1 Evaluation Measure

As a classical classification task, we use four evaluation metrics to observe the model performance. The model output and the true labels are stored on the .npy files. The evaluate function loads the true labels (y_true) and predicted labels (y_pred) from the specified file paths. After loading, it computes several key evaluation metrics: accuracy, precision, recall, and F1 score. Each metric is calculated using functions from the sklearn.metrics module. The equations used to calculate the metrics are as follows:

Accuracy: The fraction of correct predictions among the total number of cases evaluated.

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

Precision: The ratio of correctly predicted positive observations to the total predicted positives.

$$\text{Precision} = \frac{TP}{TP + FP}$$

where (TP) are true positives and (FP) are false positives.

Recall (Sensitivity): The ratio of correctly predicted positive observations to all actual positives.

$$\text{Recall} = \frac{TP}{TP + FN}$$

where (TP) are true positives and (FN) are false negatives.

F1 Score: The weighted average of Precision and Recall.

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

The results of the test are also visualized by the confusion matrix as a heatmap, which will be also plotted by the evaluation script score.py.

In the domain of fake news detection, while a model may exhibit high classification results, it is necessary to assess the model ability under varying circumstances. The F1-Score provides a balanced measure in cases where precision and recall may trade off against each other in imbalanced datasets. Precision is also important if falsely identifying real news as fake can have significant impacts on the results (Mridha et al., 2021).

2 Simple Baseline

The simple baseline is built by a simple and shallow neural network written with Keras, consisting of a stack of densely connected layers (Dense) with the 'relu' activation function. The output layer uses the 'sigmoid' activation, appropriate for binary classification tasks.

We use two datasets in this project. It reads data from two distinct datasets, the LIAR dataset and the FNC-1 dataset, processes the text to remove noise and standardize the format, and subsequently converts the textual data into numerical representations using Term Frequency-Inverse Document Frequency (TF-IDF) vectorization. Specifically, we map the multi-class labels from both datasets to a binary classification scheme:

In the LIAR dataset, statements categorized as "false", "half-true", "pants-fire", and "barely-true" are mapped to 0, and "mostly-true", and "true" are mapped to 1.

In the FNC-1 dataset, labels "agree", "disagree", and "discuss" are mapped to 1, while "unrelated" is mapped to 0. This mapping suggests a focus on distinguishing related from unrelated statements rather than the traditional stance detection.

We train a 4-layer neural network model with only the LIAR dataset of 10 epochs and a batch size of 256. The accuracy on testset is **64.72%** and the F1 score is **0.40**.

To use the FNC dataset which gives the relation label for news body and news title, we trained a simple neural network. We got ***64.85%*** accuracy and ***0.51*** f1 score for this model. We are assuming this model should have some feature extraction function after training. Then, we feed the LIAR data into this model, we can get a binary label indicating a potential relationship between the first sentence and the rest sentence.

With the extra feature obtained from the FNC model, we train a new neural network with the structure as the first model. The f1 score is better than the model using just LIAR dataset - **0.45**, but the accuracy, **62.11%**, is lower than the first model.

We noticed that the best result on the LIAR-PLUS test set is lower than our expectation. The best model achieved a **77.2%** accuracy for binary classification and a **37.4%** accuracy for 6-class classification.

3 Strong Baseline

LSTMs are used to process word vectors generated from news content(FNC and LIAR datasets), By utilizing the word2vec model to convert text into numerical vectors, these vectors become the input for the LSTM network. Traditional RNNs may have problems with long-term dependencies due to the vanishing gradient problem. LSTMs solve this with the structure of input, forget, and output gates. It allows them to retain information over longer periods, which is crucial for understanding the context of news articles. Unlike feedforward networks, LSTMs can model sequences of variable lengths, which is essential in news content where article lengths can vary significantly. In this project, the LSTM structure is a double-layered LSTM with 150 hidden units per layer, incorporating dropout of 0.4 for regularization. The batch size is 1024. It processes input features through these LSTM layers, followed by a dropout and a linear layer, to perform binary classification of textual data for fake news detection. The dataset is imbalanced so LSTMs may be biased towards the majority class. We try to solve it by introducing the weight parameter in the initialization of the criterion.

The accuracy is **61.64%**, and the F1-score is **0.52**. The accuracy is lower than other baseline models, but the F1-score is a bit higher. This indicates LSTM model is better than the baseline models. Also, it shows a conflict on two different scoring matrix: accuracy and f1 score.

4 Github

https://github.com/kevin00hu/CIS-5300-final-project

5 Reference

Mridha, M. F., Keya, A. J., Hamid, M. A., Monowar, M. M., & Rahman, M. S. (2021). A comprehensive review on fake news detection with deep learning. IEEE Access, 9, 156151-156170