Stance Prediction of Fake News Detection Problem

Guy McCombe
24th May 2021

Introduction

Stance prediction is an important steppingstone in solving the fake news detection problem. A powerful stance prediction model can be used by human fact checkers to quickly gather arguments for and against any claim and use these arguments in conjunction with their expertise to discern whether a claim is true [1]. Furthermore, this model can be used in a larger truth labelling system of models, labelling claims based on the different stances by news outlets [1].

Problem Definition

The task is to predict whether the headlines and bodies of news articles are related, and then for each related article, determine whether the body discusses, agrees, or disagrees with the claim in the headline. The dataset is imbalanced with most articles being unrelated, so mitigation techniques should be explored.

Proposed Solutions

My solution leverages three feature extraction techniques. The first feature extraction technique applied was Term Frequency-Inverse Document Frequency (TF-IDF). This method infers importance of words proportionally to the difference between the frequency of the word in the headline or body compared to the frequency of the word in the whole corpus. The intention of this is that words pertinent to a topic will only be used in documents about the topic, leading to a high TF-IDF score, whereas common, insignificant words have a low TF-IDF score.

The other feature extraction techniques are iterations on the BERT transformer. Attention in Deep Learning focuses a predictor on certain parts of a text sequence to predict an output, whereas transformers leverage self-attention to relate positions in a text sequence to produce an encoding of that sequence. Transformers are typically assembled as an encoder/decoder pair; however, BERT differs from the norm as it is a language model and therefore only features an encoder. BERT sets itself apart from other language models as it uses Masked Language Modelling on the full context of a sentence, rather than just the words that have come before. The BERT encoder represents the semantic meaning of a sentence as 768-size vector. Two BERT iterations were applied: SBERT, a modification for sentence similarity comparisons [2], and DistilBERT, a faster version of BERT [3]. For both transformers, pretrained, cased models were selected to preserve the additional semantic information implied by capitalisation. TF-IDF vectors are much larger than BERT embeddings, but however can be obtained much faster. In addition, TF-IDF provide a strong metric for calculating the similarity of text but is lacking when attempting to derive its semantic meaning.

A logistic regression model is proposed for the binary related/unrelated classification problem. This model was fitted to the cosine difference of the encodings of the headline and body. Separate models were trained for both TF-IDF and SBERT feature

extractors. In attempt to mitigate the imbalance problem, these features will be also be oversampled by SMOTE to produce a balanced dataset to produce four total machine learning techniques.

A GRU RNN is also proposed for the related/unrelated task. A GRU is chosen over a “vanilla” RNN as it solves the short-term memory problem of RNNs, and it was chosen over LSTMs as it produces a similar quality of results [4], but with a fewer parameters due to a reduction in the number of gates in the network as shown in Figure 1.

