Fake News Detection Using TF-IDF, BERT, Random Forest Classification and Hierarchical LSTM Models

I. INTRODUCTION

Classification within the domain of natural language processing (NLP) assigning labels to text. This study utilises popular NLP techniques to classify newspaper articles based on the claim within their headline. A comparison was carried out with TF-IDF and BERT when implemented with random forest classification (RandF) and an LSTM model respectively, to classify whether titles and bodies are related or unrelated. An additional trained multi-class LSTM was implemented on top to create an end-to-end hierarchical model.

NLP classification tasks can be solved through utilising a variety of language and artificial intelligence models, assigning probabilities to arbitrary sequences of word tokens to form predictions. Text pre-processing must be carried out first to standardize data input for further processing. This typically involves normalization to create uniform text (which may be all lowercase and without punctuation), tokenization to split text into meaningful token parts and lemmatization to reduce word tokens to their root form. At this stage global data dependencies can be captured, such as through feed forward layers with transformers. Popular models include machine learning (ML) classification, using simple algorithms on labelled text data. Deep learning (DL) can understand features through leveraging neural network architectures, including RNNs to capture temporal information from sequential data, CNNs to understand combinations in local information and LSTMs to store short term memory based on long term dependencies. Performance of these models can be improved, such as by considering attention to focus on the relevant part of the long sequences of input data.

II. PROBLEM DEFINITION

A. The Problem

This paper addresses the task of classifying news articles to articles to assess whether they are fake.

This problem requires input of a headline and a body of text to be analysed by a model. This can then assign a class: related, with subclasses of agrees, disagrees and discusses, or unrelated.

B. Data Description

Fig 1. *The word cloud of "articleBody" once tokenized, lemmatized and stopwords were removed.*

	Headline	articleBody	Stance
0	Police find mass graves with at least '15 bodi...	Danny Boyle is directing the untitled film\r\n...	unrelated
1	Seth Rogen to Play Apple's Steve Wozniak	Danny Boyle is directing the untitled film\r\n...	discuss
2	Mexico police find mass grave near site 43 stu...	Danny Boyle is directing the untitled film\r\n...	unrelated
3	Mexico Says Missing Students Not Found In Firs...	Danny Boyle is directing the untitled film\r\n...	unrelated
4	New iOS 8 bug can delete all of your iCloud do...	Danny Boyle is directing the untitled film\r\n...	unrelated

Fig 2. *The head of the merged train_bodies.csv and train_stances.csv files.*

The data used was the FNC dataset [1], which includes data regarding "Headline", "articleBody" (*Fig 1*), "Stance" and "Body ID", as seen in *Fig 2*. There are nearly 50,000 pairs of headline and article bodies.

C. Challenges

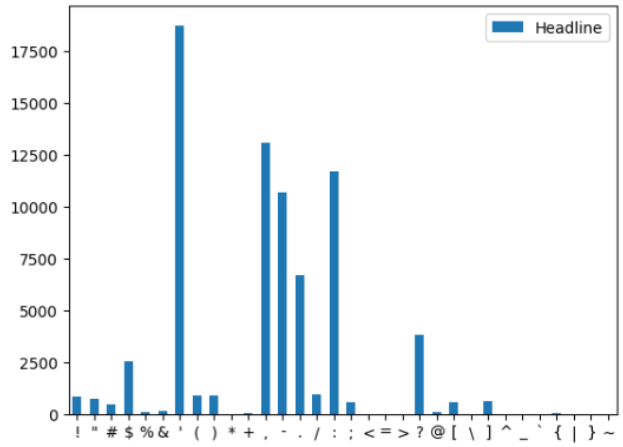

Fig 3. *The unprocessed "Headline" data punctuation count.*

The initial dataset includes a high proportion of punctuation (*Fig 3*), stopwords and mistranslated characters that are included in the csv files. To analyse the words within the "Headlines" and "articleBodies" columns appropriately, these must be removed.

