Fake News Detection

This coursework aims to train machine learning and deep learning models in the task of classifying ‘stances’ of news headlines and articles. To do this we use the Fake News dataset. This Dataset contains headlines and text bodies along with a stance which is either: unrelated, disagree, discuss and agree. A stance is the measure of how well a headline represents an articles body of text.

With this dataset we are able to train models in the task of classifying an articles stance, we do this with a two step process.

Classification pipeline

The classification process used in this coursework is a two step process. We first create models to categorise articles in the groups ‘unrelated’ or ‘related’, and then create models to classify if a related article is either ‘disagree’, ‘discuss’ or ‘agree’. This two step process allows us to combine two models to achieve the overall goal of classification among 4 classes.

The first task of classifying into ‘unrelated’ or ‘related’ begins with creating simple machine learning models as a baseline. We then further this by creating two deep learning models to achieve the same task and compare results. This set of models will have two distinct characteristics one set will be trained using TF-IDF embeddings and the other transformer embeddings. The transformer I chose in this coursework was BERT.

TF-IDF embeddings are produced through a non-learnt method, capturing both local and global term significance in a corpus while efficiently down weighting less informative common words, without needing complex analysis. However, its 'bag of words' approach ignores word context and order, limiting its effectiveness in many NLP tasks by failing to capture semantic meanings and relationships.

BERT embeddings are advanced text representations that capture semantic relationships between words and their context, improving on TF-IDF by understanding semantics and polysemy within sentences. Although more powerful, BERT requires substantial computational time for embedding a training set.

TF-IDF Embeddings:

To generate TF-IDF Embeddings I made use of the sklearn TfidfVectorizer a class which automatically performs TF-IDF Vectorisation on a dataset. The TfidfVectorizer used in this coursework was configured to fit training samples to 1024 features, it is also set to make training input lowercase and remove words with frequency less then 7 from the vocabulary.

BERT Embeddings:

As mentioned above I opted to use BERT embeddings as additional transformer embeddings, due to their benefits over TF-IDF embeddings. To facilitate this I used the hugging face library transformers and imported a BertTokenizer and a BertModel, both of these are 'bert-base-uncased' which is a smaller architecture. This was chosen to decrease computational overheads as for this task 'bert-base-uncased' is sufficient. Each output from the Bert model is a 768 long vector.

XGBoost:

The chosen baseline machine learning model was XGBoost. This model provides a great balance between computational efficiency and classification performance, it utilises a depth first approach to tree pruning which allows for better generalisation thus preventing overfitting making it more suitable for imbalanced datasets like this one.

The implementation of XGBoost I had a learning rate of 0.5 and 325 trees (n_estimators).

Table 1-XGBoost BERT Embeddings

	Precision	Recall	F1-Score	Support
Unrelated	0.992004	0.994728	0.993364	9105
Related	0.985727	0.978453	0.982088	3388
Accuracy			0.990315	
Macro average	0.988866	0.986591	0.987721	12493
Weighted average	0.990302	0.990315	0.990303	12493

Table 2-XGBoost TF-IDF

	Precision	Recall	F1-Score	Support
Unrelated	0.982298	0.993410	0.987823	9105
Related	0.981735	0.951889	0.966582	3388
Accuracy			0.982150	
Macro average	0.982017	0.972650	0.977202	12493
Weighted average	0.982145	0.982150	0.982062	12493

Deep Model for Unrelated/Related – LSTM

The deep learning architecture chosen for this task was LSTM. This non-traditional RNN makes can learn long term dependencies in sequence data making it a perfect architecture for this NLP project. A traditional RNN would have issues with vanishing or exploding gradients an LSTM can overcome this challenge with the use of gates which control the flow of information through the network dropping information over a long period of time. The particular implementation used in this project is a bi-directional LSTM, this makes use of two LSTM layers which run in parallel one processes start to end and the other end to start. This allows the model to have future context and past context making it considerably more powerful then a traditional LSTM.

Each model is trained with a CrossEntropyLoss function and an Adam optimizer with learning rate 0.001. The CrossEntropyLoss function is weighted with weights proportional to class frequency. This allows trained models to avoid overfitting on over-reperesented classes. The train and validation data is split into 75% train 25% validation.

