Leveraging the BERT deep-learning model for fake news classification

Wu Hanhui (A0218237E), Tan Jing Xue Andre (A0215123X), Phua Anson (A0216176E), Tan Ka Shing (A0218409A), Liew Xin Yi (A0219723A)
Group 26
CS3244 Machine Learning Project Report
National University of Singapore

Abstract

Obtaining news online has become the new normal for many Singaporeans in the information age. The ease of discovering and sharing news with different news sources battling to control the narrative has led to a drastic increase in misinformation. Therefore we need to differentiate between real and fake news, and this project aims to design an application to prevent the spreading of fake news by employing the pre-trained BERT model to detect fake news.

Introduction

While local publishers such as Channel News Asia (CNA) and the Straits Times are reliable and unbiased, Singaporeans often consume online news from unfamiliar sources. Unfortunately, most of this information is not verified and may even be used to push a narrative. Amidst the COVID-19 pandemic, we encounter a surge in misinformation revolving around vaccines and remedies for the disease. What we hope to achieve is to identify and safeguard the two groups who are at risk of misinformation, specifically the elderly and children. They may lack the possible awareness and insights to fact-check against such news. They could be gullible and trust this news at face value, which would be critical in this crisis pandemic.

We aim to train a classification model on general fake news while well-trained classifiers exist for fake news of a specific category. The application shall be in the form of an extension to the browser or messenger applications, predicting whether the embedded links may be fake before redirecting users to the website. In addition, it allows users to judge and decide if they want to continue to the news site in the hope that such an option aids in mitigating the propagation of fake news by heightening their alerts when reading the articles should they proceed to the page.

As this is a binary text classification problem, we will investigate the viability of a transformer model, preprocessed with the state-of-the-art Natural Language Processing (NLP) technique for our problem at hand. Specifically, we will explore the pre-trained BERT model introduced by Google and fine-tune it with our dataset.

Data

Our training dataset is from Kaggle (2018), which contains 20799 articles from various domains and sources on the Internet. The dataset has four columns: *Title*, *Author*, *Text*, and *Label*.

We used both local and global news datasets to validate and evaluate our model if the trained model can be generalized and applied with high accuracy in both local and global contexts. We scrape data from three reputable Singapore news sources for local datasets: Straits Time, CNA, and Today Online. We assumed that news from these sources was accurate as they must follow Singapore's Newspaper and Printing Presses Act (NPAA).

However, fake local news datasets were not easy to obtain, especially amidst the introduction of the Protection from Online Falsehoods and Manipulation Act (POFMA) in 2019. POFMA is an act that prohibits the electronic communication of falsehoods from the use of online platforms. Even though POFMA does not cover opinions, criticisms, satire, or parody, it covers statements of fact in the sense that a typical reader would construe the statements as accurate and credible. The aspect of fake news covers the latter; thus, the POFMA act is enforced on local fake news, where many correction notices have to be published on erroneous articles. The POFMA act serves as a powerful deterrence with its potential fine of at least $50,000, a term of imprisonment of up to 5 years, or both (Singapore Legal Advice 2022). Furthermore, when we attempted to scrap local fake news data, we realized that most of the fake articles have been taken down by the publisher(s) and consequently inaccessible to the public.

Therefore, we only managed to validate our model on real, local news, with several downsides. We will address them in the metric section.

Evaluation Metric

Instead of just accuracy, we have decided on the confusion matrix as our evaluation metric. This is because evaluation by accuracy hides the fact that the input data may be unbalanced, e.g., 90% true input and 10% false input, where a naïve model of just returning true for all inputs will result in an accuracy of 90%. This is precisely the case for us, as our test set only consists of real news. Fundamentally, it is a performance measurement for machine learning classification where output can be two or more classes. The confusion matrix constitutes four different predicted and actual label combinations: True Positive, True Negative, False Positive, and False Negative. The four combinations can be interpreted in this manner: True Positive implies the model predicting an input instance to be TRUE when the target concept is True, whereas, in the case of True Negative, the model predicts the instance to be FALSE when the target concept is indeed FALSE. A False Positive (Type 1 Error) arises when the model predicts TRUE when the target concept of the instance is FALSE. A False Negative (Type 2 Error) arises when the model predicts a FALSE label when in reality, the target concept is TRUE. We will use the four combinations to ascertain the degree of precision, accuracy, and F1-score on our global test set and local validation set, as seen in Figure 1.

