# Introduction

# Problem Definition

The Fake News Challenge (FNC) presents a dataset consisting of headlines and article bodies. These are associated in a many-to-many relationship and each pair is classified as one of either: unrelated, discuss, agree, or disagree.

The training data (train\_stances.csv and train\_bodies.csv) consists of X entries, I split this into a training set and validation where the validation set is 20% of the original data with care taken to ensure the article bodies are disjoint between sets following [CITE]. For testing, I use the provided test data (competition\_test\_stances.csv and competition\_test\_bodies.csv). The testing data consists of X entries.

There is severe dataset imbalance. The ratio of unrelated to *related* (discuss/agree/disagree) is 3 to 1 and after excluding the unrelated entries the imbalance is further amplified with discuss, agree, and disagree in the ratio 11:5:1. This presents significant issues for classifying the pairs correctly. Analysis of all three datasets reveals that the distribution is similar.

[FIGURE OF DISTRIBUTION ON TEST?TRAIN? SET]

To approach this problem, we split the problem into a two-step classification problem. Firstly, we train a model to classify pairs into either unrelated or related. Following this, we utilise only those from the first step that were classified as related and further classify these into discuss, agree, or disagree. I will then select the two most successful models to evaluate the classification procedure end-to-end.

# Proposed Solutions

## Feature Extraction

I utilise both TF-IDF features and transformer features, specifically from DistilRoBERTa provided by HuggingFace (as well as DistilBERT for comparison later on).

TF-IDF firstly creates a vocabulary of words. 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.

TF-IDF is useful because it is easy to compute, and it is easy to compare documents by using cosine similarity between the vectors. However, it is a bag of words model, so we lose information about the order of the tokens which is valuable information. These vectors are also high-dimensional and sparse which is expensive for computation.

I extract features using both DistilBERT and DistilRoBERTa from HuggingFace to compare their efficacy later on. Instead of a vocabulary of words, BERT uses sub-word segmentation called WordPiece. This breaks words down into pieces based on their value to the language model. RoBERTa uses… A tokenizer is used to convert the text to token IDs (and extract attention masks) which I pass to the transformer to extract [CLS] tokens for use later on.

Transformers work by applying attention… [see slides]

The advantages of using transformers are… the disadvantages are…

## Dataset Imbalance and Metrics

Due to the imbalance of the dataset, greater consideration needs to be given to the training and evaluation process of the models.

To train the models, I implement a sampler that is compatible with PyTorch’s [cite] data loader that enables sampling distributions to be used to over and under sample specific classes. This can be used to reduce the boost the minority classes to prevent the other classes dominating training. [LOSS FNS?]

It is also important to note that due to the imbalance metrics will need to be weighted to account for the number of samples from each class. For example, overall accuracy is not a useful metric when the classes are severely imbalanced. Consider the unrelated-related scenario, if a model classifies all samples as unrelated then it will achieve accuracy of approximately 75% despite the model’s output being meaningless. As such, I utilise the weighted F1 score, Matthew’s Correlation Coefficient, and Area under the Receiver Operating Characteristic Curve. These metrics are substantially better as they account for [WHY?]

## Standard Machine Learning Models

For the unrelated/related classification problem, I utilise a variety of different machine learning models to compare their efficacy. These are trained separately on both types of features discussed. For TF-IDF I compare [MODELS 1] and for transformers I compare [MODELS 2]. I use less models for the transformers due to the requirement for non-negativity for [MODELS 3].

I do not use standard machine learning models for the second stage of classification.

## Deep Learning Models

For TF-IDF features, I propose the use of a fully connected network consisting of X layers using ReLU activations and [NORMALISATION]. This reduces the features gradually until the final layer where pairs are classified into either related or unrelated using softmax classification according to the logits from the final layer. I utilise a fully connected network rather than other approaches such as a CNN or RNN because TF-IDF is a bag-of-words-based model as such it does not possess the structure required that other deep learning-based approaches can exploit.

For the transformer features, I propose, implement, and compare a variety of different approaches to compare their efficacy.

