# Introduction

Fake news poses a substantial problem with disinformation on the rise in recent years, especially on social media platforms[cite]. Attempting to fact check by hand is a labour-intensive process.

# Problem Definition

The Fake News Challenge (FNC) dataset consists of pairs of headlines and articles classified as either unrelated, discuss, agree, or disagree. The goal is to train a model to classify pairs correctly by using a two-stage classification approach.

The training data consists of X pairs and the distribution of classes in the training data is severely imbalanced as evidenced in Figure \ref{fig:dist}. I split this into a training and validation set using an 80\%-20\% split. The test data is similarly imbalanced and consists of Y pairs.

[BAR CHARTS OF THE DISTRIBUTION]

# Proposed Solutions

I use a two-step classification process. The first step is to classify headlines and article bodies as either unrelated or related. Related pairs are then classified further into either discuss, agree, or disagree.

## Feature Extraction

Firstly, I process the pairs into features for use with machine learning. I apply data cleaning by stripping URLs, emojis and numbers.

For TF-IDF features, I also lowercase the text. TF-IDF is a bag of words approach. Firstly, a vocabulary of words from all training documents is extracted. It then creates a term frequency vector per document containing the frequency of each word from the vocabulary in the document. These are normalised by the inverse document frequency that assigns greater weighting to rare words under the assumption they convey more unique information about a document.

It is quick and easy to compute, and it can be used to compare documents using cosine similarity applied to the vectors. However, as it is a bag of words model, we lost information about the order which is valuable. Furthermore, the vocabulary is limited to the training data which will pose problems with unseen words.

Using HuggingFace[cite], I extract DistilRoBERTa[cite] features. Unlike TF-IDF, a vocabulary of sub-words are created using Byte-Pair Encoding[cite] from GPT[cite]. Bytes from the training data are repeatedly merged based on their frequency within the training data to create a vocabulary of most common byte sequences until a pre-defined number of sequences is reached. This is then used to tokenize any data by segmenting words into sequences of these extracted bytes.

Tokenization is the first step. Following this, the tokens are processed by the DistilRoBERTa model and I extract the CLS token for each pair to use as the features for the standard ML models. The additional hidden states can also be used for more advanced models such as CNNs and RNNs.

One advantage of transformer features is the solution to the out of vocabulary problem. They are much better equipped to deal with unseen words in the test data than TF-IDF. Furthermore, unlike Word2Vec[CITE], the embeddings are not fixed per word but rather consider their context which is superior to bag of words approaches as they retain this information.

They do have some disadvantages. DistilRoBERTa has a 512 token limit as such it is not possible to directly encode all the entire headline and article body, instead they have to be truncated. Furthermore, they are computational expensive due to the matrix multiplications required for attention[cite].

## Dataset Imbalance

Due to the imbalance of the data, using accuracy to evaluate the models in not appropriate as it does not account for the size of each class. Instead, I utilise weighted F1-score, Matthew’s Correlation Coefficient and Area under the Receiver Operating Characteristic Curve to account for true positives and false positives in greater detail.

Furthermore, I implement a weighted sampler to enable over and under sampling of specific classes to boost the minority classes where appropriate to ensure the model has sufficient exposure for training.

## Unrelated-Related Classification

I utilise standard machine learning models from sklearn and LightGBM.

For TF-IDF, I will compare the efficacy of Complement Naïve Bayes[cite] and Gradient Boost[cite]. I select these because they support options to deal with the dataset imbalance.

For transformers, I will again use Gradient Boost to attempt to classify the pairs.

Following standard machine learning models, I will also use deep learning approaches for both types of features.

For TF-IDF, I will use a fully connected network. This is because it is a bag-of-words approach and as such it does not possess the structure needed for exploitation via, for example, an RNN the features are not connected temporally.

For transformers, I will utilise a classification head attached to the transformer. I also use this setup to investigate the effects of freezing the transformer on classification performance. This will influence the decisions made in the second stage of classification.

## Stance Detection

For the second stage of classification, I will use the features from the transformer solely and implement deep learning techniques. I propose the use of either an LSTM or a CNN and LSTM hybrid approach. I will compare these. Furthermore, due to the imbalance being so severe, I will apply sampling techniques to evaluate the effect of the dataset imbalance. Due to the success of finetuning the transformer, I will only use finetuning instead of freezing the transformer.

# Analysis of Results

## Unrelated-Related Classification

I find that in TF-IDF features perform poorly when applying standard machine learning techniques as shown in Table[X]. For example, Complement Naïve Bayes has a very poor MCC score of 0.086 on the test set and an AUC of only slightly better than random. On the validation set it does not perform that much better.

Gradient Boost performs well on the validation set suggesting that it has been able to model the data well. It achieves a high F1, MCC, and AUC score as well as impressive accuracy. However, this quickly falls off on the test set highlighting the out-of-vocabulary disadvantage discussed previously. As it is unable to represent the data it struggles to deal with the unseen data. Furthermore, when applying weighting to the minority ‘related’ class, the performance does not drastically change but it does perform slightly worse.

