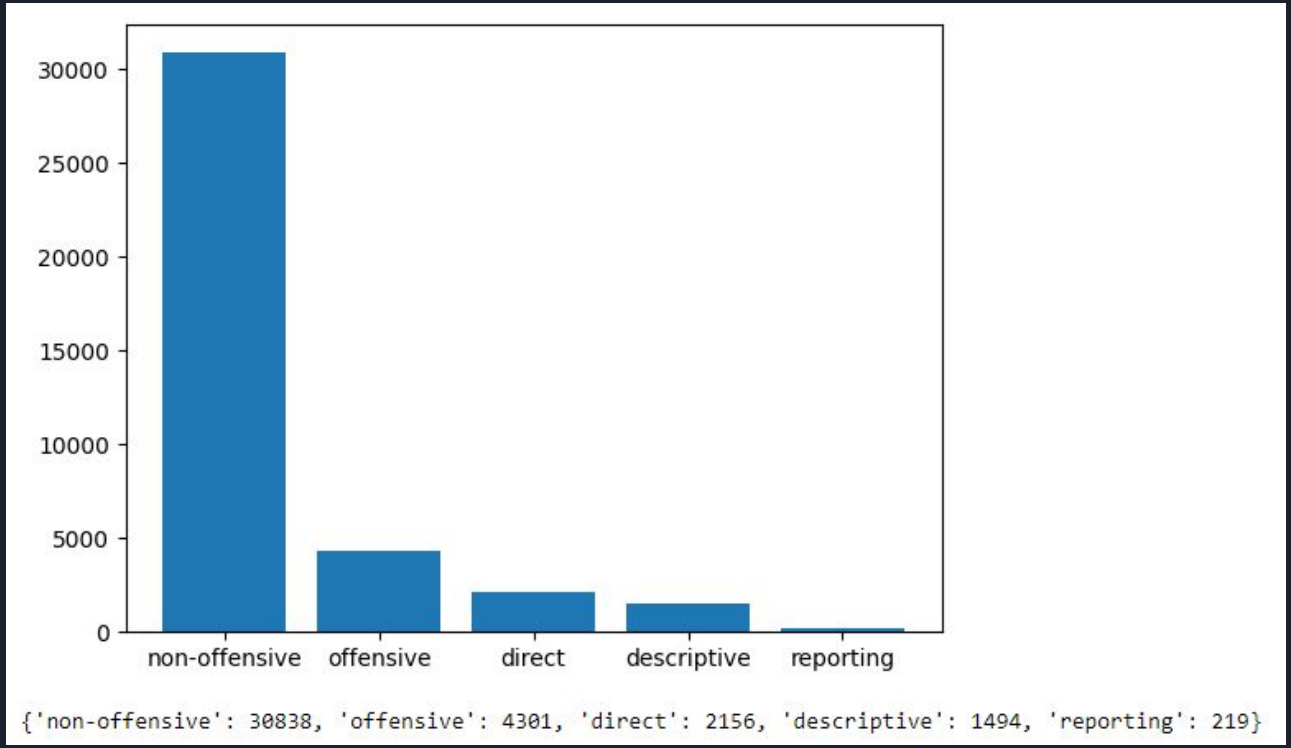
Nitro Language Processing Hackathon

Team GePeTo

Popa Mihai-Cristian, 311
Chirobocea Mihail-Bogdan, 312
Costea Răzvan-George, 311
Marin Mircea-Mihai, 312

Data Analysis




```
{'non-offensive': 30838, 'offensive': 4301, 'direct': 2156, 'descriptive': 1494, 'reporting': 219}
```

