Stance Prediction Using Word Embeddings

May 2021

1 Introduction

The goal of the Fake News Challenge is to apply artificial intelligence technologies to combat the rising problem of fake news [2]. It was first introduced in 2016-2017 but has seen a rise in popularity with the increasing use of social media as a form of news source, even though their use has declined in the past year [3]. The task can be broken down in smaller tractable sub-problems which can help produce a better overall solution for the original one.

2 Problem Definition

Specifically, the first stage describes automating a process that recognises the stance of various news sources about a specific issue. This problem is introduced as Stance Detection and specifically, given an input pair of a headline and a body text, the output of a solution should be able to accurately classify the stance of the body relative to the headline in one of four distinct categories. These categories are: **agrees**, **disagrees**, **discuss** and **unrelated**.

3 Proposed Solutions

Instead of passing the text to a classification model, we can use a embedding of it as our features. One such feature extraction method is the Term-Frequency-Inverse-Document-Frequency or TF-IDF which combines two statistics. Term-Frequency counts the number of times a term appears in a document while inverse-document-frequency counts the number of documents in

which a term appears. TF-IDF is simple to implement and as it is a combination of statistics, it is easier to understand and interpret than other machine learning techniques. However, due to its sparse tabular structure it requires large amounts of space and it will typically take longer to operate with.
Another method to extract features is the use of transformers. By using the output of a pre-trained encoder, we can get a contextual embedding per token or an aggregated one. A disadvantage of using pre-trained models, however, is that they have been trained in sequences with a set maximum length and thus, our inputs would need to be modified accordingly by either padding or truncating. Moreover, as they are a black box model, it is impossible to interpret these embeddings and have to rely on the results of our classification to measure the quality of the transformer output.

3.1 Methodology

Using a pre-trained base-case BERT transformer, we can train our models based on these embeddings. To do so, a sequence of both bodies and headlines is passed for each tokenization which returns the token encoded representation either trimmed or padded to 512 tokens, as that is the maximum length that the transformer can handle.
For our task, we compare both the TF-IDF and BERT embedding representations of the data as our inputs in both standard machine learning and deep learning methods. The machine learning methods tested and compared were the Random-Forest Classifier (RF), Support vector Machines (SVM). For deep learning methods, a multi-level perceptron (MLP) was also tested with one hidden layer as it was used by UCL Machine Reading during the challenge to achieve the third best results with the use of TF-IDF features [4]. Lastly, for TF-IDF a Convolutional Neural Network (CNN) was implemented and tested while for the BERT embeddings a Gated-Recurrent-Unit (GRU) architecture was used. CNN was used to extract spatial information from the tabular data that we had in binary classification while GRU was used as for the multi-class classification as we had the series of tokens in order with their embeddings. For binary classification Binary Cross Entropy with Logits Loss was used which also applies a sigmoid activation function to our output while for multi-class Cross Entropy loss was used instead.

4 Analysis of Results

All the results come from the competition dataset based on which the scores were calculated. We measure the performance based on Accuracy and F1-Score while for the final model we calculate the scoring end-to-end and compare it with the entries of the competition.

4.1 Binary Classification

For both the TF-IDF tests and the BERT tests the same training and testing data were used, sampled from the *train_bodies.csv* and *train_stances.csv*. The macro average of the F1 Score is taken as the classes were imbalanced and all networks were better at predicting the majority (unrelated) class. Thus, this has the potential to skew the results, favouring the majority class and seem like a model is performing better than it is.

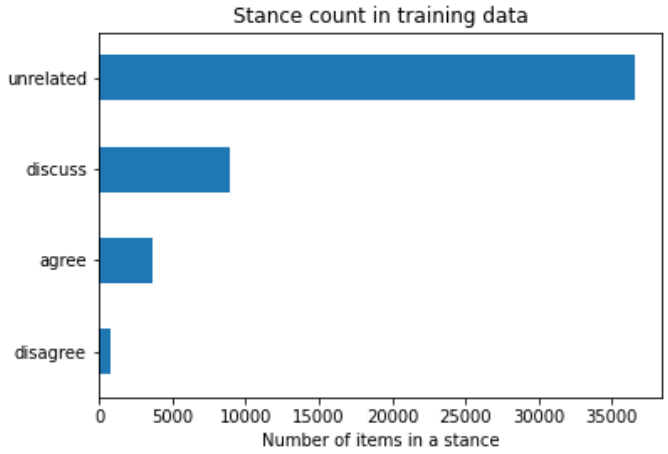

Figure 1: Class Count

4.1.1 TF-IDF Embedding Results

TF-IDF embeddings performed marginally better with the use of stop words in binary classification. Thus, we do not remove the stop-words before calculating the frequency matrices.