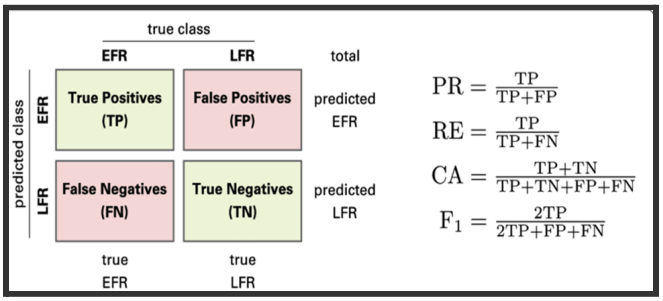

Figure 1: An image depicting how the Prediction, Recall, Accuracy and F1-Score are obtained from the Confusion Matrix (Bittrich 2019).

Problem Approach and Implementation

In our attempt to approach this classification problem, we decided to drop the *Author* column, as most are western names, and we feel that it will not generalize well in a local context. We then train three models separately with *Title* only, *Text* only, and *Title + Text* to experiment and identify the best-fit features that can produce the highest precision. To do the training, we split the Kaggle dataset into train, test, and validate, where we will train the models and validate their performance on global news data. Local data is then used for final validation to see their generalization power. During validation, we also tested the *Title*, *Text,* and *Title + Text* as the input to the same model to find out which combination would yield the best results.

Model

Since this is a text classification task, NLP models are the best fit for this task. After investigating the different models such as recurrent neural network (RNN), long short term memory (LSTM), and transformer, we chose to go with Transformer neural network as it produces better results.

Our model consists of an input layer, a BERT preprocessing layer, a BERT encoder layer, a dropout layer, and an output layer.

RNN/LSTM

Unlike traditional neural networks that cannot use prior information learned to inform about later ones, RNN allows information to be passed from one step of the network to the next (Colah. 2015). It is instrumental in NLP tasks as the understanding of each word is based on the understanding of previous words.

Unfortunately, as the input sequence grows, RNN cannot remember old connections and its ability to predict drops. This is where LSTM comes in. It is a special kind of RNN capable of solving this issue of "long-term dependencies".

However, both RNN and LSTM process data sequentially, leading to slower training as they cannot be trained parallelly. Therefore, to encode the current word, we need to compute the previous state first. This also means that LSTM does not fully solve the issue of "long-term dependencies" and there is still information loss over long sequences.

Transformer Neural Network

The transformer model was introduced in the paper *Attention Is All You Need* (Vaswani, A et al. 2017) based on a self-attention mechanism.

The self-attention works by relating different words of a single sequence and their positional values to compute a representation of the sequence seen in Figure 2.

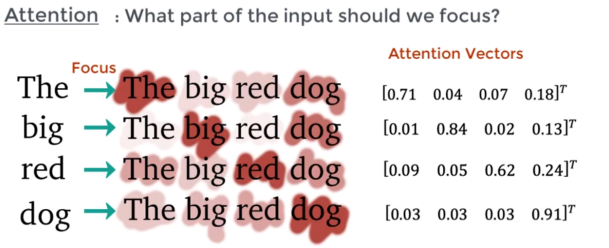

Figure 2: An image showing how the self-attention mechanism works to weigh words in the sentence (Ankit, U. 2020).

For every word, we will have an attention vector that captures the contextual relationship between words in the

sequence. As the process does not rely on previous states, the transformer is able to overcome the issue of "long-term dependencies" by a large margin. This also means that the transformer is bidirectional, unlike RNN/LSTM which is limited to either left to right or right to left. Furthermore, as each of the attention vectors can be seen as independent of each other, a parallel process is possible for the transformer, which makes training much faster (Ankit, U. 2020).

This led to the decision of the transformer model as we think that it can learn interesting relationships from the long article text and provide better classification precision.

BERT

BERT, which stands for Bidirectional Encoder Representations from Transformers, is a model developed by researchers at Google AI Language (Devlin, J et al. 2019). BERT is pre-trained using a large corpus of unlabeled text, Wikipedia (~2.5B words) and Google's BooksCorpus (~800M words). It is trained on two NLP tasks: Masked Language Modeling and Next Sentence Prediction.

This pre-training step on such a huge dataset can allow the model to pick up a deeper understanding of how languages work and this knowledge that it learns is useful for other NLP tasks. This is shown by its ability to produce state-of-the-art results in a wide variety of NLP tasks when fine-tuned with just one additional output layer.

With BERT's great results and the ability to transfer learning, we took to see if the BERT model can be fine-tuned to help in our task of fake news detection.

