Barak
Hostel

Automated Research Paper Categorisation

10 February 2024

Agenda

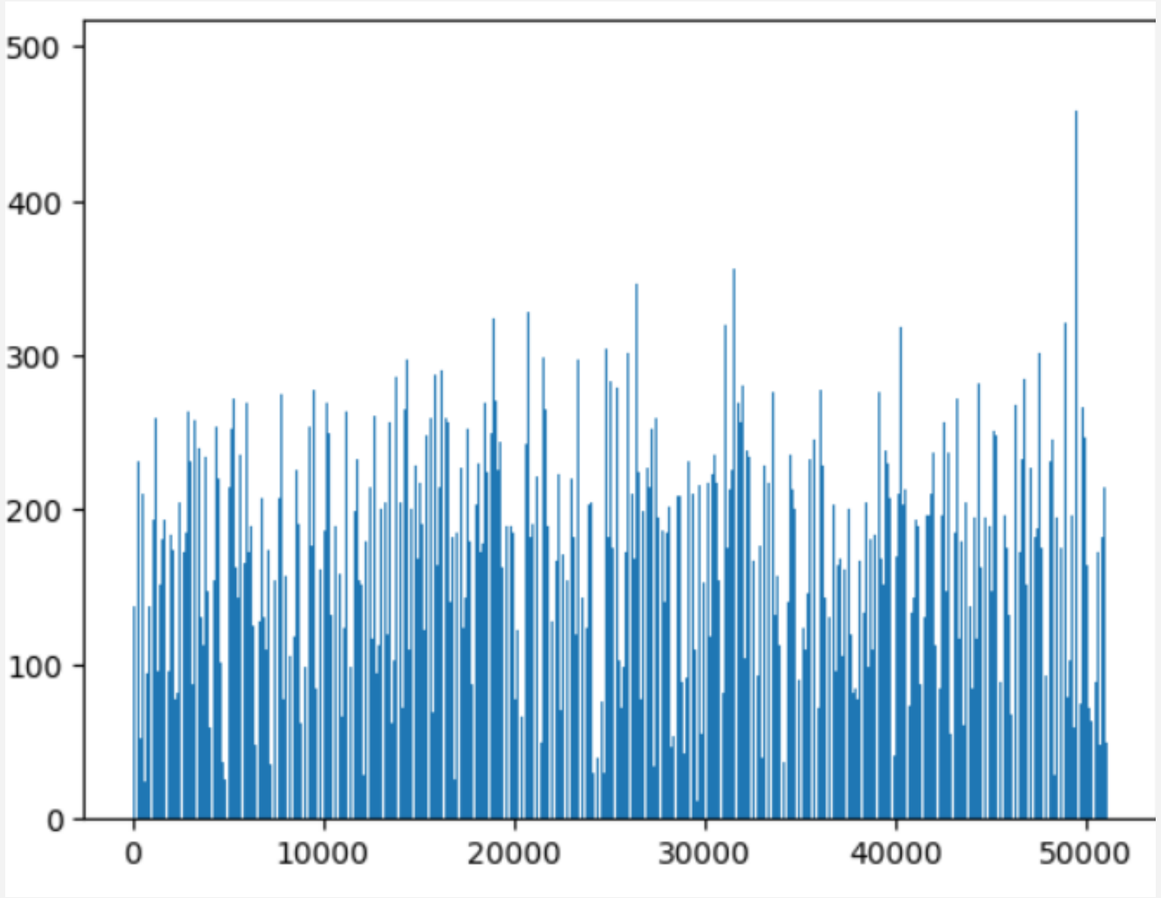
Table of Contents

About The data

	Id	Title	Abstract	Categories
0	9707	Axiomatic Aspects of Default Inference	This paper studies axioms for nonmonotonic con...	['cs.LO']
1	24198	On extensions of group with infinite conjugacy...	We characterize the group property of being wi...	['math.GR']
2	35766	An Analysis of Complex-Valued CNNs for RF Data...	Recent deep neural network-based device classi...	['cs.LG', 'cs.IT', 'eess.SP', 'math.IT']
3	14322	On the reconstruction of the drift of a diffus...	The problem of reconstructing the drift of a d...	['math.PR', 'math.ST', 'stat.TH']
4	709	Three classes of propagation rules for GRS and...	In this paper, we study the Hermitian hulls of...	['cs.IT', 'math.IT']

The given dataset consists of id, title, abstract, and categories of 51210 research papers in the train set and 10974 unlabelled data points in the test set.