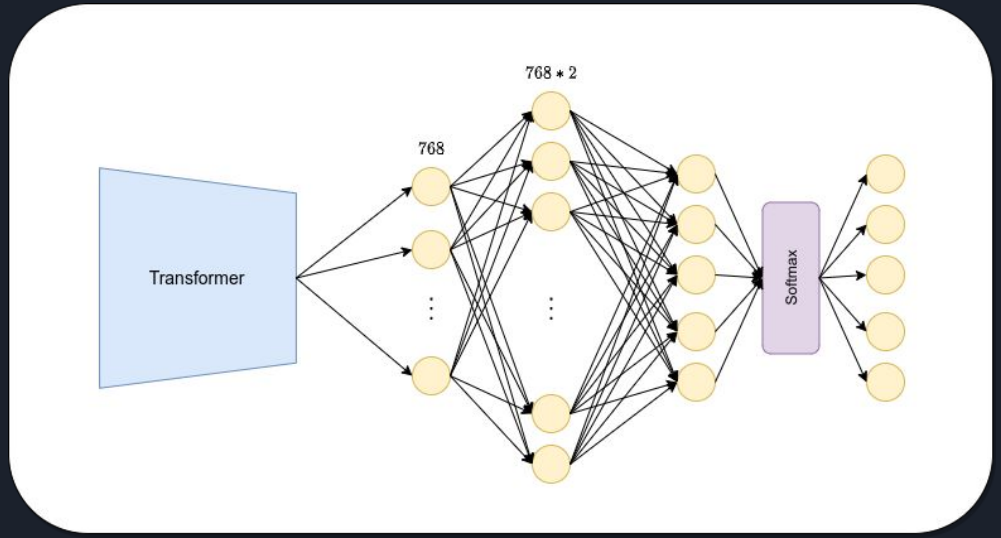
Classic Machine Learning

- TfIdf with multiple classifiers like SVC, MultinomialNB, LogisticRegression

- Max accuracy on test set was around 35%

Deep learning

- Used multiple pretrained transformers: mBERT, XLM-RoBERTa, RoBERT
- With a classifier network
- Max accuracy on test set was pretty bad because the network learned to predict only the third class (non-offensive)
- The problem was the unbalanced dataset



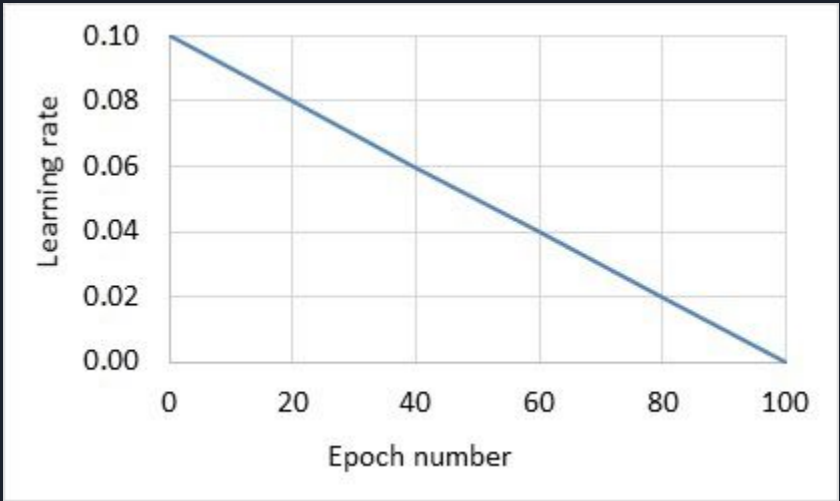

```python
class Classifier(nn.Module):
    def __init__(self):
        super().__init__()
        self.seq = nn.Sequential(
            nn.Linear(768, 768*2),
            nn.BatchNorm1d(768*2),
            nn.Dropout(0.25),
            nn.GELU(),
            nn.Linear(768*2, 5),
            nn.Softmax(),
        )

    def forward(self, x):
        x = self.seq(x)
        return x


class RoBERT_base(nn.Module):
    def __init__(self):
        super().__init__()
        self.transformer = AutoModel.from_pretrained("readerbench/RoBERT-base")
        self.classifier = Classifier()

    def forward(self, input_ids, attention_mask):
        raw_output = self.transformer(
            input_ids, attention_mask, return_dict=True)
        x = raw_output["pooler_output"]
        out = self.classifier(x)
        return out
```

One solution

- We used weighted cross entropy
- We selected the first N samples from “non-offensive” class
- The accuracy increased
- We found that for N=5000 we had the best accuracy
- Best model was RoBERT_base with 55.4% accuracy

Optimization

- AdamW optimizer
- Linear learning rate decay (10 epochs)



Ensemble learning

- We trained all three transformers and put them to vote
- We chose the answer with the most votes
- If it was a draw we chose the answer provided by RoBERT_base
- The best accuracy was 55.6%

Future Work

- Preprocessing
- Weighted ensemble learning
- Freeze x% of transformer weights (prevent overfitting)
- Data augmentation (especially for the reporting class)

Thank you
for your (*se*/*f*) attention!