Detecting Deception: An NLP Pipeline for Fake News Classification

Abstract

Online misinformation has led to the spread of fake news which has become a problem that is worth addressing automatically. This project entails constructing a natural language processing pipeline that classifies news articles as real or fake using supervised learning methods. Our starting point is classical machine learning algorithms by means of TF-IDF representations and logistic regression, and we proceed to a deep learning model of a bidirectional LSTM to extract contextual and sequential information in text. Another major concern of this work is on data quality and preprocessing because our first experiments discovered that the data leaked significantly distorting performance artificially. We then set up realistic baselines and measure the models based on accuracy, precision, recall, F1-score and confusion matrices after cleaning up and restructuring our dataset. We also examine the model behavior through the explainability techniques to gain insight into the linguistic signs that affect predictions. Our findings show that contextual modeling is better than lexical baselines as well as indicating limitations due to dataset bias, stylistic cues and generalization.

1. Problem and Motivation

The advent of the internet news medium and social media has increased the pace of dissemination of information and the magnitude of information dissemination drastically. On the one hand, this democratization of information has numerous positive aspects, although on the other hand, it has facilitated the popularization of misinformation and fake news. There is the potential that fake news can affect political results, worsen the situation of a public health crisis, and undermine the credibility of legitimate journalism. Although manual fact checking works, it has not been able to keep up with the amount of content that is created on a daily basis, which has led to the desire to have automated fake news detection systems.

Detection of fake news is essentially a text classification challenge, in which the goal is to separate trustworthy and untrustworthy news stories using linguistic cues. Nevertheless, this is not an easy task since fake materials are usually composed in a way that they sound more like a standard journalism. Models can also study superficial stylistic signs and not substantive signs of credibility. Consequently, high accuracy is not sure to be robust or practically useful.

This project aims at designing and testing an NLP-based fake news classification pipeline with a focus on data integrity, generalization, and interpretability. We do not only adopt complex architectures but explore the extent to which we can reach performance when we carefully preprocess, and successively more elaborate modelling methods are used.

2. Related Work

Machine learning and NLP methods have seen wide usage in detecting fake news. Initial methods used consisted of handcrafted characteristics like n-grams, sentiment score, readability metrics, and metadata/article associated metadata. Naive Bayes, logistic regression, and support vectors machine were used as traditional classifiers on these features.

More recent studies have developed deep learning models that are automatically trained on representations of text. Sequential dependencies in news articles have been learnt with recurrent neural networks, especially LSTMs and GRUs. Transformer-based architectures also allow this ability to be expanded by self-attention mechanisms, which allows the modeling of long-range context.

In addition to the classification performance, a number of works highlight the relevance of explainability in detecting misinformation. Such techniques as LIME and SHAP have been used to learn more about what words or phrases are most valuable in prediction of a model. And it is in particular necessary since the societal impact of automated misinformation labeling has a certain implication.

In spite of these developments, previous literature has pointed out ongoing difficulties with the bias of the datasets, topic leakage, and source under-generalization. These issues were highly in consideration of the design decisions and assessment plan in our project.

3. Dataset

3.1 Dataset Description

We have a labeled dataset of 39,105 news articles, which are either classified as real or fake. The dataset is moderately imbalanced with about 54.8% fake articles and 45.2% of the data is real. Each sample is a combination of the text of the article and a binary label.

3.2 Data Leakage Analysis

Initial experiments with simple models resulted in values of accuracy exceeding 99%, which raised the issue of data leakage. Upon closer inspection we found several problems that add to artificially boosted performance. These included duplicate articles appearing in different splits of the dataset and near-duplicate articles with little textual variation.

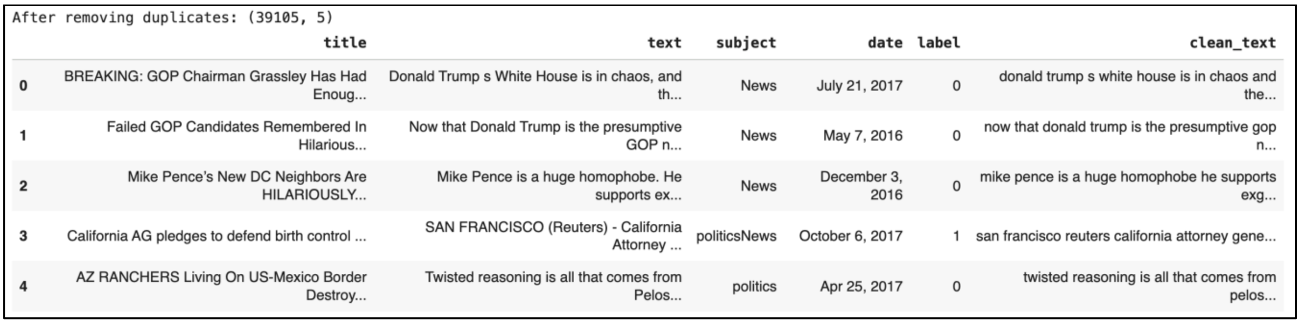
Such leakage makes models prone to memorizing training examples instead of learning generalizable patterns, making metrics of evaluation unreliable. Identifying and solving this problem was a critical step in our pipeline.