Figure 4 - LSTM,TF-IDF Embeddings

	precision	recall	f1-score	support
Unrelated	0.944338	0.912403	0.928096	9167
Related	0.779153	0.851774	0.813847	3326
accuracy			0.896262	12493
macro avg	0.861745	0.882089	0.870971	12493
weighted avg	0.900361	0.896262	0.897679	12493

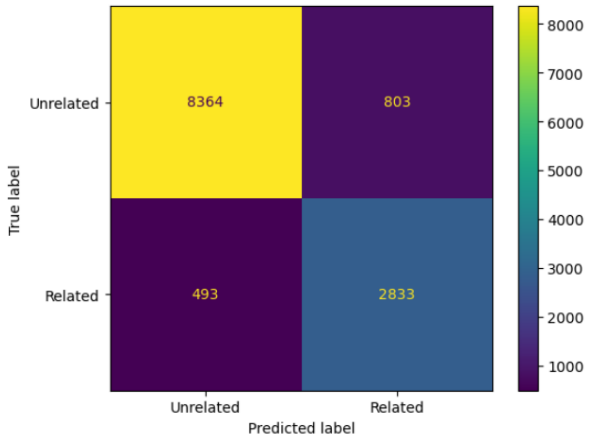


Figure 3- LSTM, BERT Embeddings

	precision	recall	f1-score	support
Unrelated	0.995037	0.984506	0.989744	9165
Related	0.958540	0.986478	0.972309	3328
accuracy			0.985032	12493
macro avg	0.976789	0.985492	0.981026	12493
weighted avg	0.985315	0.985032	0.985099	12493

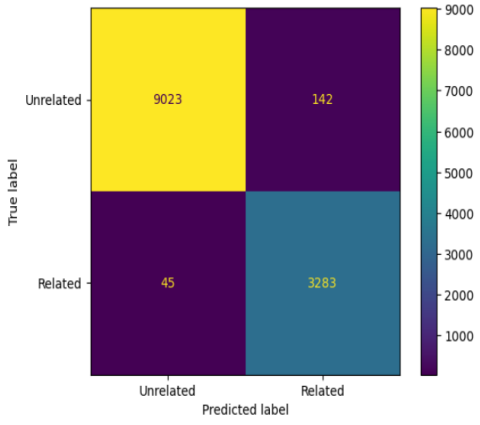
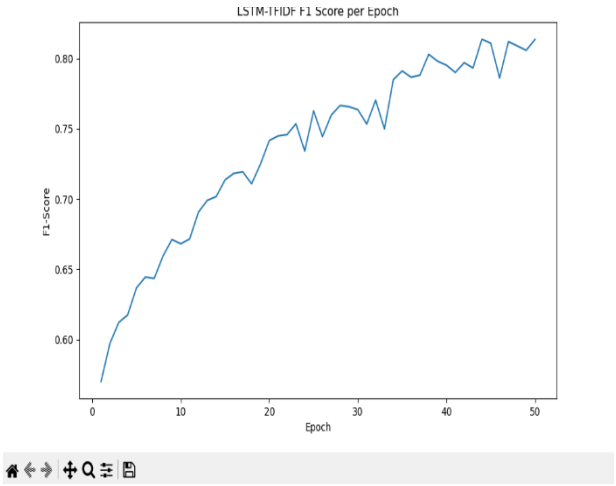
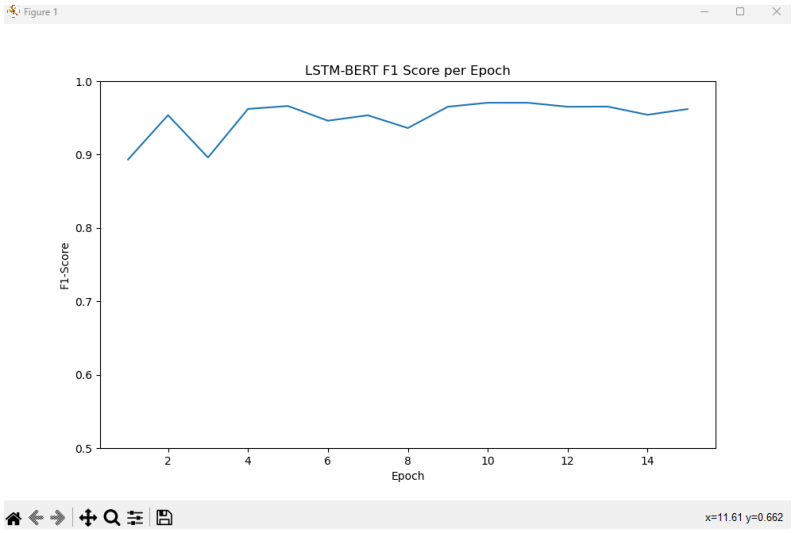