Transformer-based features also struggle in the standard machine learning setup as shown in Table[X]. Gradient Boost performs worse on the transformer-based features than with the TF-IDF on the validation scoring lower in all metrics by a substantial margin. However, unlike TF-IDF features, there is an improvement on the test dataset. It does not suffer as badly as TF-IDF because the solution to the out-of-vocabulary problem is substantially better.

For deep learning models, the fully connected network for the TF-IDF features achieves slightly worse results (as shown in Table[X]) than Gradient Boost but nonetheless they are high quality on the validation set with a slightly higher AUC but worse F1 and MCC. However, the drop in performance between the validation and test set is massive. The accuracy and F1 over half and the MCC falls drastically although this is in the same range as the Gradient Boost model. This suggests that it may have overfitted to the training data distribution that the validation set is derived from.

The most impressive results come from using the transformer features with deep learning models as shown in Table[X]. I find that by unfreezing the transformer the performance with both the CNN and classification head are excellent. The performance on both the validation and test sets are very similar and there is not substantially drop off in performance. They achieve near perfect F1, MCC, AUC, and accuracy scores suggesting high quality performance.

When freezing the model, the performance drops drastically suggesting that finetuning the transformer is essential. Without weighted sampling, I find that the frozen transformer severely struggles to classify the minority class as evidenced by the [METRIC].

## Stance Detection

Firstly, I trial the use of an LSTM with sample weighting to oversample the minority ‘disagree’ class. From testing, I find that also amplifying the ‘agree’ class leads to poor performance. This approach is successful and achieves an F1-score of 0.799 suggesting that it is able to classify accurately across all classes on the validation dataset. However, this decreases on the test set like in the unrelated-related classification stage.

I also utilise a hybrid CNN and LSTM model. I find that the performance of combining these models is superior to using the LSTM on its own. This is because the CNN provides local enhancements to the features from the transformer prior to them being used by the LSTM model. I compare two different weighting regimes here. With equal weighting for all classes, I find that the performance is slightly worse than LSTM with disagree oversampling suggesting that equal weighting is not effective. This is to be expected because there are more examples in the discuss class than the disagree class so by equal sampling we expose the model to a larger distribution of the majority class encouraging it to learn the features of this distribution easier.

I find the most effective approach is to oversample the minority class by amplifying its weighting and undersampling from both the agree and discuss classes. This approach with the hybrid model leads to an F1-score of 0.853 on the validation set and 0.741 on the test set. All of the metrics for the hybrid model using this weighting regime are the highest seen for the stance detection models. As such, this is the most effective.

## End-to-End Classification

I combine the finetuned DistilRoBERTa with an MLP Head model from the unrelated-related classification step with the CNN+LSTM Hybrid with amplified disagree weighting for the end-to-end model. I find that the performance on the validation dataset is excellent with a weighted F1-score of 0.971 suggesting that the model has high precision and recall thus most pairs are classified correctly. It also achieves a high MCC further reinforcing this.

On the test dataset, the performance is worse. There are a greater number of false positives and this is reflected in the F1 score decreasing to 0.912 and further observed via the substantial decrease in MCC compared to the validation set. This can also be seen clearly from the confusion matrices in Figure \ref{fig:end\_cm}. A drop off is expected on the unseen test dataset however.

Of course, the performance of the end-to-end model is worse than the two parts separately when training. This is expected because previously I was manually separating out the unrelated examples for testing the stance classification on its own. As such, when the first model is used to separate out the unrelated examples, the error from this will propagate to the second model. Hence it was essential that the unrelated-related classifier performed well.

# Discussion

It is clear that transformer features with deep learning models are vastly superior compared to other approaches seen here. This is unsurprising, the use of attention and large-scale pre-trained language models have revolutionised NLP[cite]. For example, the best model from the original fake news challenge was [MODEL] which achieved an F1 score of [VALUE] [COMPARISON].

With greater resources, the results could be further improved by using a transformer with greater maximum token lengths such as Longformer[cite] to better support the article lengths.

# Ethical Implications

Care needs to be taken, especially with fake news, as by automating fake news detection it may unintentionally suppress legitimate viewpoints under the guise of fake news if they are a false positive.

Fake news has been a significant problem in US Presidential Elections[cite], as such suppression of political articles could negatively impact elections. For example, Twitter utilises machine learning for personalisation of home feeds and internal research showed that this introduced bias against left-wing political parties[cite].

In addition to accidental and intentional misuse, due to the dataset imbalance this leads to biasing of the model. By amplifying the disagree class, this has led to a trade-off between the overall accuracy and the accuracy of the minority class. Even then, this was not sufficient to fully resolve issues with the minority class as such disagree articles are likely to be misclassified indicating the model is biased against the class.

# Conclusion

In conclusion, I find that the two-step classification approach is effective and transformer-based approaches offer superior performance as expected.