3.3 Data Cleaning and Preprocessing

In order to get rid of leakage and enhance the quality of data, we performed several preprocessing steps:

* Elimination of exact duplicate article within the whole dataset.
* Detection and removal of near duplicate articles using text similarity measures.
* Normalization of text (lowercasing & eliminating extraneous formatting artifacts).

All the preprocessing steps have been performed before splitting the data to avoid any overlapping between the training and evaluation data sets.



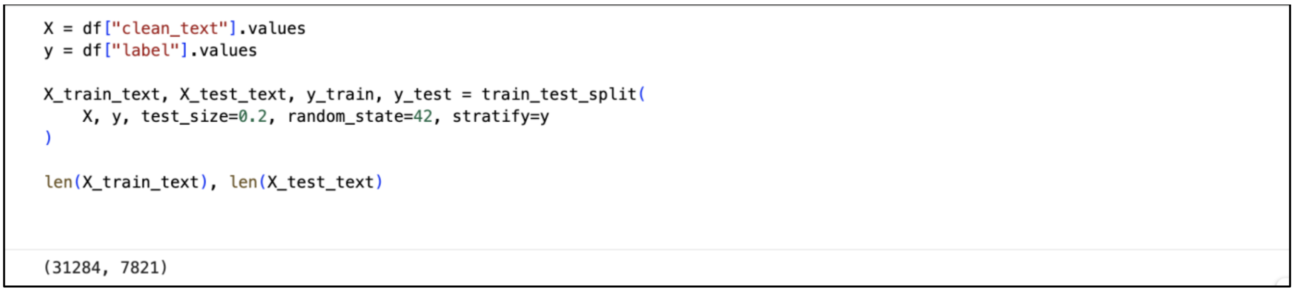
```
After removing duplicates: (39105, 5)
```

	title	text	subject	date	label	clean_text
0	BREAKING: GOP Chairman Grassley Has Had Enoug...	Donald Trump s White House is in chaos, and th...	News	July 21, 2017	0	donald trump s white house is in chaos and the...
1	Failed GOP Candidates Remembered In Hilarious...	Now that Donald Trump is the presumptive GOP n...	News	May 7, 2016	0	now that donald trump is the presumptive gop n...
2	Mike Pence's New DC Neighbors Are HILARIOUSLY...	Mike Pence is a huge homophobe. He supports ex...	News	December 3, 2016	0	mike pence is a huge homophobe he supports exg...
3	California AG pledges to defend birth control ...	SAN FRANCISCO (Reuters) - California Attorney ...	politicsNews	October 6, 2017	1	san francisco reuters california attorney gene...
4	AZ RANCHERS Living On US-Mexico Border Destroy...	Twisted reasoning is all that comes from Pelos...	politics	Apr 25, 2017	0	twisted reasoning is all that comes from pelos...

Figure 1. Dataset shape after duplicate removal and sample output of cleaned text fields used for preprocessing.

3.4 Dataset Splitting

After cleaning the data set was split into three parts (training, validation, and test) with a stratified 80/10/10 split. Stratification ensured that the proportions of classes were preserved across splits. The validation set was used to tune the hyperparameters, and the test set was used for the purpose of final evaluation only.



```
X = df["clean_text"].values
y = df["label"].values

X_train_text, X_test_text, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42, stratify=y
)

len(X_train_text), len(X_test_text)
```

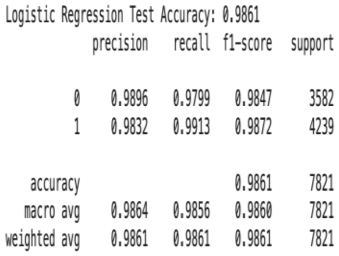
```
(31284, 7821)
```

Figure 2. Stratified train test split of cleaned text data preserving class distribution.

