SII - Prediction Homework

Mihai Anghelin

Jan 2025

1 Introduction

To address the problem of fake news detection, a pre-trained transformer model, CamemBERT, a BERT variant optimized for the French language, was used.

The data set provided was divided into two subsets:

- **Training set**: Used for model training.
- **Validation set**: Used to evaluate performance during training.

2 Methodology

2.1 Data Preparation

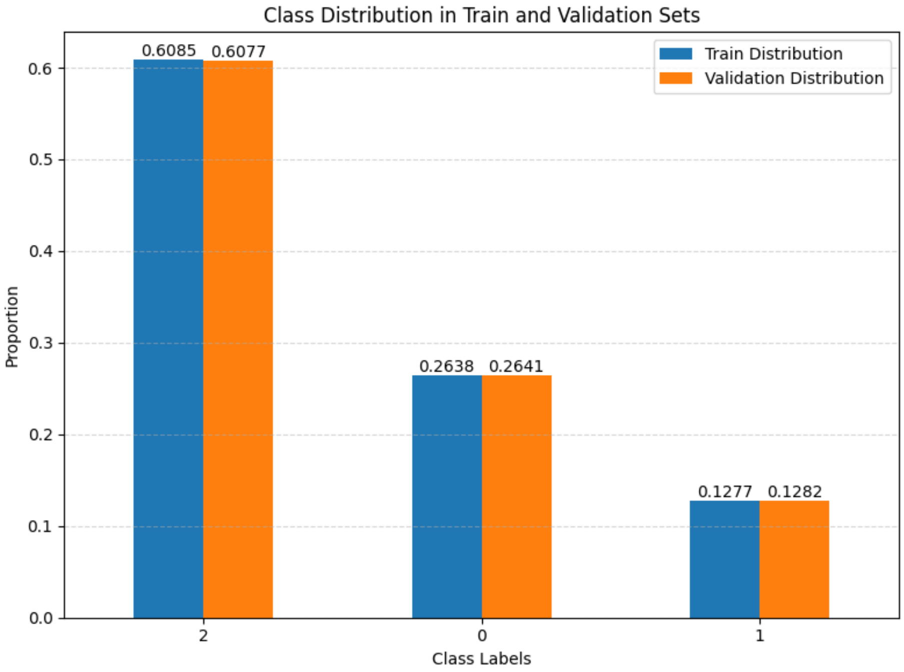

Figure 1: Class Distribution in Train and Validation Sets

The data set was divided into training and validation sets using *train_test_split* from *sklearn*[1], with 80% allocated for training and 20% for validation. Figure 1 represents how the *stratify* parameter was used to ensure that the class distribution in the training and validation sets remained consistent with the overall data set.

Labels were mapped numerically as follows: *fake* → *0*, *biased* → *1*, *true* → *2*

A custom class *NewsDataset* implemented *torch.utils.data.Dataset* to manage data loading, returning for each sample:

- **input_ids**: Token IDs generated by the tokenizer.

[1] `https://scikit-learn.org/stable/index.html`

- **attention_mask**: Indicating real tokens vs padding.
- **labels**: The numerical label for each news item.

2.2 Text Tokenization

The **CamemBERT** tokenizer ("camembert-base") transformed text into token sequences suitable for the transformer model, with the following settings:

- **Padding**: Ensuring all sequences have the same length.
- **Truncation**: Limiting sequence length to 512 tokens.
- **Tensor Conversion**: Converting tokens into PyTorch tensors.

2.3 Training Configuration

The training process was configured using *TrainingArguments* from transformers with the following settings:

- **Epochs**: 10
- **Learning rate**: 2e-5
- **Batch size**: 8 examples per device for both training and validation
- **Evaluation strategy**: Evaluated at the end of each epoch
- **Model saving**: The best model saved based on validation accuracy

To evaluate model performance, a *compute_metrics* function was used to calculate:

- **Accuracy**: Proportion of correctly classified samples.
- **Classification Report**: Precision, recall, and F1-score for each class.

3 Results

The model's performance was tracked across ten epochs using metrics such as training loss, validation loss, and accuracy. The metrics are summarized below in Table 1:

Epoch	Training Loss	Validation Loss	Accuracy
1	0.573800	0.589476	0.771795
2	0.513400	0.632872	0.774359
3	0.423000	0.734434	0.756410
4	0.473800	0.815394	0.769231
5	0.184700	0.754072	0.802564
6	0.110900	0.885092	0.764103
7	0.094000	0.910120	0.782051
8	0.105100	0.980540	0.779487
9	0.011300	1.024335	0.776923
10	0.158600	1.035487	0.774359

Table 1: Training and validation metrics across 10 epochs

The highest accuracy (0.802564) was achieved in Epoch 5. Consequently, the model from Epoch 5 was selected as the final model for further evaluation and predictions.

The prediction label distribution for the test set is shown in Figure 2:

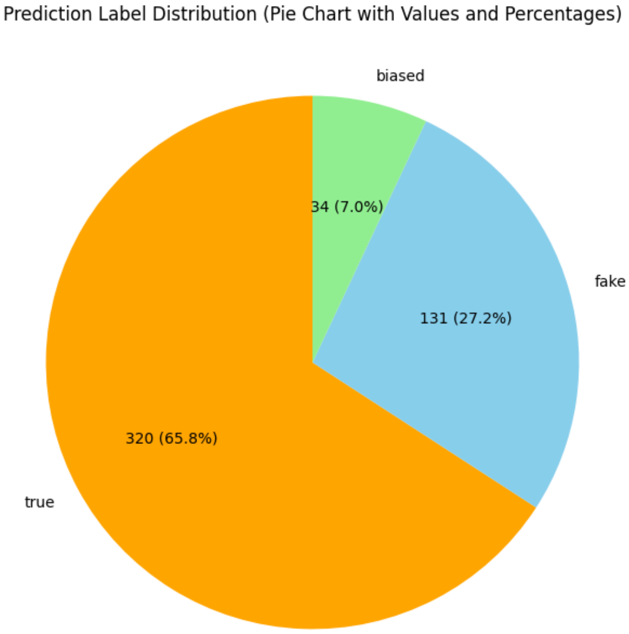

Figure 2: Prediction Label Distribution

4 Code

The code for training CamemBERT for this homework can be found on GitHub: `https://github.com/MihaiAnghelin/FakeNewsPredictionCamemBERT/blob/master/PredictionHomework.ipynb`