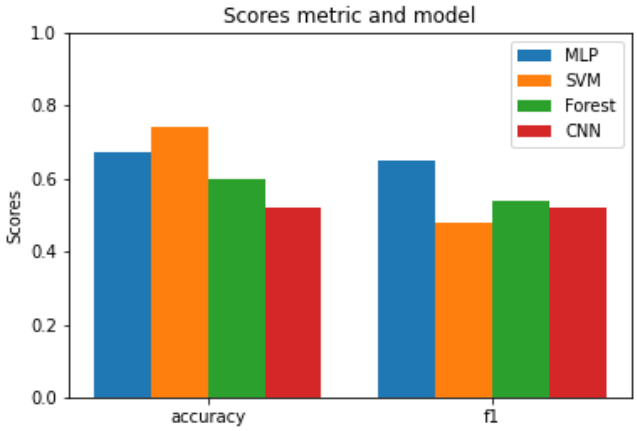

Figure 2: Model Results with TF-IDF on test set

While the TF-IDF embeddings generally exhibited adequate performance for the train and validation sets, their ability to predict the binary stance in the test set was sub-par for every algorithm. Because of the sparsity of the embeddings, the models took a long time to train.

Model	Time to Train (minutes)
RF	0.4
SVM	100.5
MLP	18.4
CNN	13.3

4.1.2 BERT Embedding Results

For the binary classification, the pooler (aggregated) output of the embeddings were used to train and test the models.

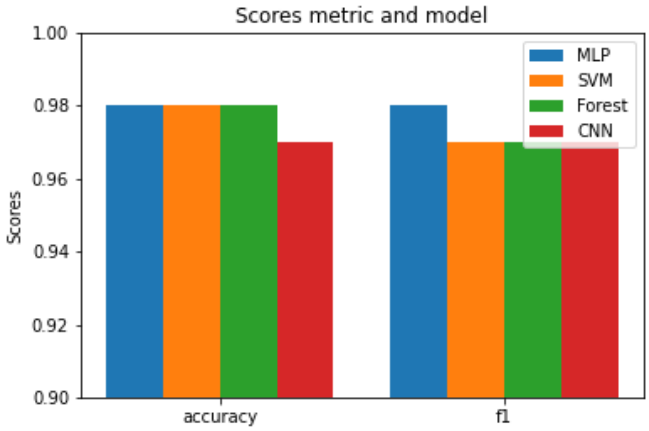

Figure 3: BERT Classification Results

While BERT embeddings performed much better, the clear winner here is the MLP model which had the highest accuracy. Thus, this is the model used for binary classification in the final end-to-end architecture. Below is the confusion matrix of the BERT-MLP for the competition test set:

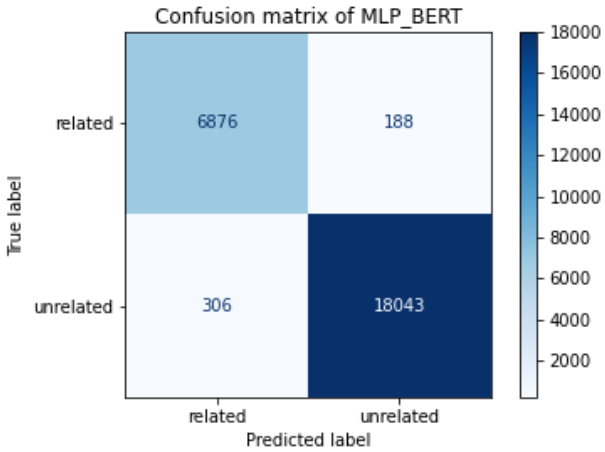

Figure 4: Confusion Matrix of Binary MLP

This classification was already able to better classify the unrelated and related classes compared to the baseline. As the inputs were arrays of 768

elements, the training times were much shorter than TF-IDF. However, one must factor in the time taken to tokenize and pass these tokens through BERT to get the embedding. Encoding and creating embeddings for the training data took 6 minutes and 30 minutes respectively.

