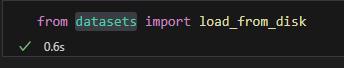
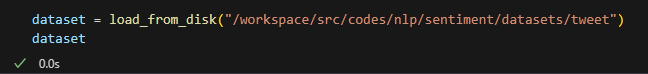
首先导入包datasets中的load\_from\_disk

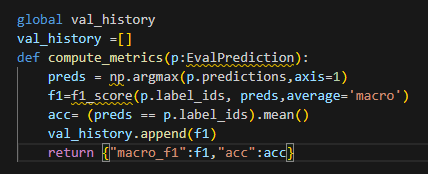
datasets是NLP中技术的开源供应商开发的数据集加载库。其格式可能有所不同，但是有相关文档可以参阅。



导入本地的数据集



计算指标函数



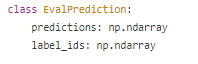
首先声明全局变量val\_history并置为空值。

定义函数，传入EvalPrediction类型的数据。

EvalPrediction

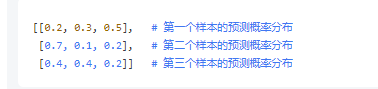
需要引入transformers库，这是Huggingface开发的库。Transformers提供了数以千计针对于各种任务的预训练模型模型，开发者可以根据自身的需要，选择模型进行训练或微调，也可阅读api文档和源码，快速开发新模型。

其包括两个变量：



predictions表示模型预测的值（二维数组），而label\_ids表示真实的值（一维数组）。

比如说，一共有0,1,2三种结果，则predictions为



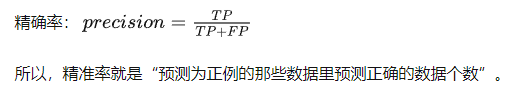
使用argmax函数，设定axis=1，算出每行最大值索引，作为预测结果。

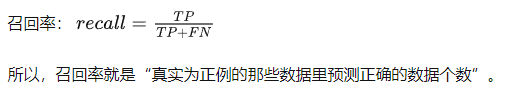


[F1分数](https://so.csdn.net/so/search?q=F1%E5%88%86%E6%95%B0&spm=1001.2101.3001.7020)（F1-score）是分类问题的一个衡量指标。一些多分类问题的机器学习竞赛，常常将F1-score作为最终测评的方法。它是精确率和召回率的调和平均数，最大为1，最小为0。

F_{1}=2\cdot \frac{precision\cdot recall}{precision+recall}

精确率和召回率：

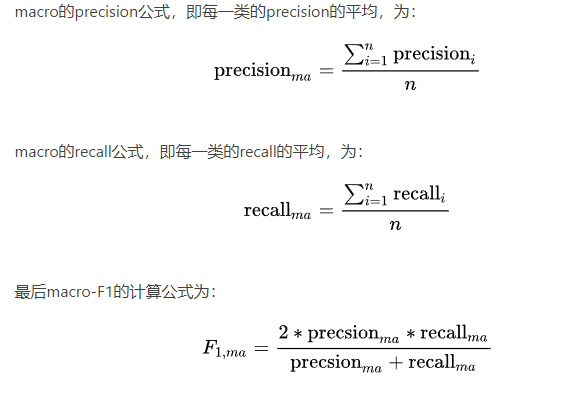




在这里，使用sklearn库中的f1\_score，以达到衡量模型的指标



注意：average=macro的意思是：



计算准确度，并返回





加载数据

图形用户界面, 文本

描述已自动生成

python中三引号可以将复杂的字符串进行复制:

python三引号允许一个字符串跨多行，字符串中可以包含换行符、制表符以及其他特殊字符。

三引号的语法是一对连续的单引号或者双引号（通常都是成对的用）。

读取数据，不概述了

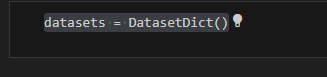
文本

描述已自动生成

读取结果：



创建了一个名为datasets的DatasetDict对象。



DatasetDict是Hugging Face库中的一个数据结构，用于管理多个数据集（datasets）。它类似于Python的字典（dict），但是专门设计用于处理多个数据集，并提供了一些方便的功能和方法。

