**作业项目：文本情感分析**

组员：贺妍（231017000045）陈罡廷（231017000053）黄援章（231017000051）

**一、项目目标**

本项目将尝试通过中文情感分析数据集seamew/ChnSentiCorp对预训练的Bert模型bert-base-chinese进行训练，并测试其效果。

**二、模型训练及测试过程**

STEP 1：环境准备

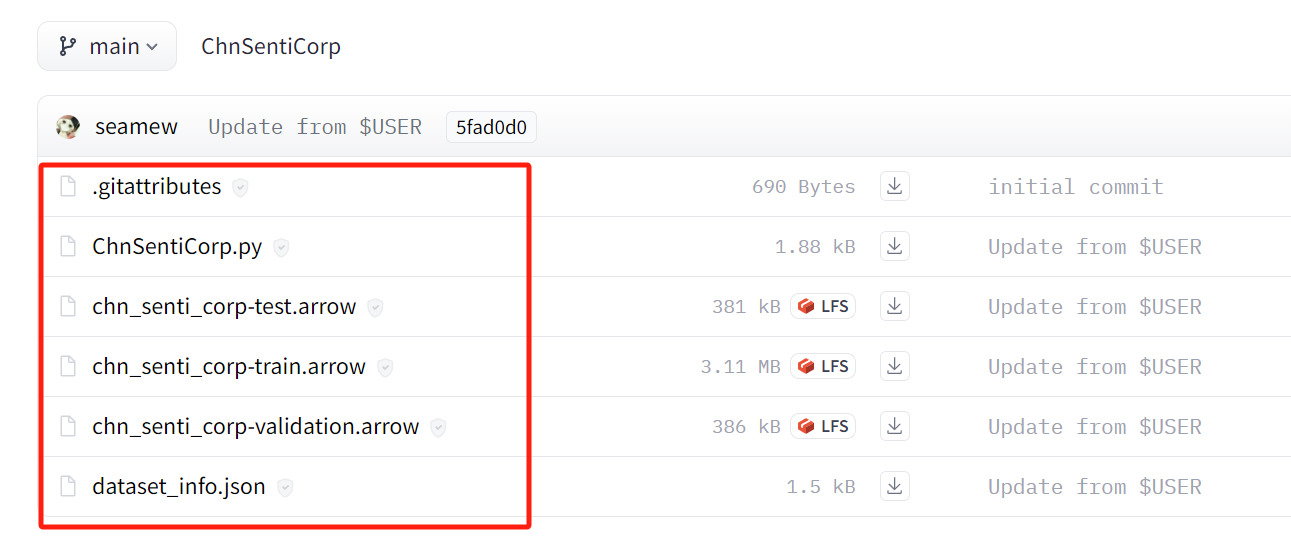
在实现作业项目时同时搭建两套环境，一套是本地windows10环境，一套是云服务器centOS7环境。两套环境互为后备，以免因为环境的原因影响的项目的及时完成。

两套环境均准备了anaconda、pyTorch、transformers、torch、datasets和git。

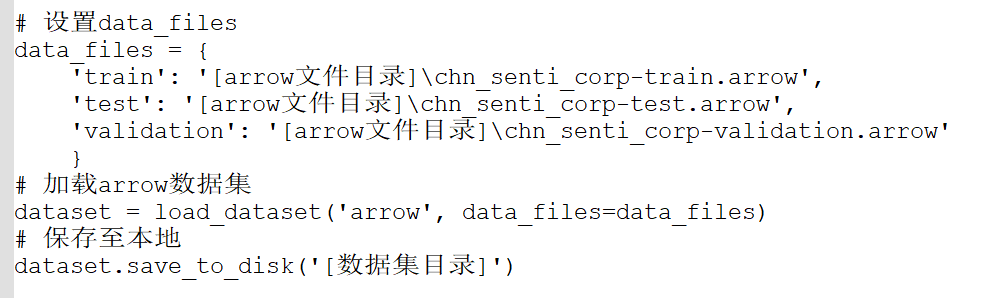
本地windows10环境的代码执行比较顺利，而云服务器上数据集和模型的下载总有些问题，从本地上传的到云服务器的效率也非常慢，所以最终模型的训练是在本地环境跑的。