4. Baseline Model TF-IDF + Logistic Regression

As a non-neural baseline, we represent each article by Term Frequency Inverse Document Frequency TF IDF features calculated from the cleaned text. To avoid data leakage, the TF IDF vectorizer was only fit on the training split and applied to validate and test sets. This ensures that no vocabulary or frequency statistics from the evaluation data affect the training of the model. A Logistic Regression classifier was trained on these resulting TF IDF vectors. Model training was done on training data set with its performance measured on the held-out testing data set. This approach has a good lexical baseline by virtue of its simplicity, interpretability, and efficiency and is often used in text classification tasks. Despite not being able to model the semantic context, word order, or long-range dependencies, the TF IDF based Logistic Regression model gave a good performance on the cleaned dataset. This indicates that fake news detection has a lot of signal at the lexical and stylistic level, where word usage patterns alone are very informative. However, this baseline is only fundamentally limited by its bag of words representation. It is unable to capture negation, sentence structure, and contextual meaning that is the reason for using transformer-based models in the next experiments.

To establish a strong non neural baseline, we trained a Logistic Regression classifier using TF IDF features extracted from the cleaned text, ensuring that vectorization was fit only on the training set to avoid data leakage, as shown in Figure 4.



```
Logistic Regression Test Accuracy: 0.9861
              precision    recall  f1-score   support

           0     0.9896    0.9799    0.9847      3582
           1     0.9832    0.9913    0.9872      4239

    accuracy                         0.9861      7821
   macro avg     0.9864    0.9856    0.9860      7821
weighted avg     0.9861    0.9861    0.9861      7821
```

Figure 4. Classification report and test accuracy for the TF IDF Logistic Regression baseline model on the cleaned dataset.

4.1 Hyperparameter Tuning for Logistic Regression

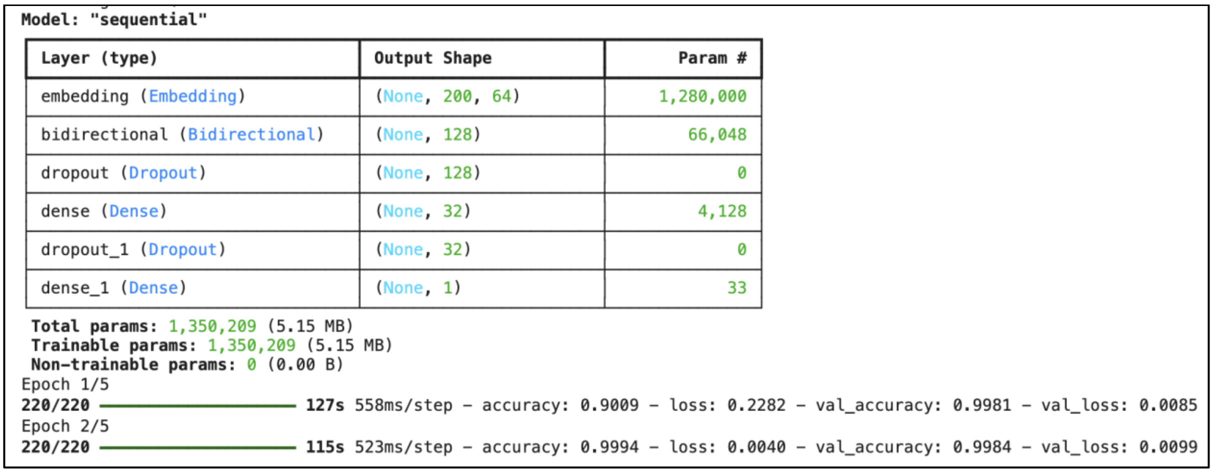
To improve the performance of the baseline Logistic Regression model, we performed hyperparameter tuning using GridSearchCV. The parameters tested included the regularization strength C ([0.1, 1.0, 3.0, 10.0]) and class_weight (None or balanced). Three-fold cross-validation was used on the training set, optimizing for accuracy. The best combination of parameters found was C = 10.0 with class_weight = balanced, achieving a cross-validated accuracy of 0.9895. Evaluating this tuned model on the test set yielded a test accuracy of 0.9909. This hyperparameter optimization ensures that the Logistic Regression model generalizes better while accounting for class imbalance, providing a stronger baseline for comparison with the deep learning models.