The "Stance" data appears to classify 70% of data as "unrelated". This noise results in difficulty in assessing writing intent. Hence this imbalance can result in biased model training, resulting in poor classifier performance on the minority classes. This would reduce the generalisation of the model to input of data in the real world.

The multi-data input produces a challenge related to information fusion similarly, since it is difficult discerning subtle nuances and contextual clues to make a single decision. Additionally, the increased dimensionality of data can become an issue when training due to the increased likelihood of overfitting, and as more data is usually required to train such models appropriately.

III. PROPORSED SOLUTIONS AND REASONS FOR CHOICES

A. Solution Formulation

To tackle the main task, a pipeline can be created to classify if a headline and body pair are related or unrelated, and then classify related pairs and if they agree, disagree or discuss the headline. This can be completed appropriately through understanding the context of words within the heading and body. This pipeline can assist in tackling the data imbalance issue since the second model can specialise in and prioritise understanding nuances between subclass categorisations.

B. Path to the Solution

To achieve such a solution, data was pre-processed. This overcame the first challenge, regarding the content of the data in the dataset. The original train files were merged and inspected. Apostrophes were removed and all other punctuation and new lines were replaced by spaces to assist in the creation of word tokens; text was also converted to lowercase. The strings were lemmatized and stopwords were removed following this to reduce the search space.

TF-IDF and the BERT transformer were implemented for further processing. TF-IDF creates a distribution to measure the importance of a word in a document based on use and frequency. BERT is contextual language model which only requires an encoder part. [2] It was chosen as it can aptly understand the semantic relationship between the inputs as it was pre-trained on a large corpus of data.

Model training was completed on the full ~50,000 pairs. The RandF ML model and an LSTM DL model were implemented for binary classification of related or unrelated.

RandF was implemented since it can handle high-dimensional data well. It assesses the frequency and presence of tokens in a tree-based manner, and so can handle imbalanced data well. To address the challenge of an increased dimensionality, PCA hyperparameter tuning was included here.

LSTMs were implemented since they are most suitable type of DL model when attempting to understand the usage of words / the long-term dependencies of sequential tokens, which is necessary for this problem. They are commonly used for text classification [3]. LSTMs improve upon RNNs in avoiding vanishing gradients, as well as GRUs in using memory in addition to gates. CNNs focus on short-term differences, which are not as applicable to this task.

Similarly, an LSTM model (multi-LSTM) was most appropriate for the multi-class classification of agree, disagree or discuss for the related class.

Finally, a pipeline, including the best classification model and the multi-LSTM, was created as a solution to the original problem.

The merged training set was split into a training, validation and test set. The final pipeline was tested on the competition test data.

IV. ANALYSIS OF RESULTS

A. Evaluation Metrics' Results

a)

articleBody
Money makes the world go round, right?\r\n\r\n...
North Dakota voted to name a new 650-acre publ...
It was a heartwarming story for legions of pet...

b)

articleBody
money makes the world go round right luncht...
north dakota voted to name a new 650 acre publ...
it was a heartwarming story for legions of pet...

c)

articleBody
money world round right lunchtime es...
north dakota vote new 650 acre publicly site c...
heartwarming story legion pet owner animal lov...

Fig 4. *The stages of pre-processing from (a) the original data, (b) the lemmatized data and (c) the data without the stopwords.*

Model	Class-ification	Precision	Recall	F1-Score
RandF + TF-IDF	0 (Unrelated)	0.95	0.98	0.97
	1 (Related)	0.94	0.86	0.90
RandF + BERT	0 (Unrelated)	0.73	1.00	0.85
	1 (Related)	0.00	0.00	0.00
LSTM + TF-IDF	0 (Unrelated)	0.80	0.97	0.88
	1 (Related)	0.82	0.32	0.46
LSTM + BERT	0 (Unrelated)	0.73	1.00	0.85
	1 (Related)	0.00	0.00	0.00

Fig 5. *The precision, recall and F1-scores for the binary classification models, on the test set.*