STEP 2：数据集准备

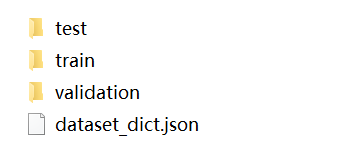
（一）、数据集ChnSentiCorp可直接从huggingface社区下载



将这几个文件直接下载到本地后，执行以下代码，便可直接作为训练数据源



下载的数据目录结构如图：



“train”和“validation”可用来对模型进行训练和验证的数据集。

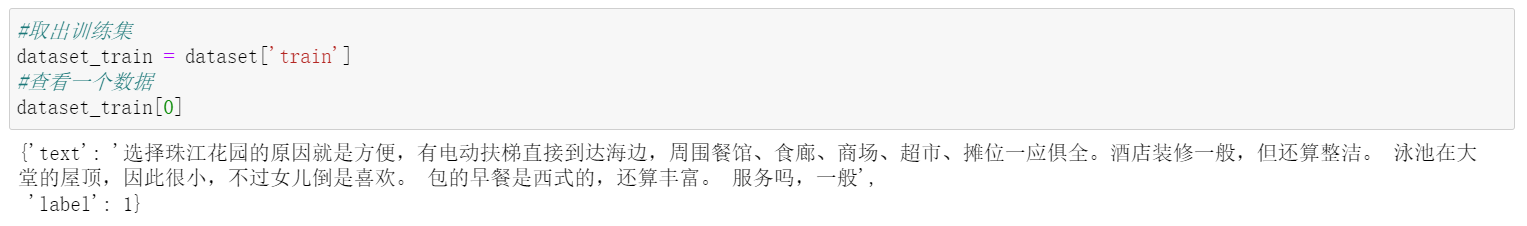
“test”用来对训练好的模型进行预测测试。

1. 、数据集下载到本地后，可以直接查看数据内容：



数据从磁盘加载到dataset字典中，此时dataset包含三个数据集：训练集train共9600条数据、验证集validation共1200条，测试集test共1200条。