Model	Hyperparameters	CV Accuracy	Test Accuracy
Logistic Regression (TF-IDF)	C = 10.0, class_ weight = balanced	0.9895	0.9909

5. Bidirectional LSTM: Deep Learning Model

To address the limitations of the TF IDF baseline, we trained a bidirectional Long Short Term Memory LSTM model that operates on token sequences rather than bag of words representations. This allows the model to capture sequential dependencies and contextual relationships between words within an article. The bidirectional structure enables the network to incorporate information from both preceding and following tokens when learning representations.

We experimented with different numbers of hidden units and dropout rates to balance model capacity and regularization. Increasing the number of hidden units improved representational power, while higher dropout values helped mitigate overfitting observed during training. Compared to the TF IDF baseline, the LSTM model achieved improved evaluation performance, particularly in cases where contextual information is important. However, the performance gains were modest relative to the increase in computational cost, highlighting the strength of well-tuned lexical baselines and motivating the use of more expressive transformer-based models.



```
Model: "sequential"
┌──────────────────────────────────┬───────────────────────────┬───────────────┐
│ Layer (type)                     │ Output Shape              │       Param # │
├──────────────────────────────────┼───────────────────────────┼───────────────┤
│ embedding (Embedding)            │ (None, 200, 64)           │     1,280,000 │
├──────────────────────────────────┼───────────────────────────┼───────────────┤
│ bidirectional (Bidirectional)    │ (None, 128)               │        66,048 │
├──────────────────────────────────┼───────────────────────────┼───────────────┤
│ dropout (Dropout)                │ (None, 128)               │             0 │
├──────────────────────────────────┼───────────────────────────┼───────────────┤
│ dense (Dense)                    │ (None, 32)                │         4,128 │
├──────────────────────────────────┼───────────────────────────┼───────────────┤
│ dropout_1 (Dropout)              │ (None, 32)                │             0 │
├──────────────────────────────────┼───────────────────────────┼───────────────┤
│ dense_1 (Dense)                  │ (None, 1)                 │            33 │
└──────────────────────────────────┴───────────────────────────┴───────────────┘
 Total params: 1,350,209 (5.15 MB)
 Trainable params: 1,350,209 (5.15 MB)
 Non-trainable params: 0 (0.00 B)
Epoch 1/5
220/220 ━━━━━━━━━━━━━━━━━━━━ 127s 558ms/step - accuracy: 0.9009 - loss: 0.2282 - val_accuracy: 0.9981 - val_loss: 0.0085
Epoch 2/5
220/220 ━━━━━━━━━━━━━━━━━━━━ 115s 523ms/step - accuracy: 0.9994 - loss: 0.0040 - val_accuracy: 0.9984 - val_loss: 0.0099
```

Figure 5. Model summary and training progress of the Bidirectional LSTM on the cleaned dataset showing layer details, parameters, and epoch-wise loss and accuracy.

6. Transformer-based and LSTM Model Experiments

To extend our experiments beyond classical and shallow neural models, we implemented a DistilBERT-based classifier and tested multiple configurations of a Bidirectional LSTM model.

For the DistilBERT model, we tokenized the input texts using DistilBertTokenizerFast and trained the DistilBertForSequenceClassification on sample news texts. Training was performed on GPU when available, and the model's loss was monitored across epochs. The output of the training loop (Figure 6) shows decreasing loss values, indicating that the model was able to learn from the input sequences.

For the LSTM experiments, we evaluated three configurations with varying hidden units, dropout rates, and embedding dimensions. Early stopping was used to prevent overfitting. Each configuration was trained for a small number of epochs to test feasibility, and performance was evaluated on the test set. Table 1 summarizes the accuracy obtained for each configuration.

Units	Dropout	Embedding Dim	Test Accuracy
64	0.3	64	0.9986
128	0.3	64	0.9983
128	0.5	64	0.9995

Figure 6. Training output of the DistilBERT model on sample news texts showing epoch-wise loss values.

7. Training and Evaluation

7.1 Training Strategy

Both models were trained using the training set, with the model performance monitored using the validation set. Early stopping was used for LSTM using validation loss to avoid overfitting. Hyperparameters were chosen from the validation performance rather than the test performance.

7.2 Evaluation Metrics

We tested the models based on accuracy, precision, recall, F1-score, and confusion matrix. While accuracy is a summary measure, precision and recall are especially important in misinformation detection, where the consequences of false positives and false negatives are different.
Confusion matrix analysis revealed that both models gave balanced performance among classes, though there were some misclassifications, especially with less-misleading articles.

7.3 Confusion Matrix Analysis

The confusion matrices show how each model classified fake versus real news on the test set.