Figure 5- LSTM, BERT F1 Score per Epoch

Figure 6- LSTM, TF-IDF F1 Score per Epoch





LSTM Unrelated/Related Results

As seen in the above figures we can see that both models achieved significantly high accuracy on the validation set. However the model utilising BERT embeddings achieved higher classification performance across all metrics. It is also worth noting in from figure 5 and 6 how quickly the Bert embedded model took to converge when compared to TF-IDF. The BERT model was only trained for 15 epochs to avoid overfitting as you can see from figure 5, there may be some slight overfitting even with this short epoch amount. The increased performance with BERT is expected due to their ability to capture deep contextual representations to word meaning.

Agree/Disagree/Discuss Classification

The final model produced is designed to classify three classes Agree Disagree and Discuss, given a BERT embedding as input. For this I expanded on the previous bi-directional LSTM by adding some additional fully connected layers separated by ReLu activation functions.

The intention of this architecture was to expand on the successful previous model by adding more layers which would let the model learn a more complex distribution from the input data.

This model like previous made use of a weighted cross entropy loss function to overcome dataset imbalances.

Figure 7 - Agree/Disagree/Discuss Results

	precision	recall	f1-score	support
Disagree	0.304020	0.584541	0.400000	207
Discuss	0.856310	0.721304	0.783030	2239
Agree	0.513514	0.604830	0.555444	911
accuracy			0.681263	3357
macro avg	0.557948	0.636892	0.579491	3357
weighted avg	0.729229	0.681263	0.697651	3357

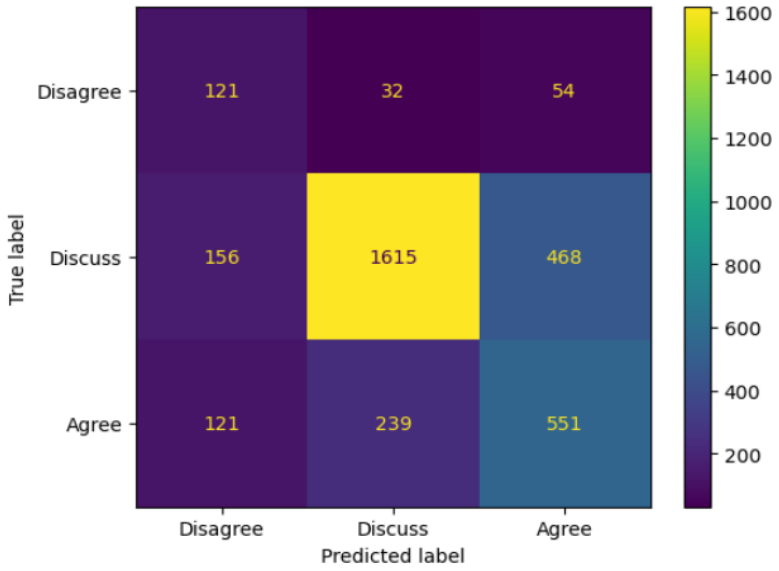


Table 8-ROC for Three Class Classifier

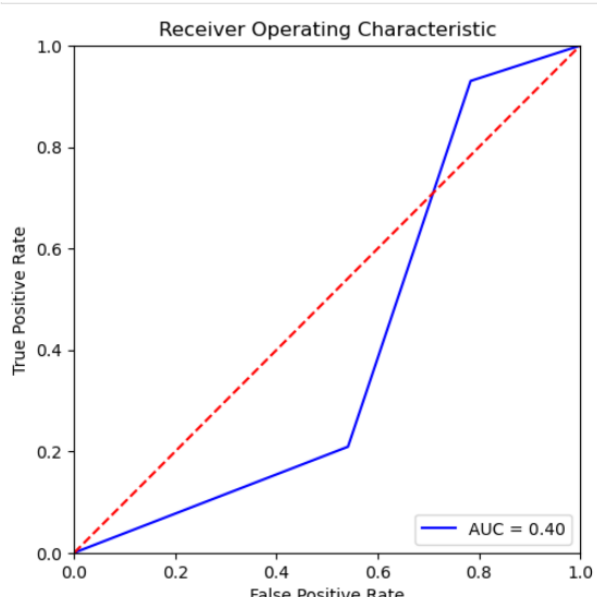


Figure 7 shows the results of the 3 class classifier on the validation dataset, while some of these accuracies are not too bad for a multi class classification problem we can see from the relatively high scores on the Discuss class that this model is overfitting 'Discuss' due to the dataset imbalance.