同时可以取出其中数据集，并查看某一条数据：



所以该数据集的数据格式为：

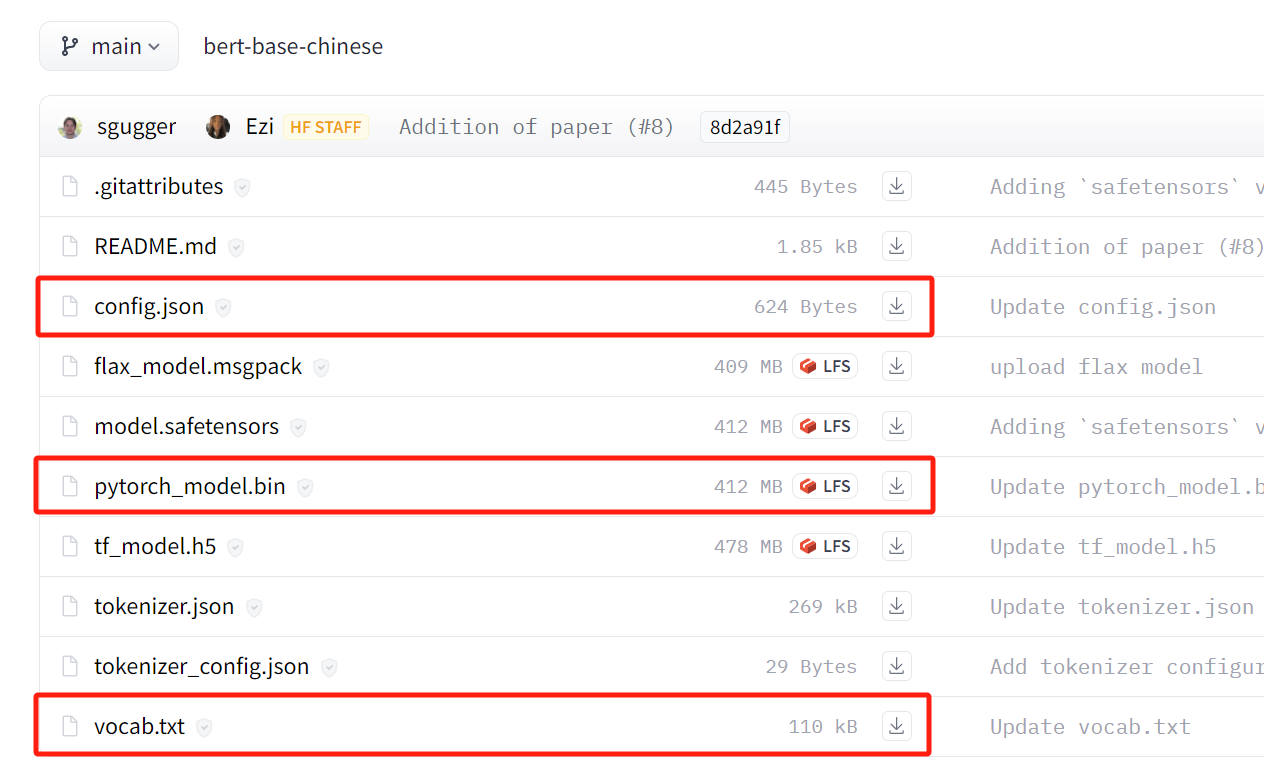
{'text': '选择珠江花园的原因就是方便，有电动扶梯直接到达海边，周围餐馆、食廊、商场、超市、摊位一应俱全。酒店装修一般，但还算整洁。 泳池在大堂的屋顶，因此很小，不过女儿倒是喜欢。 包的早餐是西式的，还算丰富。 服务吗，一般', 'label': 1}

每条数据的“text”为待分析的文本，“label”为1时代表积极的评论，为0时代表消极的评论，且label取值只有0/1

STEP 3：模型准备

同样，模型bert-base-chinese也可以从huggingface社区下载。

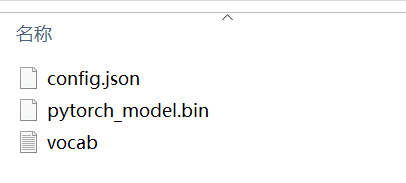
常识几种方法下载的效果均不如人意，于是最后通过直接手工下载下图红色标注的文件至本地，才能正常引入模型并成功训练：