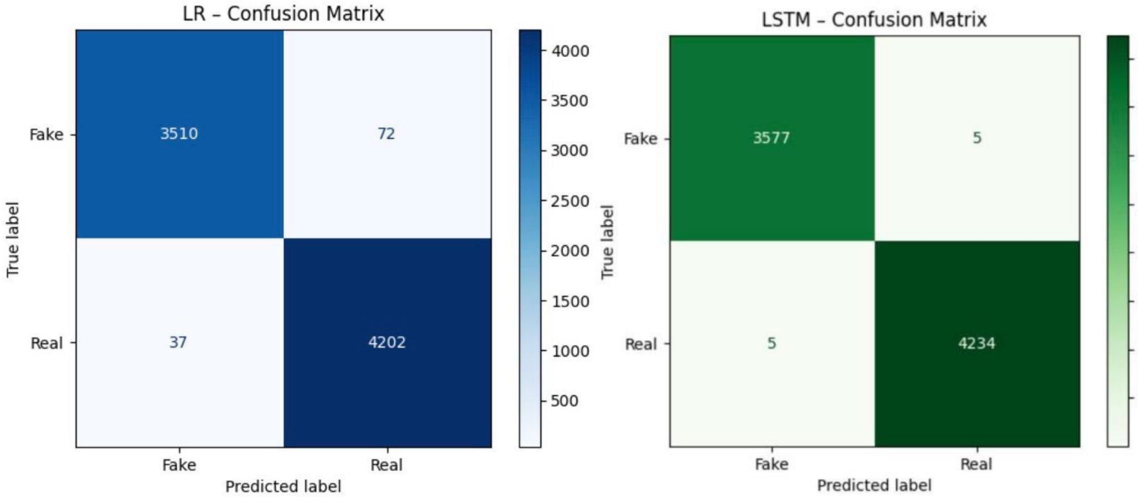


Figure 7. Confusion matrices for fake news classification on the test dataset. Left: Logistic Regression (TF-IDF) baseline model. Right: Bidirectional LSTM model. True labels are shown on the vertical axis and predicted labels on the horizontal axis.

Explanation: Both models show strong classification performance across classes. The LSTM slightly improves classification of nuanced articles, but misclassifications remain for less-obvious cases.

7.4 Precision-Recall Analysis

Next, we evaluate the trade-off between precision and recall for both models.

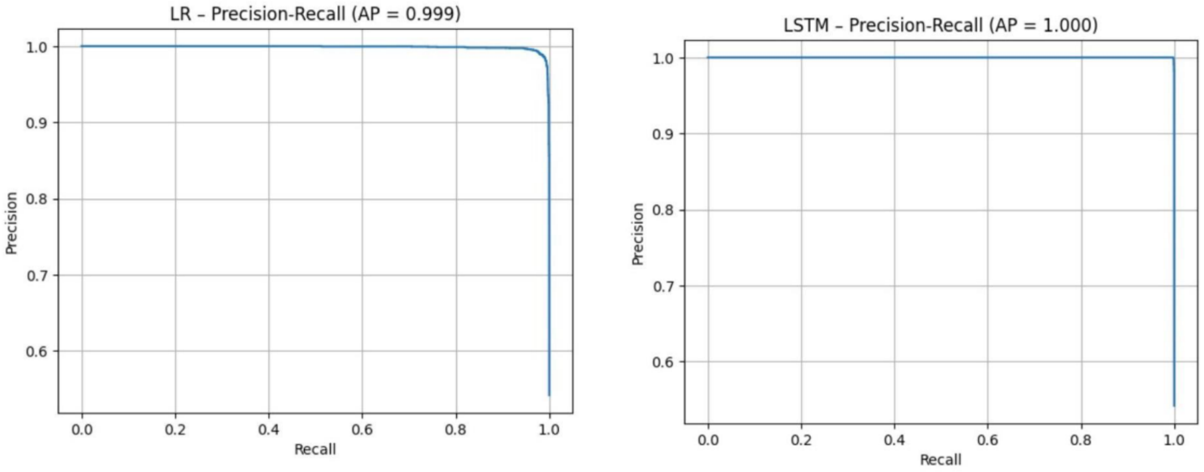


Figure 8. Precision-Recall curves for fake news classification on the test dataset. Left: Logistic Regression (TF-IDF) model with Average Precision (AP) score. Right: Bidirectional LSTM model with AP score.