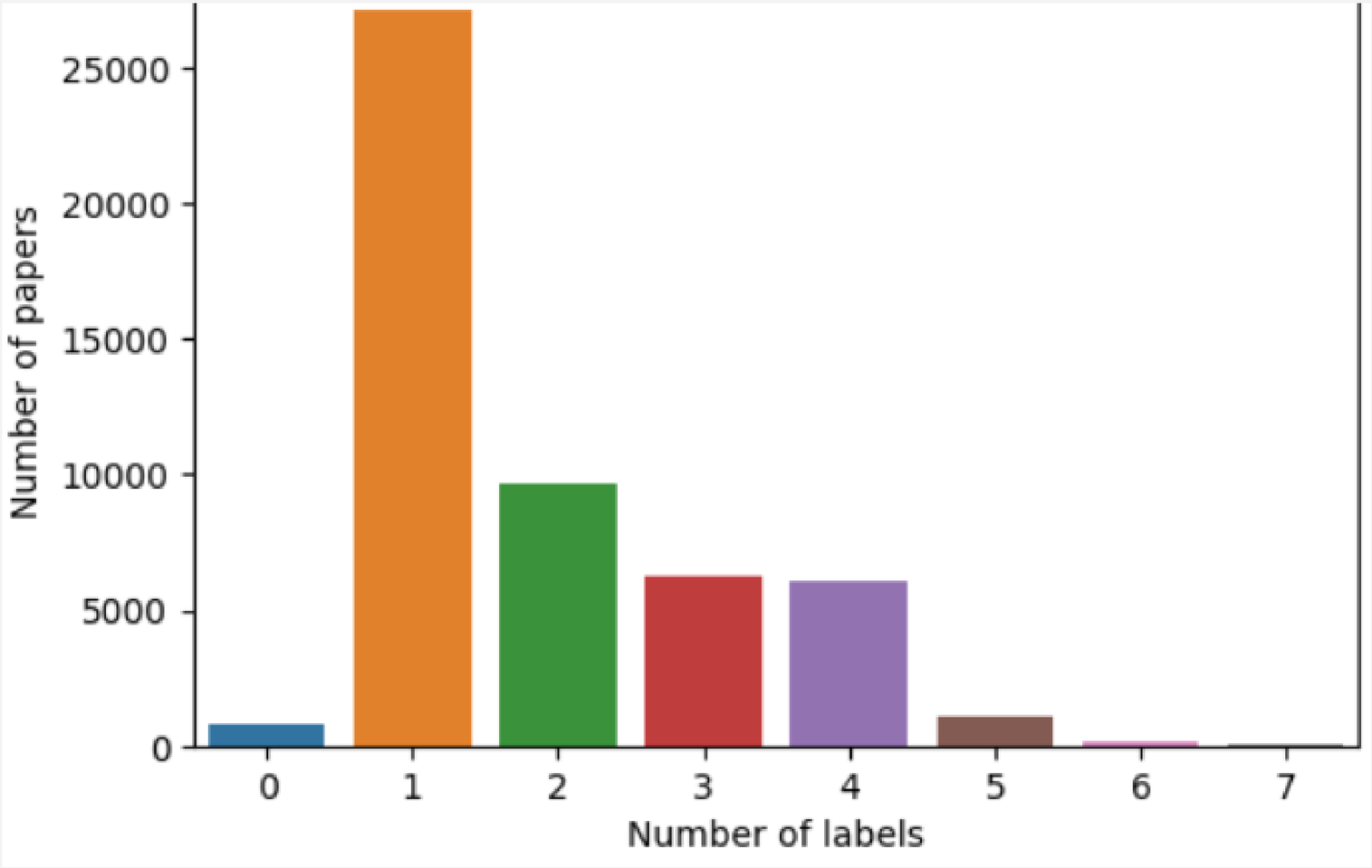
Words Count



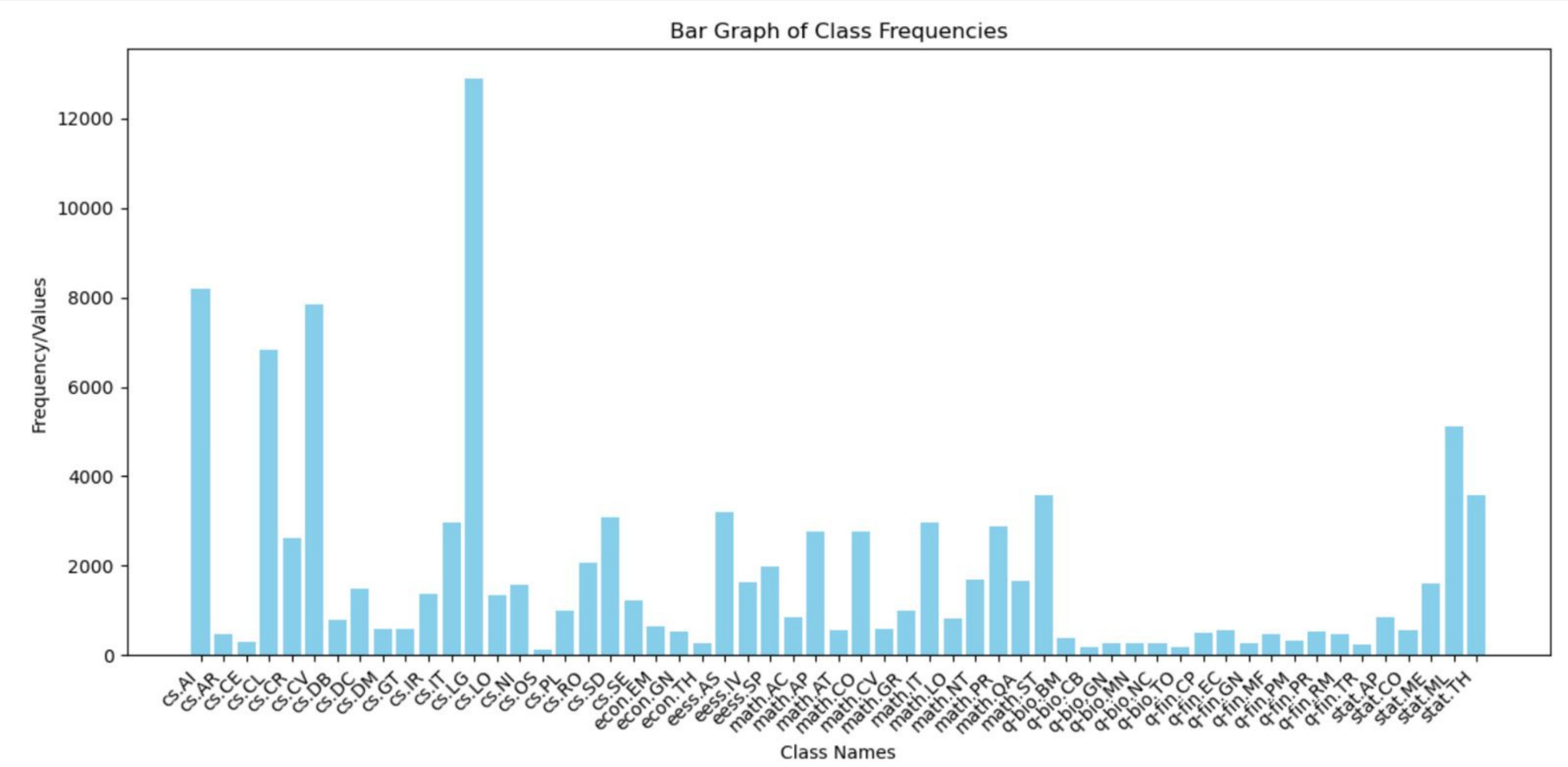
Bar Graph of words and their frequency

We deleted all the stop words, and converted the text to lower case before using it for training of our model .