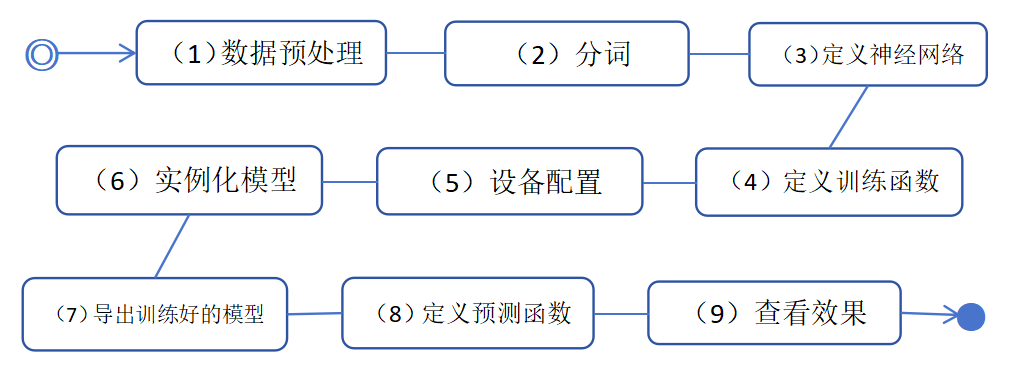
“vocab.txt”是该预训练模型的词汇表。

“pytorch\_model.bin”是pyTorch引用该模型的必要文件。

下载到本地目录结果如下截图：



STEP 4：训练过程



（一）数据预处理

下载的数据集本身是有固定结构的，不像爬虫数据，所以不需要进行数据清洗，但需要将数据加载为Tensor格式并实例化DataLoader。

（二）分词

使用BertTokenizer 编码成Bert需要的输入格式。

（三）定义神经网络

加载预训练模型、定义Bert模型、定义线性函数、将768维的向量输入到线性层映射为二维向量。

（四）定义训练函数

定义模型的损失函数、梯度清零；在训练函数中每2步进行一次打印，每50步调用一次验证函数进行模型效果验证。

（五）设备配置

如果有可用的cuda则将模型加载到cuda训练，若无则用cpu。

（六）实例化模型

将模型、训练集、验证集均载入验证函数进行训练与验证。

（七）导出训练好的模型

当训练数据都训练完成时，模型训练也完成，导出和保存此时训练好的模型。

到这里，模型的训练已完成，花费训练耗时：3小时左右(CPU环境)。

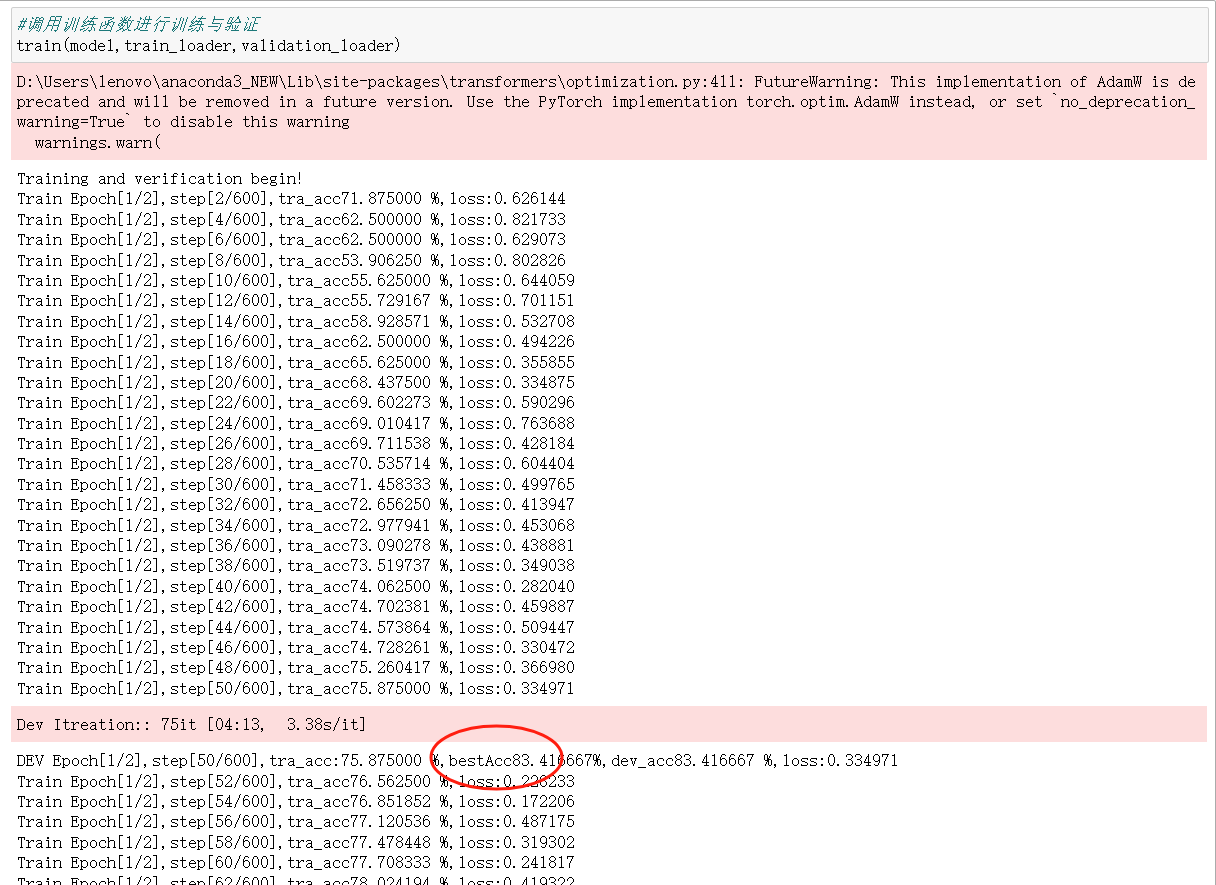
（八）定义预测函数

定义预测函数，并将训练好的模型和测试集载入预测函数进行预测。

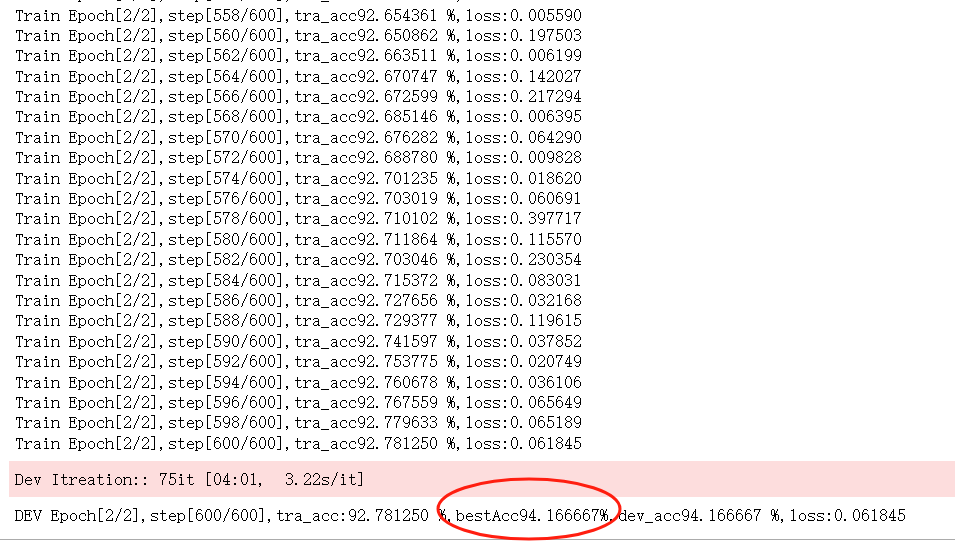