Furthermore, I also compare these models when the transformer is frozen versus trainable. I also compare DistilRoBERTa to DistilBERT to investigate if there is a substantial difference in performance.

# Analysis of Results

## Unrelated/Related Classification

### TF-IDF

I find that TF-IDF features struggle to achieve high quality results on both the standard machine learning and deep learning models. [TABLE] shows the results of the solutions and metrics outlined previously on both the validation and test data. Generally, all models struggle to classify correctly – especially on the unseen test data.

[TABLE]

As expected, Naïve Bayes performs poorly with the lowest F1-score on the validation set. Surprisingly, Complement Bayes does not gain in performance despite being generally better for imbalanced data [CITE] and instead achieves the lowest F1-score on the test set. Decision Trees and Random Forest suffer similarly and tend to suffer greater overfitting to the training data as evidenced by the greater drop off on the test set.

Gradient Boost performs the best but there is substantial difference between the validation and test sets, even when accounting for the imbalance by weighting the classes. This highlights a significant problem with the TF-IDF approach, because the test set contains words that will be outside of the vocabulary these are poorly represented by the TF-IDF features. As such, the test performance will always suffer without a substantially larger corpus of training data.

When using the fully connected network, the problem exposed by Gradient Boost is seen again. The fully connected network performs very similar to Gradient Boost on the validation set but suffers extremely on the test data.

This suggests that TF-IDF features may not be suitable for this task.

### Transformers

Transformer features perform substantially better. With the standard machine learning models they still suffer poor performance but the test performance is improved further confirming the suspicions that the issues of TF-IDF vocabulary are problematic. These problems are alleviated with transformers as their sub-word segmentation approach to the out-of-vocabulary problem is superior. Table [X] shows the results for the DistilRoBERTa CLS tokens used to train standard ML models:

[TABLE]

Again, Gradient Boost is the best performing standard machine learning model. While I conduct less experiments here due to the nature of the standard machine learning models not supporting negative values, it is clear that the test set performance is better at the cost of worse validation performance. [EXPAND]

More interestingly, the results for the deep learning models with the transformers are excellent on both the validation and testing sets. Table [Y] shows the results of finetuning approaches on both DistilRoBERTa and DistilBERT. Table [Z] shows the results of the same approaches but with the transformer frozen.

[TABLES]

It is immediately clear that finetuning the models – even without any class weighting – leads to incredibly high-performance classification on both the validation and test set. This applies to both using an LSTM and using a classification head. [EXPAND]

Comparatively, without finetuning, even with class weighting, the results are poor. [EXPAND]

I find that DistilBERT and DistilRoBERTa…

## Stance Classification

For stance classification, I utilise only the DistilRoBERTa features due to time and resource considerations. As previously proposed, I experiment with a classification head, CNN, LSTM, GRU, and a hybrid CNN-[RNN BASED ON BEST]. Following the results of unrelated/related it is clear that finetuning the transformer at the same time is essential to achieving the best results, as such I follow this approach.

[TABLE]

I find that…

## End-to-End Two-Step Classification

Following the analysis of each stage individually, I utilise the [FEATURES] and use [MODEL] for the first step and [MODEL] for the second step. Instead of filtering the data directly to test the classification I use the results from the first step directly with the second step to evaluate classification end-to-end.

# Discussion

It is clear that using transformers 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 has revolutionised natural language processing [CITE] and as such I would expect their application to the FNC to follow a similar trend. For example, the best model from the FNC was [MODEL] which achieved an [METRIC] of [VALUE] using [TECHNIQUE] [COMPARISON].

# Ethical Implications

Due to the large dataset imbalance, I employ techniques to reduce the impact of the class sizes on the model. Without this, the model would be heavily biased towards the Unrelated class in part 2.a and the Discuss class in part 2.b.

Care needs to be taken especially with respect to fake news because people may trust the machine learning models to correctly classify the relationship between the headlines and bodies without caring about the potential for false positives. As such, if someone with an ulterior motive, say political, publicised a model with knowingly poor classification (e.g., over classifying discuss instead of agree/disagree) people may take the information in the articles as fact without checking themselves because a model has classified it as such.

# Conclusion