Data Preprocessing

In text classification problems, where inputs are texts, these inputs need to be transformed into numerical values before passing into the corresponding classifier. NLP is often used to analyze the text inputs in these problems. As the state-of-the-art model in NLP, each encoder model in the BERT family comes with a matching preprocessing model. Since the text preprocessor is a model, it can be incorporated directly into the final model to prevent common text preprocessing challenges, such as training-serving skew, efficiency and flexibility, and complex model interface.

Unlike traditional tokenization, where sentences are split into individual words, the text inputs are split into sub words or wordpieces instead, i.e., individual words are further split, according to a vocabulary generated from the Wordpiece algorithm. An interesting point to note is that besides the wordpieces, the vocabulary contains special tokens, such as [CLS] and [SEP], which will be elaborated on later in the section. After which, each wordpiece is converted into an integer, denoting its index in the vocabulary.

Since BERT inputs are to be of fixed size and shape, the inputs are required to undergo trimming. There are several ways to do trimming, e.g., trimming the end of sentences. After trimming, the trimmed segments are flattened and combined into a single tensor by adding special tokens in between segments. [CLS] is inserted at the beginning of the first segment while [SEP] is inserted at the end of every segment, including the first segment. Finally, the combined inputs are padded into a fixed 2-dimensional tensor.

The BERT preprocessor model returns three values, namely *input_word_ids*, *input_mask* and *input_type_ids*. *input_word_ids* denotes the indices of the wordpieces in the vocabulary, *input_mask* denotes whether the corresponding wordpiece is padded or not, and *input_type_ids* denotes which input the wordpiece belongs to.

Training

Loss Function

The most common loss function used in binary classification, which is the case of our problem of fake news classification, is binary cross-entropy or logs loss. The formula for binary cross-entropy is as follows

$$\frac{1}{N}\sum_{i=1}^{N} - (y_i \cdot log(p_i) + (1 - y_i) \cdot log(1 - p_i))$$

where y_i is the label of the input and p_i is the probability of the input being real news.

The usage of log value provides lesser penalties for smaller differences between the predicted and true probabilities, and vice versa, which helps optimize the performance of the undertrained model.

Optimization

We use the Adamw optimizer for our neural network training. Adamw computes individual adaptive learning rates for different parameters compared to stochastic gradient descent that maintains a single learning rate for all weight updates. Thus, training with Adamw, the model can usually converge a lot faster.

Results

Training/Test	*Title + Text*		*Title*		*Text*	
Title + Text	P:	1	P:	0.81	P:	0.56
	A:	0.99	A:	0.90	A:	0.66
	R:	0.99	R:	0.99	R:	0.99
	F1:	0.99	F1:	0.89	F1:	0.71
Title	P:	0.45	P:	0.98	P:	0.43
	A:	0.48	A:	0.98	A:	0.43
	R:	0.99	R:	0.98	R:	0.99
	F1:	0.62	F1:	0.98	F1:	0.60
Text	P:	0.95	P:	0.45	P:	0.99
	A:	0.97	A:	0.47	A:	0.99
	R:	0.99	R:	0.99	R:	0.99
	F1:	0.97	F1:	0.62	F1:	0.99

Figure 3: A table depicting attributes that are trained/tested on the global dataset.

Training/Test	*Title + Text*		*Title*		*Text*	
Title + Text	P:	1	P:	1	P:	1
	A:	0.93	A:	0.99	A:	0.91
	R:	0.93	R:	0.99	R:	0.91
	F1:	0.96	F1:	0.99	F1:	0.95
Title	P:	1	P:	1	P:	1
	A:	0.99	A:	0.81	A:	0.99
	R:	0.99	R:	0.81	R:	0.99
	F1:	0.99	F1:	0.89	F1:	0.99
Text	P:	1	P:	1	P:	1
	A:	0.76	A:	0.99	A:	0.37
	R:	0.76	R:	0.99	R:	0.37
	F1:	0.86	F1:	0.99	F1:	0.54

Figure 4: A table depicting attributes that are trained/tested on the local dataset.

Figure 3 and Figure 4 show the results from the global and local validation set with the rows as the training method and the columns as the testing method. Their precision, accuracy, recall, and F1-score were collected, rounded down, and labeled P, A, R, and F1.

Studying the results we obtained, we can see that both global and local validation sets manage to achieve an accuracy of 99%. However, the training and testing methods that yield the best results differ for the global and local validation sets. The global validation set from Kaggle performs the best when its training and testing set is the same type as expected. In contrast, the local validation set does not perform as well when using the same combination; instead, having seemingly random combinations yield the best results.