1. 查看效果

1、模型训练效果

下列两图分别表示模型刚开始训练时的日志输出和经数据集训练完成后的日志输出：



（output图一）

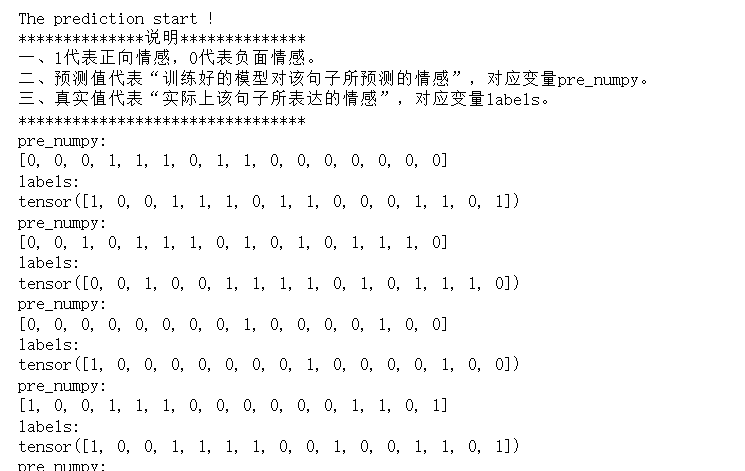


（output图二）

通过比对观察上图一和图二，我们发现，对于训练集（Train开头），最开始的模型训练准确度tra\_acc只有70%左右，损失率loss也达0.70，而到训练后期，准确度tra\_acc已达92%，而损失率下降到0.05上下；对于验证集（DEV开头），预测准确度bestAcc由83.41%提高至94.16%。

说明通过数据集的训练，模型对文本情感分析的结果准确度已有较高的提升，产生了更好的效果。

2、模型测试效果



2.1日志输出头部有些必要说明：

The prediction start !

\*\*\*\*\*\*\*\*\*\*\*\*\*\*说明\*\*\*\*\*\*\*\*\*\*\*\*\*\*

一、1代表正向情感，0代表负面情感。

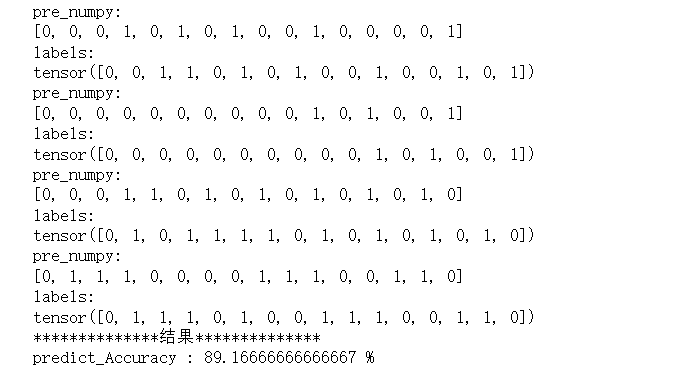
二、预测值代表“训练好的模型对该句子所预测的情感”，对应变量pre\_numpy。

三、真实值代表“实际上该句子所表达的情感”，对应变量labels。

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