Model	Accuracy	Precision	Recall	F1-Score
RandF + TF-IDF	Accuracy	//	//	0.95
	Macro Average	0.95	0.92	0.93
	Weighted Average	0.95	0.95	0.95
RandF + BERT	Accuracy	//	//	0.73
	Macro Average	0.37	0.50	0.42
	Weighted Average	0.54	0.73	0.62
LSTM + TF-IDF	Accuracy	//	//	0.80
	Macro Average	0.81	0.65	0.67
	Weighted Average	0.80	0.80	0.77
LSTM + BERT	Accuracy	//	//	0.95
	Macro Average	0.37	0.50	0.42
	Weighted Average	0.54	0.73	0.62

Fig 6. *The accuracy, macro average and weighted average results for binary classification on the test set.*

Model	Confusion Matrix
RandF + TF-IDF	[[10772 203] [550 3467]]
RandF + BERT	[[10975 0] [4017 0]]
LSTM + TF-IDF	[[10689 286] [2722 1295]]
LSTM + BERT	[[10979 0] [4013 0]]

Fig 7. *The confusion matrices for binary classification on the test set.*

Figs 5-7 demonstrate the results of the binary classification. The TF-IDF feature extraction generally works well at capturing relevant patterns. The loss in the LSTM model with TF-IDF was 0.14 less than with BERT. BERT fails to predict any "related" classes and instead classifies all input pairs as "unrelated".

The RandF model performs exceptionally well achieving an accuracy of 0.95, whereas the LSTM had difficulty in classifying the "related" data as seen by a recall of 0.32.

Model	Class-ification	Precision	Recall	F1-Score
Multi-LSTM + TF-IDF	0 (Agree)	0.69	0.64	0.67
	1 (Disagree)	0.53	0.15	0.24
	2 (Discuss)	0.84	0.92	0.88
[LSTM + BERT] + [Multi-LSTM + TF-IDF]	0 (Agree)	0.00	0.00	0.00
	1 (Disagree)	0.00	0.00	0.00
	2 (Discuss)	0.00	0.00	0.00
	3 (Unrelated)	0.72	1.00	0.84

Fig 8. *The precision, recall and F1-scores for the three-classification model and the final pipeline, on the test set.*

Model	Accuracy	Precision	Recall	F1-Score
Multi-LSTM + TF-IDF	Accuracy	//	//	0.80
	Macro Average	0.69	0.57	0.59
	Weighted Average	0.78	0.80	0.78
[LSTM + BERT] + [Multi-LSTM + TF-IDF]	Accuracy	//	//	0.72
	Macro Average	0.18	0.25	0.21
	Weighted Average	0.52	0.72	0.60

Fig 9. *The accuracy, macro average and weighted average results for the three-classification model and the final pipeline, on the test set.*

Figs 8-11 show the results of the multi-class LSTM model and the final pipeline. The multi-LSTM model has difficulty in classifying the "disagree" class (recall of 0.15 and an F1-score of 0.24). However it classifies the "discuss" class very well (recall of 0.92, F1-score of 0.88).

The final pipeline faces the same BERT issue as previously, seen in *Fig 11*.