Intuitively, we would expect the *Text* of an article to be an essential factor as it contains the most entropy. Therefore, it was a surprise that the local news *Text* did not yield a high accuracy when the model was trained using the *Text* of the global news and was, in fact, the worst-performing combination overall, with an accuracy score of only 37%.

However, training the model with *Title* and testing with *Text* or *Title* + *Text* allows BERT to predict local news with near-perfect accuracy. Training the model with *Text* or *Title* + *Text* and testing with *Title* seems to hold this finding.

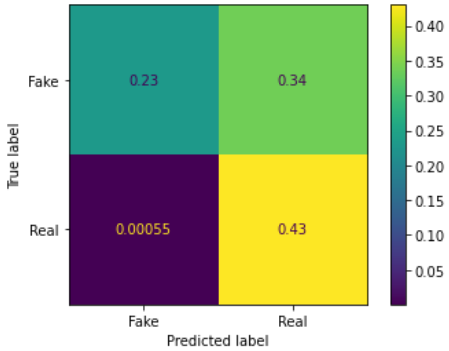

Figure 5: An example of data from the confusion matrix.

Evaluations

Generally, our model could predict near-accurate results for many of the testing dataset's labels. However, the results from the global and local sets differ significantly with the combination of training and testing methods to obtain the best results differing. One possible reason could be the difference in the writing styles for global news and local news, with global news preferring to embellish its stories while local news is more factual and objective.

It is possible to deduce that the *Title* and *Text* features are highly correlated to classifying its validity and perform differently in the local news context than in global news. For example, a hypothesis can be made that BERT seemingly views *Text* and *Title* + *Text* similarly when we are using the local validation set during testing. Another hypothesis that can be made is that if we want to train the model with the global news *Title* or *Text*, we would need to use the other if we are using local news as the validation set to achieve the best results.

However, a limitation we face is that, given the black box and our tokenization, we cannot identify the keywords that determine one's label, i.e., we do not know the set or combination of words that allow BERT to classify a news label as fake. This would have been useful information if we wanted to improve our model further.

We could also have explored another feature - Author, to train the model and have BERT form a correlation between the author's name and the validity of an article. An argument could be made that given how strong our existing classification power already is, would it be worth running additional computational costs to improve our model?

Other Discussions

Comparing possible models

Naive-Bayes vs Linear Regression

Naïve-Bayes utilizes a probabilistic model that assumes every feature is mutually independent and equal. It predicts the posterior probability, P (x | y), of a problem instance by applying Bayes' Rule after learning the joint probability P (x, y). On the other hand, Linear Regression considers both independent and dependent variables to find the best fit. Therefore, it is no surprise that Linear Regression performs much better than Naïve-Bayes in detecting fake news due to the nature of Naïve-Bayes assuming that every word is independent of another.

NLP

However, that is not to say that Linear Regression is the best model we can have. We should also consider NLP for our machine learning model as we are trying to solve a text-based problem. Therefore, using probability or regression only might be insufficient in predicting language-based problems with high probability as both models do not use NLP.

BERT vs GloVe

GloVe, which stands for Global Vectors for Word Representation, uses unsupervised learning to obtain a vector representation for words performing training on a corpus's aggregated global word-word co-occurrence statistics. Therefore, it is no surprise that BERT performs better as GloVe is a context-free model that generates only single word embedding representation for words in the vocabulary (Pennington, J et al. 2014). On the other hand, BERT considers the context for each word occurrence. For example, GloVe will have the same vector representation for the word "left" in the sentences "Take a left turn" and "I left my house", while BERT would be able to provide a contextualized embedding that will differentiate the sentences.

Observation after training with BERT

The bar charts in Figure 6 below show the top 20 frequently used words after BERT classifies the news article as real or fake from the global dataset used to train the model. The word cloud in Figure 7 is a general visual representation of common words found in our input instances of real and fake news.

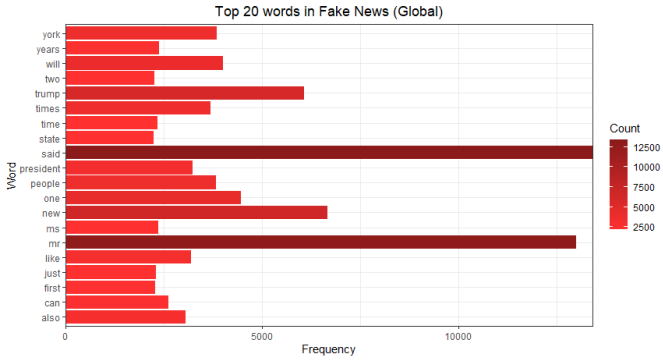

Figure 6: A bar plot featuring the top 20 words found in fake news from the global dataset.