Explanation: The curves indicate that both models maintain high precision and recall, with LSTM slightly improving on more challenging samples.

7.5 ROC Curve Analysis

Finally, we compare the ROC curves to assess the models' ability to distinguish between classes.

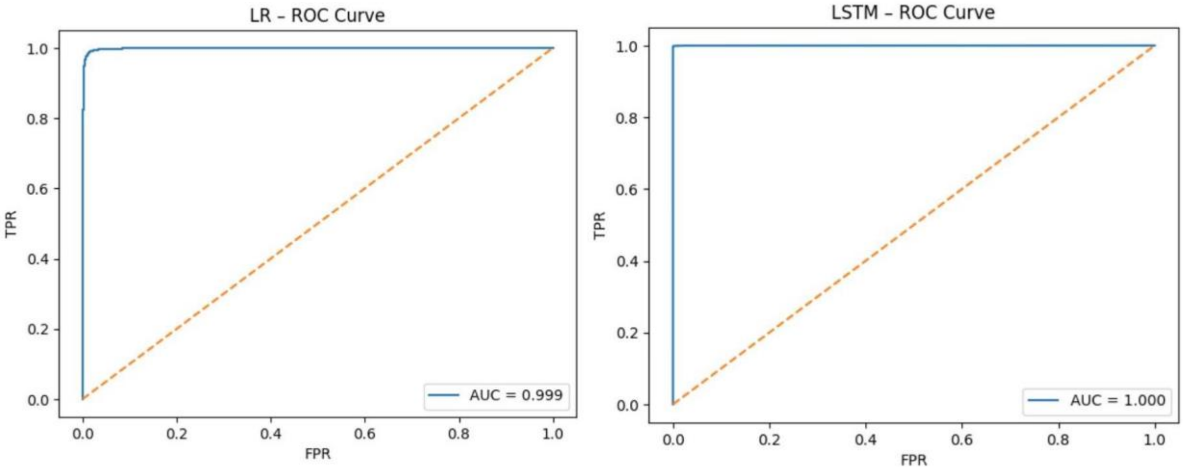


Figure 9. ROC curves for fake news classification on the test dataset. Left: Logistic Regression (TF-IDF) model with AUC score. Right: Bidirectional LSTM model with AUC score.

Explanation: Both models achieve high AUC values, confirming strong discrimination between fake and real articles. LSTM shows marginal improvement in capturing subtle distinctions.

8. Explainability Analysis

High accuracy is not a sufficient requirement to put fake news detection systems into trustworthy practice. To better understand the model behavior, we used SHAP and LIME for explainability.

SHAP gives contribution values to individual words, depending on how they affect predictions. Words associated with sensational or emotional language tended to drive predictions towards the fake class, while neutral and formal words tended to go towards real classifications.

LIME gave local explanations for individual predictions by perturbing input text and fitting an interpretable surrogate model. This allowed cases to be inspected in detail, at the article level, as to why particular articles were identified as fake or real. These analyses suggest that the models are learning mostly the stylistic and linguistic cues of credibility and not the correctness of factual information.

9. Limitations and Failure Cases

Despite the values of the performance, there are still a number of limitations. First, the biases in the stylistic features of this dataset can be exploited by models, limiting the ability to generalize to sources that have not been seen or changing misinformation strategies. Second, the models have difficulties with more subtle language such as sarcasm or satire, which can look very similar to legitimate reporting.

Additionally, our evaluation is of a single dataset, and cross-domain generalization was not tested. These limitations point to the need for more diverse data sets and evaluation protocols.

10. Conclusion and Future Work

In this project, we have created a NLP pipeline for fake news classification, moving from lexical baselines to contextual deep learning models. Careful preprocessing of the data was critical to setting realistic performance benchmarks. While deep learning models brought improvements, the results highlight that the quality of data and the rigor of evaluation are often more important than the complexity of the model.

Future work may involve the potential incorporation of Transformer-based architectures, the cross-source generalisation evaluation and potential external knowledge integration for factual verification. Addressing these challenges is important to achieving robust, responsible misinformation detection systems.

11. Contributions

Harshitha Talla contributed towards data preprocessing, leakage analysis, TF-IDF and LSTM models implementation, evaluation and report writing.

Sanjana Chowdary Muppuri has contributed to model experimentation, explainability analysis, interpretation of the results and report writing.
Contribution was roughly 50% - 50%.

12. AI Usage Statement

The AI tools were only used to make minor changes for grammar and clarity (around 5%). All modeling, analysis, experimenting and writing were done by the authors.