设计思路

注：本作业全部代码及资料存放在
https://github.com/cauchy221/BUPT-ML-FinalProject

1 预处理策略

在开始前我们对从 Kaggle 下载的数据集进行了观察分析（具体见上述仓库中的 `data_preprocess.ipynb` 文件），可以很直观的感受到，数据集具有以下特点：

1. 数据量非常少（不到 1k 条），这意味着单纯用这些数据训练一般很难得到理想的结果

2. 存在不平衡问题，中性情感对应的数据量远大于极端情感，若不进行处理，模型将很难学会识别 1、5 等类别

3. 格式杂乱，存在很多需要清洗的地方，且包含很多口语化的表达。对此我们有针对性的提出了以下处理策略：

(1) 删除 url 链接

(2) 删除 @ 引导的用户名

(3) 删除 "" 双引号

(4) 删除 `Ì¢‰âÂ‰ã¢` 等乱码

(5) `&` 在 HTML 中表示 `&`，考虑暂时替换成单词 `and`（或许可能还有其它 HTML 编码）

(6) `#xxx` 表示话题，包含有信息，不能直接删除，只能删掉符号

2 方法选择

2.1 模型选择

小组成员调研了近年来在 Sentiment Analysis 方向常见的方法（具体见上述仓库中的 `./research` 文件夹），大致分为机器学习方法（如 SVM、决策树等）和深度学习方法（如 finetune pre-trained model）。

考虑这些方法在多个情感分类 benchmark 上的得分，我们最终选择了微调大规模预训练模型的方法，并选择 BERT（bert-base-uncased）预训练模型。原因如下：

1. BERT 在多个指标上都取得了 SOTA 的效果

2. Twitter 推文有字数限制，不会出现文本过长超出 512 input_max_length 的限制

3. 采用双向编码，融合了上下文信息，有利于情感决策

4. 基于 Transformer 结构，文本特征提取能力强

2.2 数据处理

根据一开始对原始训练数据的分析，我们很快意识到需要进行数据增强。关于 Data Augment 我们考虑并尝试了以下几种方案：

1. 使用 EAD 对数据量较小的类别 1、5 进行数据扩充（并无明显提升）

2. 添加其它评价类多分类数据集进行训练，如 SST-5、Yelp（metric 下降，原因在于其它数据集并不同分布）

3. Google 搜索到了数据量较大的 Self-Driving Car Sentiment 数据集，包含较多评价，且同样来源于 Twitter，和我们的原数据集同分布（最终选择此方案）

由于 BERT 参数量大，一般需要较多数据进行微调，我们最终从搜索到的大数据集中抽样了 6800 条数据，并按 9:1 的比例划分训练集和验证集，并最终在 Kaggle 提供的测试集上进行测试和提交。与此同时我们也不断尝试基于原数据集，尽量取得较高的正确率

3 模型具体架构

```python
class SentimentClassifier(nn.Module):
  def __init__(self, n_classes):
    super(SentimentClassifier, self).__init__()
    self.bert = BertModel.from_pretrained(model_path)
    self.drop = nn.Dropout(p=0.3)
    self.out = nn.Linear(self.bert.config.hidden_size, n_classes)

  def forward(self, input_ids, attention_mask):
    _, pooled_output = self.bert(
      input_ids=input_ids,
      attention_mask=attention_mask,
      return_dict=False,
    )
    output = self.drop(pooled_output)
    return self.out(output)
```

模型架构如上，在 BertModel 的基础上，首先添加一层 Dropout 防止过拟合，并在最后添加一层 Linear 层，将输出映射到长度为 5（num classes）的向量中进行分类