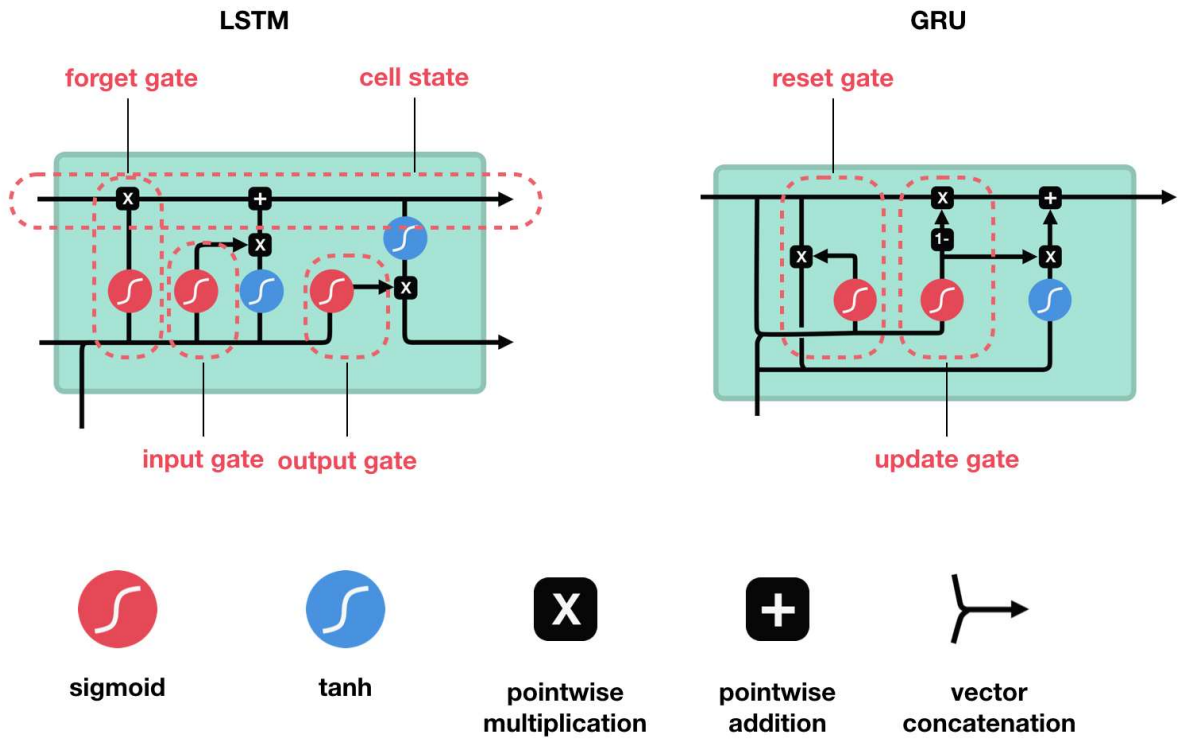

Figure 1: A Comparison of LSTM and GRU Network Structure [5]

After five GRU layers, the data is passed through a ReLU, Linear, and Sigmoid layer to reduce the output to a singleton probability value. This model was trained on both TF-IDF and SBERT data, vis the encodings of the headline, body, and the cosine difference of the two vectors. The full architecture is described in Figure 2.

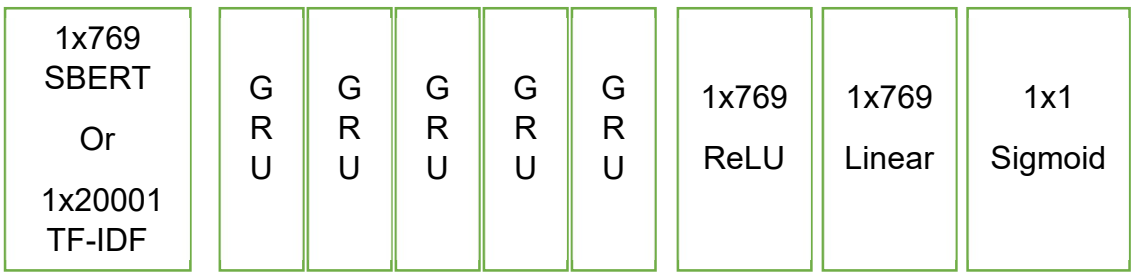

Figure 2: Binary Classifcation Architecture

The ADAM optimiser is used to backpropagate the gradients, with the optimal learning rate automatically determined by the cyclical learning rates technique [5]. The loss function selected is binary cross-entropy, the de-facto standard for binary classification tasks [6]. An alternative weighted loss function was also trialled, with the weight

proportional to the ratio of related and unrelated articles to mitigate their imbalance. The batch size was 512 and 256 for the TF-IDF and SBERT models, respectively. These values were chosen experimentally as the largest power of two before the model failed to converge or the memory limit was exceeded.

For the same reasons as before, a similar GRU architecture was selected for the multi-class classification problem. The difference is driven by the dimension of the output: the final layer is replaced by a Softmax to yield a probability distribution across the three classes. The input is now a batch of 128 DistilBERT encoded headline and body vectors, as shown in the architecture diagram in fig. The optimiser and learning rate remain the same as the previous solution, but the loss function is replaced with cross-entropy and weighted cross-entropy loss.

The final model consists of the most successful multi-class classifier, which receives as input the articles deemed “related” by the most successful binary classifier.

Analysis of Results

Related/Unrelated Logistic Regression

Model	Related Accuracy Support: 7064	Unrelated Accuracy Support: 18,349	Overall Accuracy Support: 25,413
TF-IDF	93%	97%	96%
Oversampled TF-IDF	94%	98%	97%
SBERT	94%	98%	97%
Oversampled SBERT	**96%**	**98%**	**98%**

The logistic regression model trained on SBERT is the most accurate, with oversampling only improving the TF-IDF model.