将数据从文件中取出并读入Dataset中。

文本

描述已自动生成

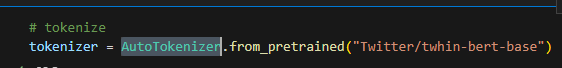
最终结果：

文本

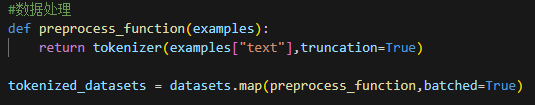
描述已自动生成

使用AutoTokenizer来加载模型

tokenizer进行分词。其主要的任务是将文本输入转化为模型可以接受的输入，因为模型只能输入数字，所以 tokenizer 会将文本输入转化为数值型的输入



进行数据预处理：也就是使用分词器将词汇分开。



可以看出，数据在处理后转化为了数字

文本

描述已自动生成



Attention Mask：

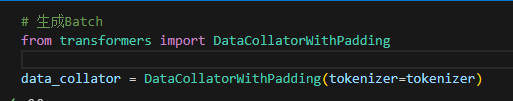
在自然语言处理中，Attention Mask是一个与输入对应的二进制向量（0,1），用于指示模型在进行注意力机制时要关注哪些位置，而忽略哪些位置。

对于序列任务（例如文本分类、命名实体识别等），输入通常由单词或字符组成，并且长度可能不一致。为了使所有的输入具有相同的长度，我们需要进行填充（padding）操作。填充后的序列通常包含特殊的填充符号（如0），以示区分。

Attention Mask的作用就是告诉模型哪些部分是真实的输入内容，哪些部分是填充的。它通常是一个与输入序列相同长度的二进制向量，其中1表示真实的输入，0表示填充。

在本次学习中，使用的模型支持不同长度输入，故不需要1,0表示，默认全为1。

生成data\_collator



准备模型

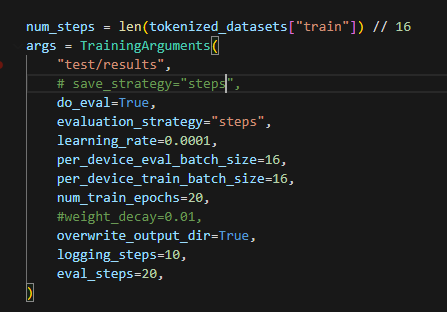
文本, 日程表

描述已自动生成

并定义名称

0 为负面的，1 为中立 2为 积极的。

选择参数：



TrainingArguments是Hugging Face的transformers库中的一个类，它用于定义训练过程中的各种参数。以下是你提供的代码中使用的参数及其含义：

"test/results": 指定训练结果的保存路径。

do\_eval=True: 设置为True时，在训练过程中会进行评估。

evaluation\_strategy="steps": 设置评估策略为按步骤进行评估。

learning\_rate=0.0001: 学习率的初始值设为0.0001。

per\_device\_eval\_batch\_size=16: 每个设备上用于评估的批处理大小为16。

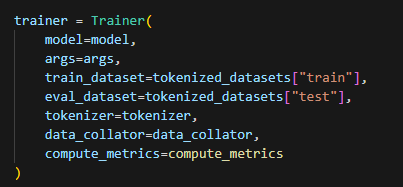
per\_device\_train\_batch\_size=16: 每个设备上用于训练的批处理大小为16。

num\_train\_epochs=20: 训练轮数设置为20。

overwrite\_output\_dir=True: 如果设置为True，则会覆盖以前保存的结果。

logging\_steps=10: 每隔10个步骤记录一次日志信息。

eval\_steps=20: 每隔20个步骤进行一次评估。



设置模型参数并开始训练。

图形用户界面, 文本

描述已自动生成