Class Distribution



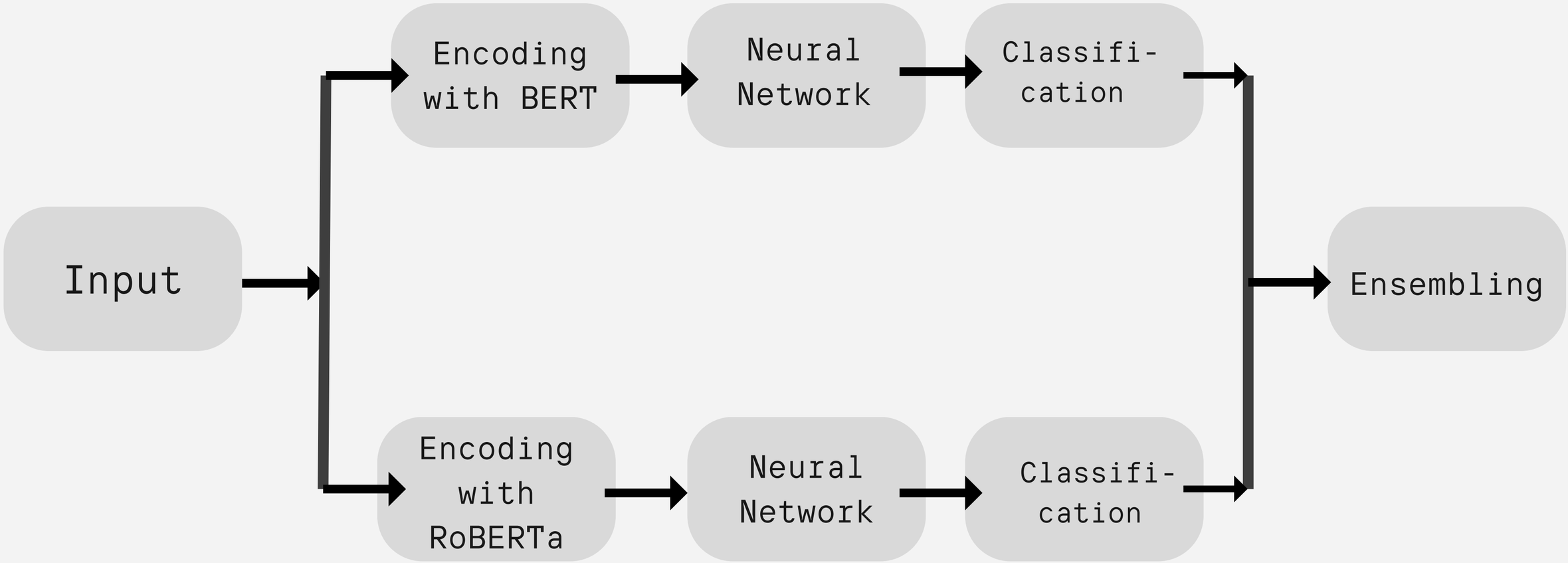
No. of labels vs No. of papers



Labels vs Frequency

We can see that there is huge data imbalance, as the no. of research papers per Class varies between 12000 and 134.

Flowchart


Input
Encoding with BERT
Neural Network
Classifi-cation
Encoding with RoBERTa
Neural Network
Classifi-cation
Ensembling