2.2 可以通过查看pre\_numpy和labels的值（0/1）可以比较直观看到预测结果和真实值之间的差异，他们展示的数组长度均相同，且同一位置的值代表对同一句子的情感打分。同时，因为测试集的数据并未打乱（shuffle=False），所以将所有pre\_numpy或者labels依次前后拼接起来，正好对应dataset\_test（测试集）中从前到后的数据，尤其是”text”文本数据。

2.3 模型测试结果如下图，



之前有几次的测试结果达90%以上，后来再测几次都稳定保持在89.16%的数值上。

**三、优化与展望**

在完成模型训练的过程中，我们发现本项目还有一些可优化的地方，比如以下4点：

1、安装CUDA，把模型训练放到GPU上执行，提高训练效率。

2、模型训练时可同时对模型进行微调(fine-tune)，能提高训练后的模型的预测表现。

3、添加效果图，即将一些中间过程用图来展示，效果会更清晰直观，比如将训练过程中输出的预测准确率tra\_acc和损失率loss的走势都以动态图的方式展示变化，或者将预测值和真实值分别以不同颜色在同一坐标轴上描绘成两条曲线，两条曲线重合的点越多，说明模型效果更好等(实现自动画图)。

4、功能完善，向模型输入一个评价(中文句子)，则模型会直接返回预测值及该预测值的准确率。

对文本挖掘课程学习与作业项目的研究和完成让我们对模型训练与当下热门且越来越主流的人工智能有了更新的了解，虽然今年本课程已经结束，但是学习没有结束。以本项目出发，相信后续对本项目的优化研究和对文本挖掘的继续探索能让我们对本课程本专业有更进一步的熟悉与体验。