaking-nlp-framework/.