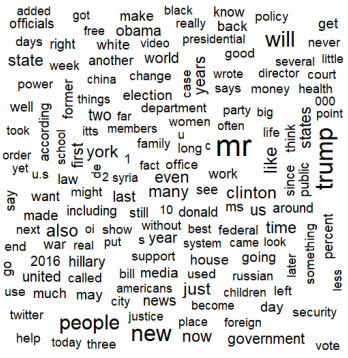

Figure 7: An image of the word cloud featuring common words found in real and fake news from the global dataset.

The top 20 real and fake news words are similar in political and official contexts, e.g., the common word "trump". This coheres with the word cloud, showcasing that political words are prevalent in real and fake news.

Unlike other NLP models' classification, BERT avoids single-word analysis too. Therefore, common words in our bar plots were not consequential in BERT's role in its prediction. Contrary, BERT capitalizes on the Transformer, an attention mechanism that learns the contextual relations between words/subwords in texts for prediction. This bolsters its accuracy and precision, as substantiated in the Results and Evaluations.

Conclusion

The notion of fake news may undermine Singapore citizens' confidence in their nation's media. This may likely stir internal conflict and possibly divide communities. However, for a person to identify all fake news online is difficult. That is why we need a machine

learning model to help us with detection. We have chosen BERT due to its ability to transfer learning and outperform other models.

Using the BERT Deep-Learning Model, we have come up with a model that accurately predicts the validity of local news through various combinations of training and testing methods. While we can form the hypothesis that to obtain the best results for local news sources, training the model with the global news *Title* or *Text* would require us to use the other, explorations with additional features such as Author or Date of publication might yield us better accuracy thus further testing could be conducted.

With the current model being able to predict fake news with such high accuracy for local news, there should be an effort to fine-tune it further. Therefore, it is crucial to provide Singaporeans with an option to check the legitimacy of a new article to protect us from misinformation, especially in the information age.

Roles

For the project, we split it into five stages, research, implementation, training, evaluation, and report writing. All team members participate equally in every stage of the process with no clear separation of roles.

References

Ankit, U. 2020. Transformer Neural Network: Step-By-Step Breakdown of the Beast. https://towardsdatascience.com/transformer-neural-network-step-by-step-breakdown-of-the-beast-b3e096dc857f. Accessed: April 15, 2022.

Bittrich, S.; Kaden, M.; Leberecht, C.; Kaiser, F., Villmann, T.; and Labudde, D. 2019. Application of an interpretable classification model on Early Folding Residues during protein folding. BioData Mining, 12(1), 1. doi.org/10.1186/s13040-018-0188-2.

Brownlee, J. 2020. How to choose a feature selection method for machine learning. https://machinelearningmastery.com/feature-selection-with-real-and-categorical-data/. Accessed: April 15, 2022.

Colah. 2015. Understanding LSTM Networks. https://colah.github.io/posts/2015-08-Understanding-LSTMs. Accessed: April 15, 2022.

Deepak, p. 2020. Ethical Considerations in Data-Driven Fake News Detection. In: Data Science for Fake News. The Information Retrieval Series, vol 42. Springer, Cham. doi.org/10.1007/978-3-030-62696-9_10.

Devlin, J.; Ming-Wei, C.; Kenton, L.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805.

Kaggle. 2018. Fake news. https://www.kaggle.com/c/fake-news. Accessed: April 15, 2022.

Pennington, J.; Manning, C. D.; and Socher, R. 2014. Glove: Global vectors for word representation. Stanford. https://nlp.stanford.edu/projects/glove/. Accessed: April 15, 2022.

Singapore Legal Advice. 2022. Singapore Fake News Laws: Guide to POFMA (Protection from Online Falsehoods and Manipulation Act). https://singaporelegaladvice.com/law-articles/singapore-fake-news-protection-online-falsehoods-manipulation/. Accessed: April 15, 2022.

Tensorflow. 2022a. Bert preprocessing with TF text. https://www.tensorflow.org/text/guide/bert_preprocessing_guide. Accessed: April 15, 2022.

TensorFlow. 2022b. Classify text with Bert. https://www.tensorflow.org/text/tutorials/classify_text_with_bert. Accessed: April 15, 2022.

Vaswani, A,; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, N.; Kaiser, L.; and Polosukhin, L. 2017. Attention is all you need. arXiv:1706.03762.