# HugginFace 中文情感分类实验

## 学员信息

姓名：王泽玉

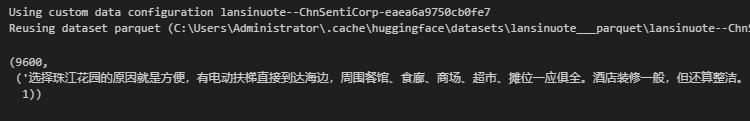
学号：231017000058

## 实验过程

### 数据集

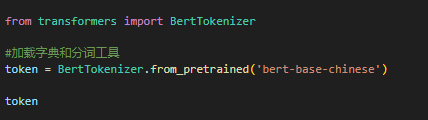
这里使用'lansinuote/ChnSentiCorp'，这是一个情感分类数据集，每条数据中包括一句购物评价和标识(好评或差评)。

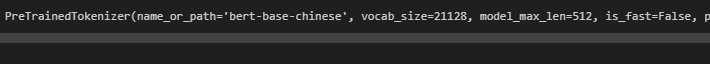




### 编码工具

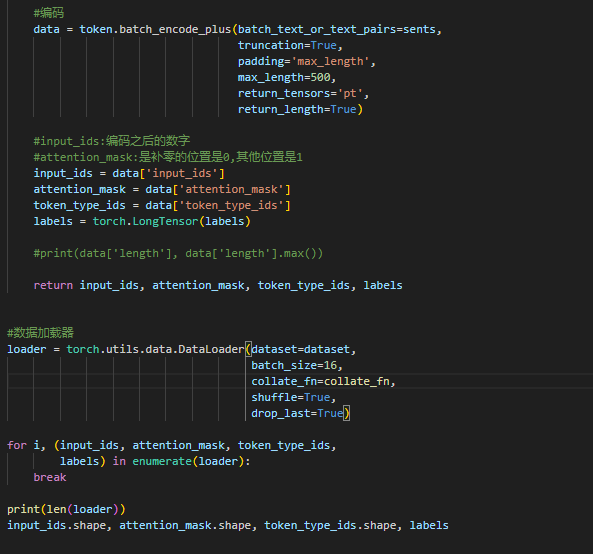
这里加载的编码工具为"bert-base-chinese" ，是一种基于 Transformer 模型架构的预训练语言模型，用于处理中文文本。BERT（Bidirectional Encoder Representations from Transformers）是由Google开发的一种自然语言处理模型，通过在大规模文本数据上进行预训练，可以用于各种下游任务，如情感分析、命名实体识别、问答系统等。

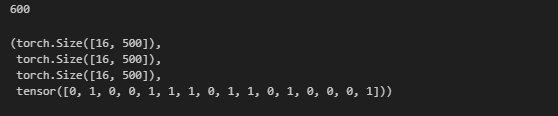




### 数据加载器

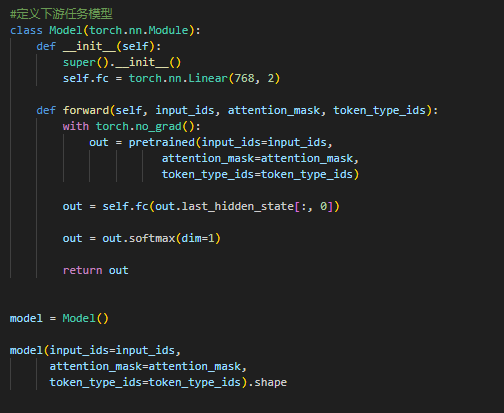
这段代码实现了一个数据加载器，它将数据集分批加载并进行编码。通过定义 collate\_fn 函数，对每个批次的数据进行处理和编码，生成输入编码、补零标识和文本对标识等信息。然后，使用数据加载器将数据集按照设定的批次大小进行批量加载，并在每个 epoch 开始时打乱数据顺序。最后，通过遍历数据加载器，获取每个批次的编码后的输入、补零标识、文本对标识和标签。此代码的目的是为了方便训练和处理大规模数据集时的批量操作。





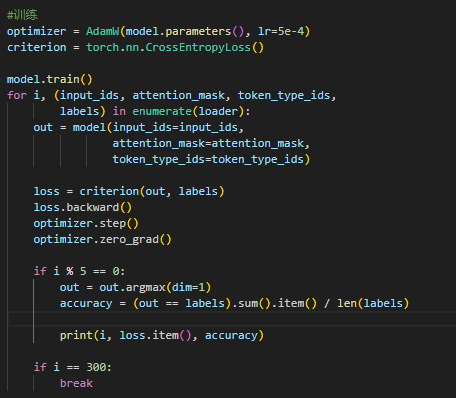
### 下游任务模型

以下代码定义了一个名为 Model 的 PyTorch 模型类，它包含一个全连接层。在前向传播方法中，输入通过预训练模型进行处理并获取最后隐藏状态，然后经过线性变换和 softmax 函数得到最终输出。通过创建 Model 类的实例，并传递输入张量，可以获取输出张量的形状。

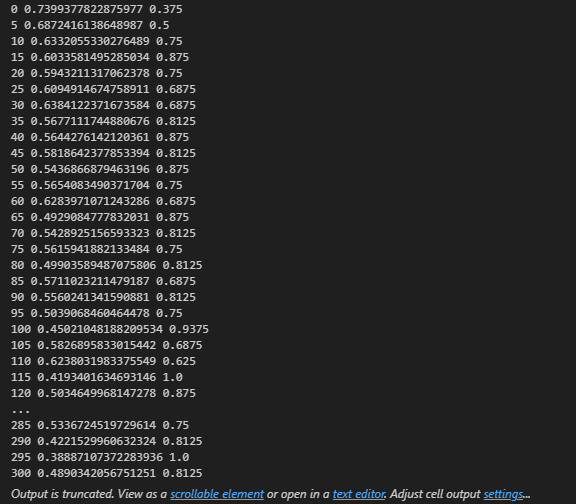


### 训练

下段代码使用了 transformers 库中的 AdamW 优化器和交叉熵损失函数 CrossEntropyLoss 进行训练。通过将模型参数和学习率传递给优化器，以及将模型输出和标签作为输入传递给损失函数，实现了模型的训练过程。在每个批次的迭代中，计算损失并执行反向传播更新模型参数，然后使用优化器清除梯度。每隔一定的迭代次数，计算准确率并打印相关信息。当达到指定的迭代次数时，结束训练过程。



最终训练完毕后，输出的观察数据图表。从图表可以看出，在训练到大约300个steps时，模型已经能够达到大约81%的正确率，并且能够观察到loss是随着训练的进程在不断地下降，学习率也如预期的一样，也在缓慢地下降。



### 测试

下段代码定义了 test 函数，用于在评估模式下对模型进行测试。通过创建数据加载器加载验证集数据，并在每个批次上执行模型的前向传播。计算正确预测的数量和总样本数量，最后输出准确率在85%左右。