Related/Unrelated Deep Learning

Model	Related Accuracy Support: 7064	Unrelated Accuracy Support: 18,349	Overall Accuracy Support: 25,413
TF-IDF	76.82%	96.10%	90.92%
Weighted TF-IDF	31.82%	**99.44%**	81.29%
SBERT	**90.36%**	97.64%	**95.63%**
Weighted SBERT	84.63%	97.38%	93.39%

Once again, SBERT is the strongest of the metrics. Surprisingly, weighting to adjust for the imbalance does not result in an increase in the accuracy of the under-represented class.

Agree/Disagree/Discuss Classification

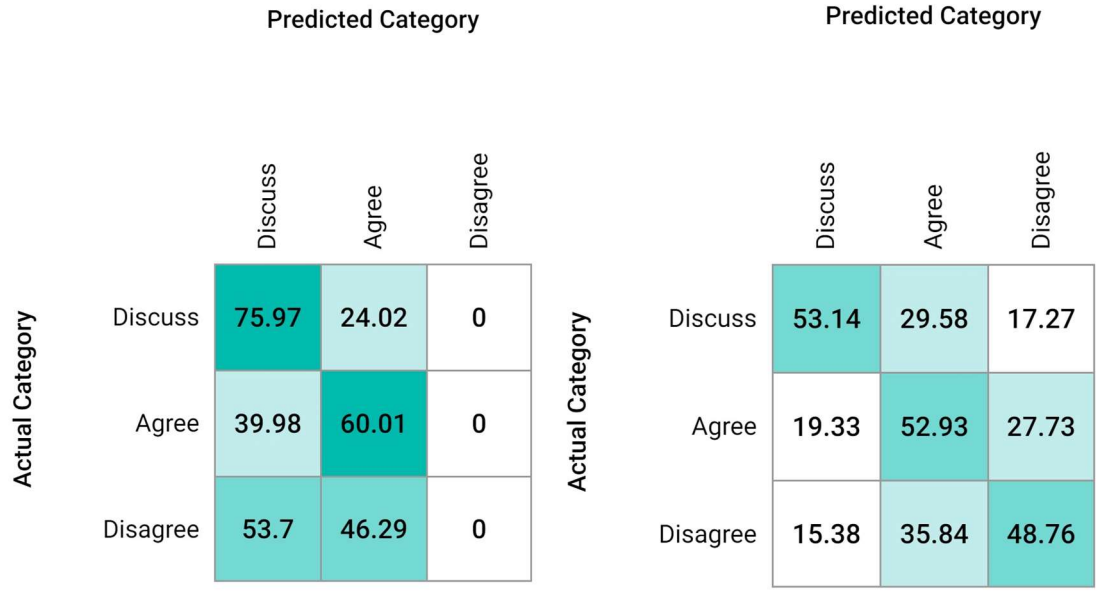

Figure 3: Confusion Matrices for unweighted (left) and weighted (right) multi-class predictors

This provides the expected result of the unweighted classifier having a strong overall performance but, performing extremely weakly on the rare disagree class. Whereas, the performance of the weighted model is weaker, but uniform across all the classes.

Combined

Stance	Accuracy
Unrelated	98%
Discuss	75%
Agree	59%
Disagree	0%

These results are obtained from combining, end-to-end, the SBERT logistic regression and the unweighted multi-class classifier. The low disagree accuracy could be increased by replacing the last classifier with the weighted variant, for a lower overall accuracy but a higher accuracy in the disagree category.

Discussion

Despite access to more data, parameters and a longer training time, the Deep Learning approaches were unable to beat machine learning at the binary classification task. This isn't unexpected as determining whether two texts are similar is a relatively simple problem, best suited to simple machine learning approaches.

In these binary classification tasks, the deep learning models quickly trained to a plateau due once again to the simplicity of the task. A sample loss graph from a training cycle is included in Figure 4. As both SBERT and TF-IDF encodings can be feasibly computed before training, the training time is similar for both families of models.