DistilBERT Pipeline

DistilBERT is a lightweight version of the BERT (Bidirectional Encoder Representations from Transformers) model, developed by researchers at Hugging Face. In conjunction with the DistilBERT model, we employ a custom neural network tailored specifically for classification purposes. The input layer of the neural network is configured with a size of 768 units, which is followed by 2 hidden layers with sizes of 1024, and 512 and an output layer of size 57.

```
DistilBERTClass(
  (l1): DistilBertModel(
    (embeddings): Embeddings(
      (word_embeddings): Embedding(30522, 768, padding_idx=0)
      (position_embeddings): Embedding(512, 768)
      (LayerNorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
      (dropout): Dropout(p=0.1, inplace=False)
    )
    (transformer): Transformer(
      (layer): ModuleList(
        (0-5): 6 x TransformerBlock(
          (attention): MultiHeadSelfAttention(
            (dropout): Dropout(p=0.1, inplace=False)
            (q_lin): Linear(in_features=768, out_features=768, bias=True)
            (k_lin): Linear(in_features=768, out_features=768, bias=True)
            (v_lin): Linear(in_features=768, out_features=768, bias=True)
            (out_lin): Linear(in_features=768, out_features=768, bias=True)
          )
          (sa_layer_norm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
          (ffn): FFN(
            (dropout): Dropout(p=0.1, inplace=False)
            (lin1): Linear(in_features=768, out_features=3072, bias=True)
            (lin2): Linear(in_features=3072, out_features=768, bias=True)
            (activation): GELUActivation()
          )
          (output_layer_norm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
        )
      )
    )
  )
  (pre_classifier): Linear(in_features=768, out_features=768, bias=True)
  (dropout): Dropout(p=0.1, inplace=False)
  (additional_fc1): Linear(in_features=768, out_features=1024, bias=True)
  (additional_fc2): Linear(in_features=1024, out_features=512, bias=True)
  (classifier): Linear(in_features=512, out_features=57, bias=True)
)
```

RoBERTa Pipeline

RoBERTa, short for "Robustly optimized BERT approach," is a variation of the BERT model, introduced by Facebook AI. It builds upon the success of BERT while addressing some of its limitations and improving overall performance. Along with RoBERTa, we used a custom neural network of input size 768 taking embeddings from the RoBERTa tokenizer,having hidden layers of sizes 512,256,128,64 and at last a classifier of size 57.

```
class RoBERTaClass(torch.nn.Module):
    def __init__(self):
        super(RoBERTaClass, self).__init__()
        self.l1 = RobertaModel.from_pretrained("roberta-base")
        self.linear1 = torch.nn.Linear(768, 512)
        self.dropout = torch.nn.Dropout(0.1)
        self.linear2 = torch.nn.Linear(512, 256)
        self.leaky_relu = torch.nn.LeakyReLU()
        self.linear3 = torch.nn.Linear(256, 64)
        self.tanh = torch.nn.Tanh()
        self.classifier = torch.nn.Linear(64, 57)

    def forward(self, input_ids, attention_mask, token_type_ids):
        output_1 = self.l1(input_ids=input_ids, attention_mask=attention_mask)
        hidden_state = output_1.last_hidden_state
        pooler = hidden_state[:, 0]

        linear1_output = self.linear1(pooler)
        linear1_output = self.dropout(linear1_output)

        linear2_output = self.linear2(linear1_output)
        linear2_output = self.leaky_relu(linear2_output)

        linear3_output = self.linear3(linear2_output)
        linear3_output = self.leaky_relu(linear3_output)

        output = self.classifier(linear3_output)
        return output

model = RoBERTaClass()
model.to(device)
```

Experimentations

Implemented Approaches	Public Score
• Bert-Base Uncased + Simple Classifier	0.31
• Longformer + xGBoost(using grid search)	0.43
• distilBERT-base-uncased + NN	0.49
• distilBERT-base-uncased + NN(more epochs)	0.64
• RoBERTa + NN	0.63
• DistilBERT + RoBERTa ensembled	0.67

Scope of Improvement

To enhance the quality of our results, we can explore a couple of methods.Some of them are listed below:

- Sci-BERT is a specialized variant of the BERT model crafted by researchers at the Allen Institute for AI. Sci-BERT is meticulously designed to excel in processing scientific text, making it exceptionally suitable for tasks within scientific domains.
- Addressing data imbalance through dedicated methods can further refine our approach. Implementing strategies tailored to mitigate data imbalances ensures that our model receives balanced and representative training data, thereby enhancing its overall performance and reliability.
- Considering the advancement in natural language processing models, integrating newer and more sophisticated models such as DeBERTa can potentially elevate our performance. DeBERTa, known for its superior performance compared to models like RoBERTa, offers enhanced capabilities and can potentially yield better results for our text classification tasks.

THANK YOU

Team Barak