4.2 Multi-class Classification Results

For multi-class classification, BERT embeddings were used as they exhibited the best classification performance in the binary case. However, the pooler output performance was not satisfactory thus, the last hidden state embedding was used instead. Therefore, each of the 512 tokens had its own contextual embedding. Here, an MLP and a GRU were tested with MLP having the pooler output embedding as input while GRU received the last hidden state as input. As the last hidden state was too large to be stored in memory for every sample, it was iteratively generated for every batch in training and testing.

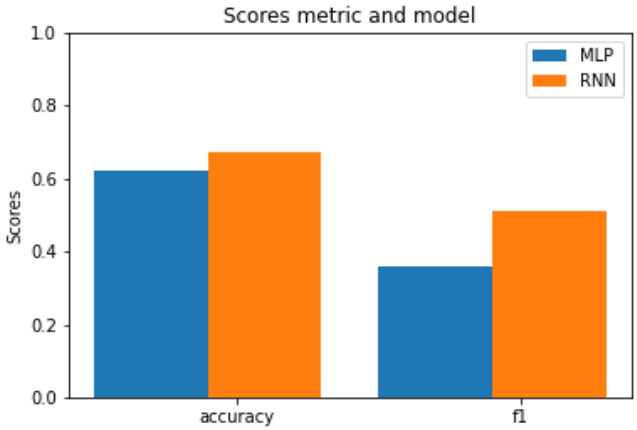

Figure 5: Multi-class Classification MLP and GRU scores

Model	Time to train (minutes)
MLP	19.2
GRU	65.4

While the GRU took much longer to train, it exhibited much better performance as it had a higher macro F1-Score. Having a higher score is difficult

as the class "disagree" has only a few data points and MLP failed to detect any of them. Below, a confusion matrix of the classification of the GRU is shown.

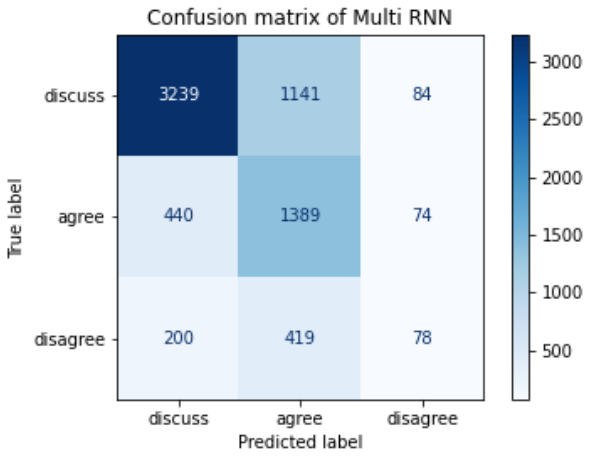

Figure 6: Multi-GRU Confusion Matrix

The results are taken from the test set without the unlabelled data assuming a perfect binary classification. However, we need to test the model end-to-end to see the combined accuracy of the architecture.

4.3 End-to-end Results

Using the binary MLP and the multi-class GRU, we are able to obtain results better than the winner of the competition. Specifically, the accuracy and F1 Score of the entire architecture were 0.89 and 0.61 respectively. Moreover, the total score was calculated based on the algorithm provided from the fake news challenge competition. The architecture score 9669.25 points out of 11651.25 possible for a total of 82.99% relative score. Below is the confusion matrix of the final end-to-end classification:

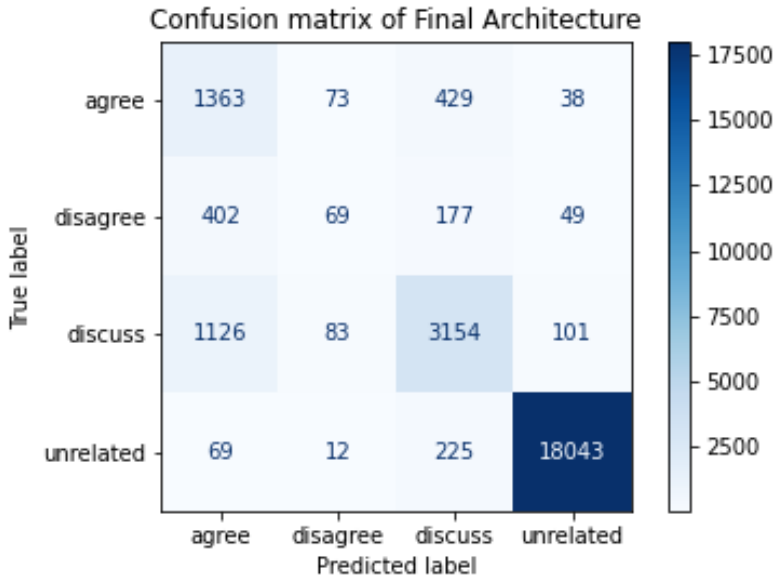

Figure 7: Confusion Matrix of End-To-End Testing

5 Discussion

As expected, the unrelated class was easy to predict in most cases while the disagree class was not. Methods to account for class imbalance could be utilised, such as oversampling, that have the potential to improve the detection rate of the minority classes further. Separating the task between a binary and a multi class classification assisted in getting better results for these minority classes.

The winner of the competition, Talos Intelligence, had a final score of 9556.50 and a relative score of 82.02% while the baseline had scores of 8761.75 and 75.20% respectively. Thus, our model was able to outperform both using BERT features. As the hidden state holds more information, utilising that for the harder, multi-class task provided better results than the lighter and faster pooler output embedding. In the future, other embeddings may be used that can more accurately represent the data and thus, perform better in the classification.

6 Ethical Implications

As TF-IDF is a statistical metric it cannot capture the context behind terms for which it calculates their frequencies. The use of it in a classification architecture can have both positive and negative results as on one hand it will not have many of the potential biases that originate from natural language use. On the other hand, it will not capture the negative context behind sentences. So, by using BERT and data sources that have implicit or explicit bias, our model will better understand and predict the context of a word but will also include its bias.
While the aim of the original challenge and of this report were to assist in tackling fake news, applying this report in different scenarios could potentially lead to more misinformation by blocking articles which a user may disagree with. A recent study on the presence of echo-chambers online found that: "Indeed, users online tend to prefer information adhering to their worldviews" [1].
Such, and other similar applications could also be of use by social media themselves to classify and potentially remove certain articles. However, as stated before, these pre trained-models, word embeddings and even training data may carry potential bias, most commonly in relationship with gender, race, and religion [5]. Such methods should be applied after extensive testing into their bias and how to mitigate them. Thus, it is important to critically analyse the application of the above solution as it can have a significant impact on the type and quality of information which the public receives.

7 Conclusion

In this report we look at predicting the stance that an article body has in relation with a headline. By using features extracted from a BERT transformer and a two-step classification we were able to get better performance than the winner of the competition with a final score of 9669.25 on the competition test data. The final accuracy was 89% while the F1-score it was 61%

References

[1] Matteo Cinelli et al. "The echo chamber effect on social media". In: *Proceedings of the National Academy of Sciences* 118.9 (2021). ISSN:

0027-8424. DOI: 10.1073/pnas.2023301118. eprint: https://www.pnas.org/content/118/9/e2023301118.full.pdf. URL: https://www.pnas.org/content/118/9/e2023301118.

[2] D Pomerleau and D Rao. *Fake News Challenge Stage 1*. URL: http://www.fakenewschallenge.org/.

[3] Jigsaw Research. *News Consumption in the UK: 2020*. URL: https://www.ofcom.org.uk/__data/assets/pdf_file/0013/201316/news-consumption-2020-report.pdf.

[4] Benjamin Riedel et al. "A simple but tough-to-beat baseline for the Fake News Challenge stance detection task". In: *CoRR* abs/1707.03264 (2017). arXiv: 1707.03264. URL: http://arxiv.org/abs/1707.03264.

[5] Tony Sun et al. "Mitigating Gender Bias in Natural Language Processing: Literature Review". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, July 2019, pp. 1630–1640. DOI: 10.18653/v1/P19-1159. URL: https://www.aclweb.org/anthology/P19-1159.