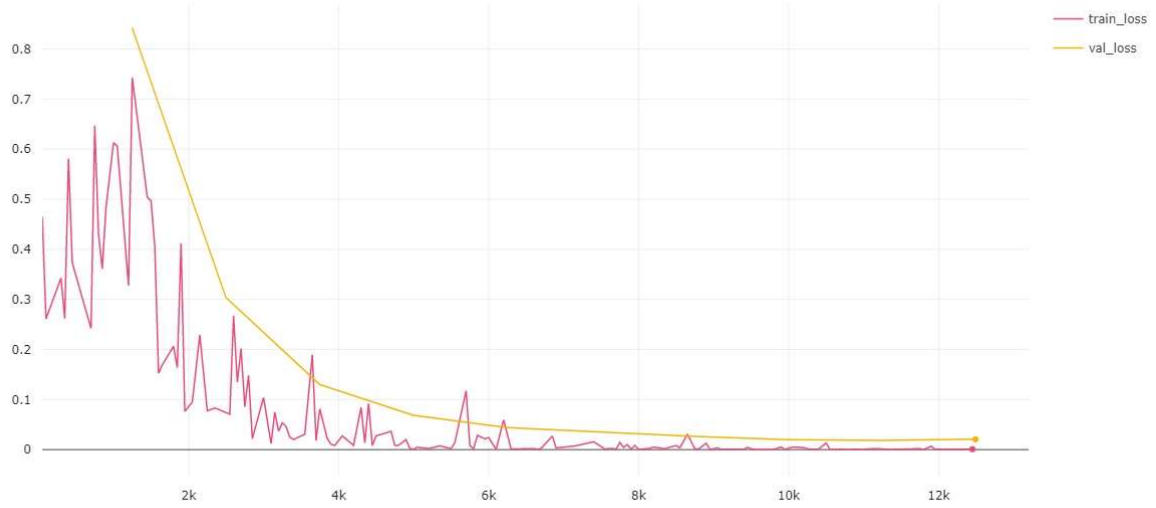

Figure 4: Binary Classifier, Loss vs Steps

In contrast, the training time for the multi-class classifier was much more time consuming as, due to its size, it is infeasible to compute and store the DistilBERT encodings in memory prior to training. This necessity to compute the encodings on the fly introduces a large overhead to training times. As a result, it was not possible to train the DistilBERT multi-class classifiers to completion, but the results above provide a good indicator of their performance.

Both weighted and unweighted classifiers show potential for separate uses, with the unweighted model showing promise as a general predictor, and the weighted model proving excellent at detecting the sparse disagree category. Perhaps a blend of the two models will lead to a greater success.

Ethical Implications

BERT and similar transformers have been pretrained on a biased dataset, with word embeddings revealing many human biases. This bias is dangerous, as introducing bias in fact checking of fake news could favour one party over another. In a world where fake news is rumoured to be a key contributor to election results, fact checking must remain impartial to not filter out certain party's fake news and promote others' as fact instead.

Conclusion

This work proposes a solution to a sub-problem of combatting fake news. While it proves to be very accurate at classifying stance, with an overall accuracy of 97%, it should be applied carefully to the problem of fake news detection as unavoidable bias will exist within the model.

References

[1] FakeNewsChallenge, “FAQ,” 27 September 2017. [Online]. Available: http://www.fakenewschallenge.org/#faq. [Accessed 24 May 2021].

[2] N. Reimers and I. Gurevych, “Sentence-bert: Sentence embeddings using siamese bert-networks,” *arXiv preprint arXiv:1908.10084,* 2019.

[3] V. Sanh, L. Debut, J. Chaumond and T. Wolf, “Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter,” *arXiv preprint arXiv:1910.01108,* 2019.

[4] J. Chung, C. Gulcehre, K. Cho and Y. Bengio, “Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling,” *arXiv preprint arXiv:1412.3555,* 2014.

[5] M. Phi, “Illustrated Guide to LSTM’s and GRU’s: A step by step explanation,” 24 September 2018. [Online]. Available: https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21. [Accessed 24 May 2021].

[6] L. N. Smith, “Cyclical learning rates for training neural networks,” *IEEE winter conference on applications of computer vision (WACV),* pp. 464-472, 2017 .

[7] J. Brownlee, “How to Choose Loss Functions When Training Deep Learning Neural Networks,” 30 January 2019. [Online]. Available: https://machinelearningmastery.com/how-to-choose-loss-functions-when-training-deep-learning-neural-networks/. [Accessed 24 May 2021].