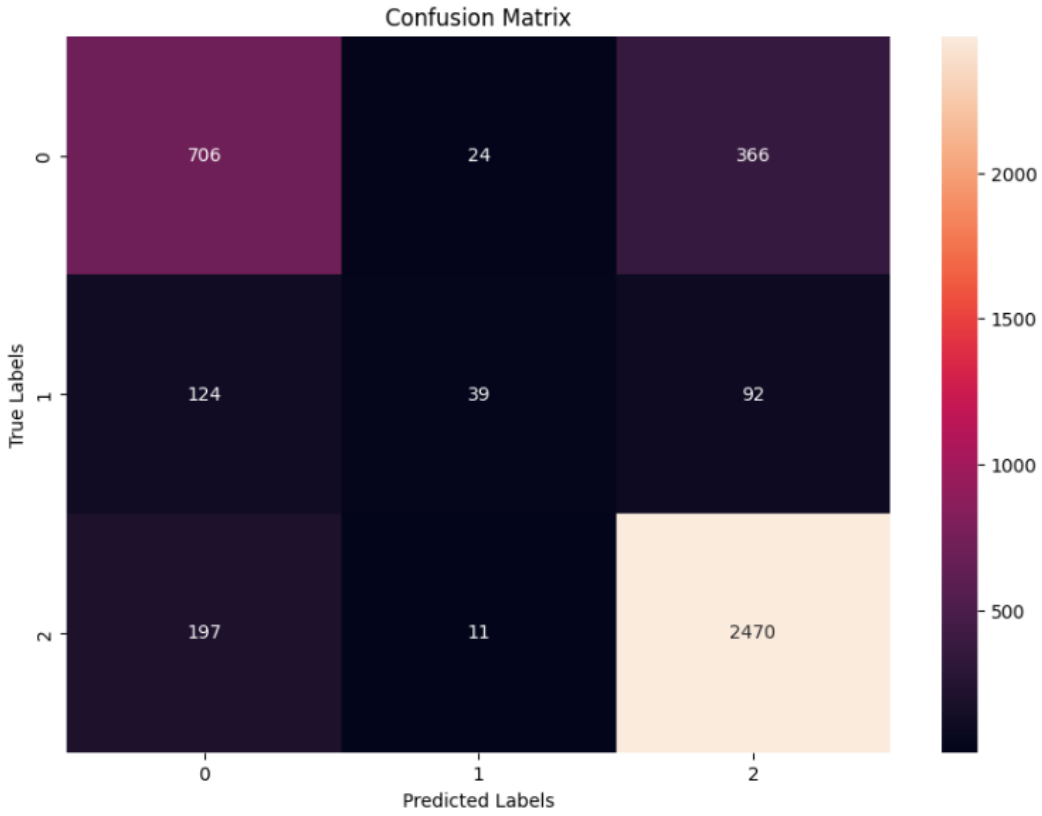

Fig 10. *The confusion matrix for multi-LSTM with TF-IDF for three-classification on the test set.*