This is occurring despite the implementation of the weighted loss function. This can be happening for a number of reasons, mainly the model complexity. As this model implements additional fully connected layers, these layers are being tuned to predict towards 'Discuss' the over-represented class. This is further supported by figure 8 showing the ROC-curve and the AUC metric. As these metrics incorporate true positive against false positive, the low AUC score shows that the model is favouring over predicting a class 'Discuss'.

End to End Test:

The end to end test is done on the fake news challenge competition dataset, this is to avoid any possibility of trained data being validated on.

The first LSTM will be used to detect unrelated/related and the 3 class LSTM will be used to determine Disagree,Discuss and agree if the first detects related.

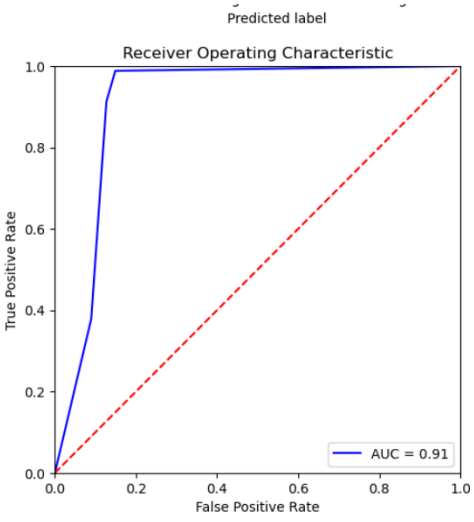


Figure 9- End to End ROC

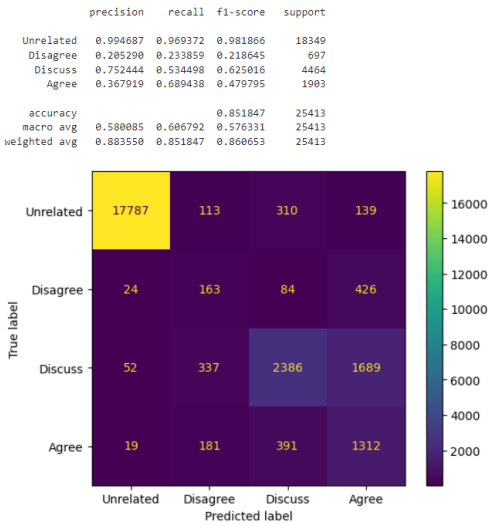


Figure 10- End To End Confusion Matrix

At first glance, it seems the end to end model is highly effective at four class classification, however when taking wider context into account we can see some floors. The high AUC score is observed due to how well the binary unrelated/related classifier performed. As the dataset imbalance favours unrelated datas the more effective first model is being used more frequently. If we inspect the confusion matrix in Figure 9 we can see this end to end system suffers when predicting ‘Agree’ and ‘Disagree’ Classes.

Further work can be done to improve the second 3 class classifier which will intern solve this issue. A solution to this could be implementing a CNN LSTM hybrid this hybrid architecture would provide spatial feature extraction from the CNN as well as sequential modelling from the LSTM making it a powerful model for text classification tasks like this one.

Ethical Considerations

With all machine learning applications there is a risk of bias within a model and depending on the model use case, this can have an impact on societal beliefs. As the models above utilise a pre-trained BERT model to embed data, we inherit some potential bias from the data-used in training the BERT model. BERT is trained using a vast corpus of text data. As biases such as gender bias, racial bias and socio-economical bias persist today, the wide scale adoption of text-based classification models like this one can perpetuate these biases through society.

The above models can be misused to discredit potentially reliable sources. As mentioned previously, the datasets used in training our models likely contain biases of many forms. This could be a model like ours predicting unrelated on a news article which contains underrepresented groups. As news is such an important factor in today’s society, this perpetuation of bias could have devastating consequences.