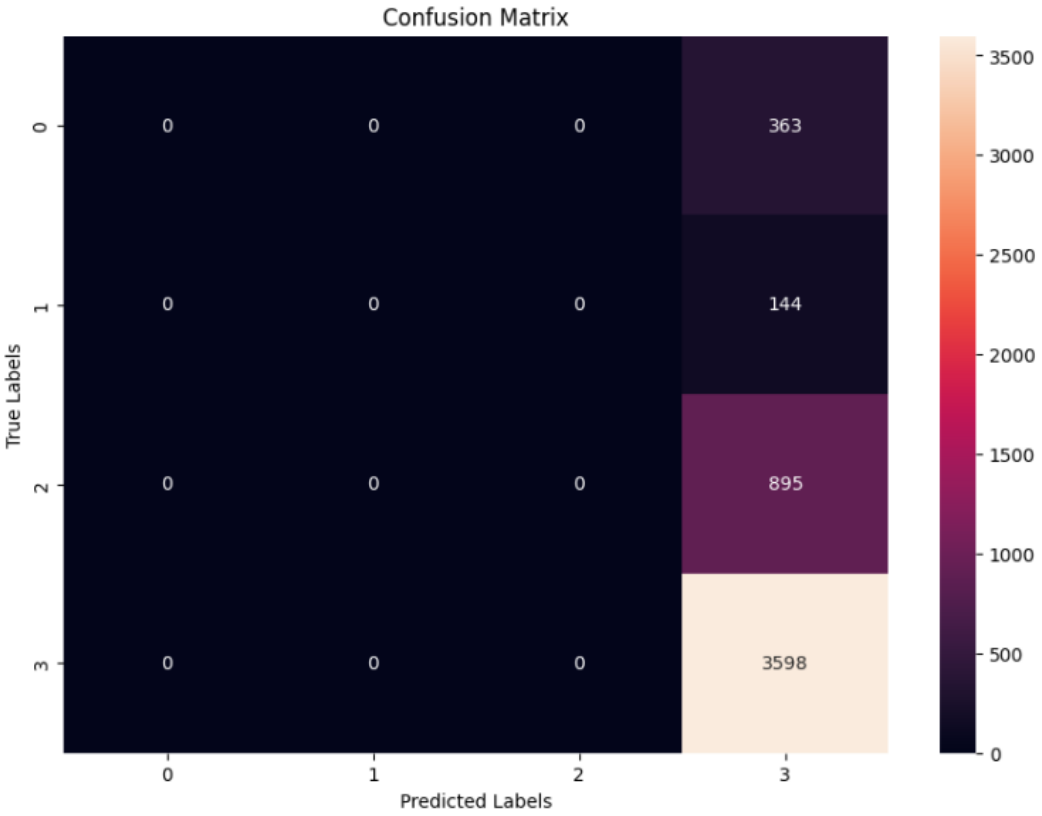

Fig 11. *The confusion matrix for the entire pipeline (LSTM with BERT and multi-LSTM with TF-IDF) on the competition test set.*

V. ANALYSIS INTERPRETATION AND DISCUSSION

A. TF-IDF and BERT

The results show that TF-IDF outperformed BERT when processing the data, as BERT resulted in only predictions of the “unrelated” class. This demonstrates that BERT was not fine-tuned enough to the data imbalance.

TF-IDF allows either two separate representations to be made, or for inputs to be merged. The latter option was chosen, and this worked well with RandF.

Theoretically BERT is more accommodating to dual inputs, as it assigns special tokens to separate inputs. BERT however has a token limit and so disregarded many generated sequences. Additionally, BERT is less interpretable, as cosine similarly can be used directly with TF-IDF.

B. Machine Learning with TF-IDF/BERT

TF-IDF worked well with the ML model as unnecessary sequential data is not assumed with RandF, unlike BERT. The dimensionality is simply reduced with the sparse vector representation.

C. Deep Learning with TF-IDF/BERT

TF-IDF worked well with the DL model. Theoretically, BERT is more suited to the LSTM model since they both handle semantic sequential data.

D. Machine Learning and Deep Learning

The ML model performed exceptionally. This is due to the strength of the tree-based model in handling noisy features and since TF-IDFs works well with ML.

E. Overall Solution Performance

The final model also performed poorly since BERT was utilised. Multi-task learning could have been implemented to train the model in one go and to pay attention to nuances between the distinct inputs.

VI. ETHICAL IMPLICATIONS

Bias within a dataset can filter down, particularly in the use of text with social biases. This system could miscategorise news articles due to learning wrong assumption and connections between tokens. To avoid such broad prototypes, further detail could be considered, such as through balancing the dataset with further “related” cases. Or data of “unrelated” cases could be augmented further to replace particular common words which lead to social or opinion bias in text, such as male or female.

These two main forms of bias could affect the system – and this can be amplified by the model. This is an ethical concern, as particular topics could be categorised as real news when this is not the case. This could be spread around, potentially harming others and leading to misinformation.

Finally, there is an issue regarding the fairness of the dataset, particularly that 70% of articles are labelled as “unrelated”, as biases may arise in the neglection of the subclasses. However, our pipeline and the methods chosen may assist in mitigating this, as more computation is dedicated to those minority classes.

VII. CONCLUSION

In conclusion, a pipeline with two distinct LSTM models was created to categorise news articles. TF-IDF improved performance and worked well with RandF, whereas BERT resulted in limited models.

REFERENCES

[1] FakeNewsChallenge, FakeNewsChallenge/FNC-1, GitHub. https://github.com/FakeNewsChallenge/fnc-1

[2] Sturgeon, D. and Al-Moubayed, N., 2023, NLP Lectures.

[3] Shekhar, S. (2023) What is LSTM for text classification?, Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/06/lstm-for-text-classification/