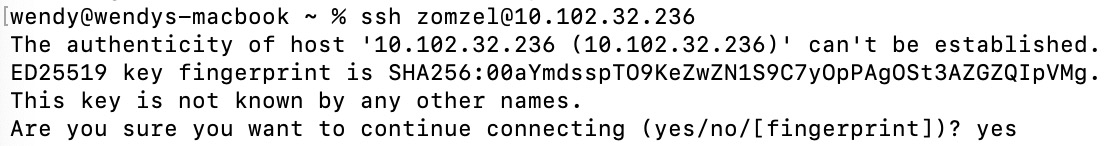
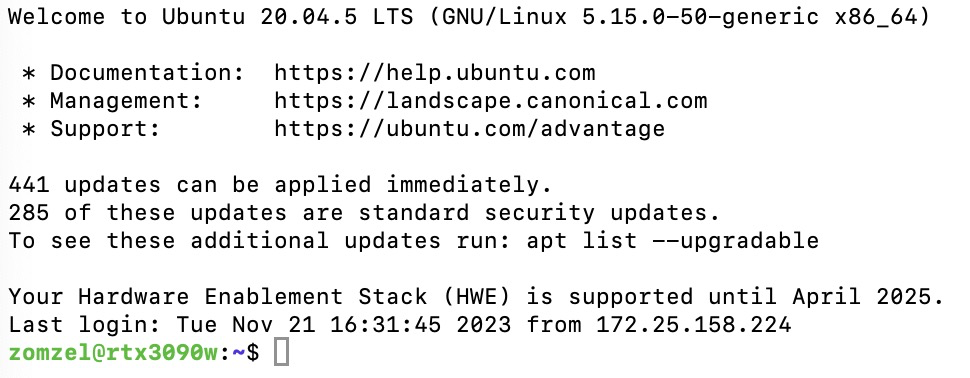
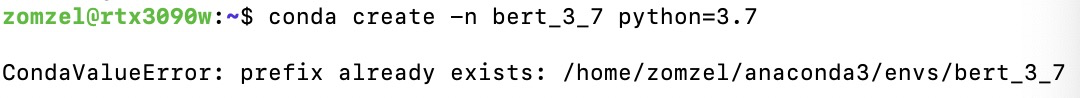
1. Bert环境配置
2. Mac远程服务器部署
3. 打开终端，输入ssh username@hostname命令，如果是首次连接到远程服务器，终端会询问是否继续连接并显示指纹信息。输入yes确认连接。

（这里的服务器是我组组长周明政同学提供的！感谢！）

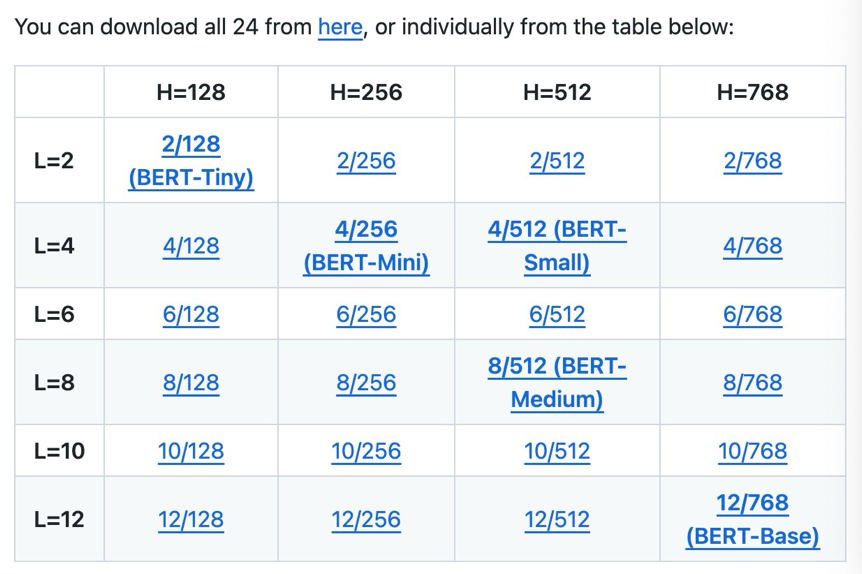
1. 接下来，按提示输入密码即可。
2. 出现如下提示即表示连接成功。
3. 创建虚拟环境运行bert（组长同学已经建好了，再次感谢！）

创建了一个新的项目文件夹。

部署好了服务器！

但是对于小规模的任务和较小的bert模型，只需在本地运行BERT即可，所以这次还是没有重新下pycharm专业版配置环境，效果其实还可以。

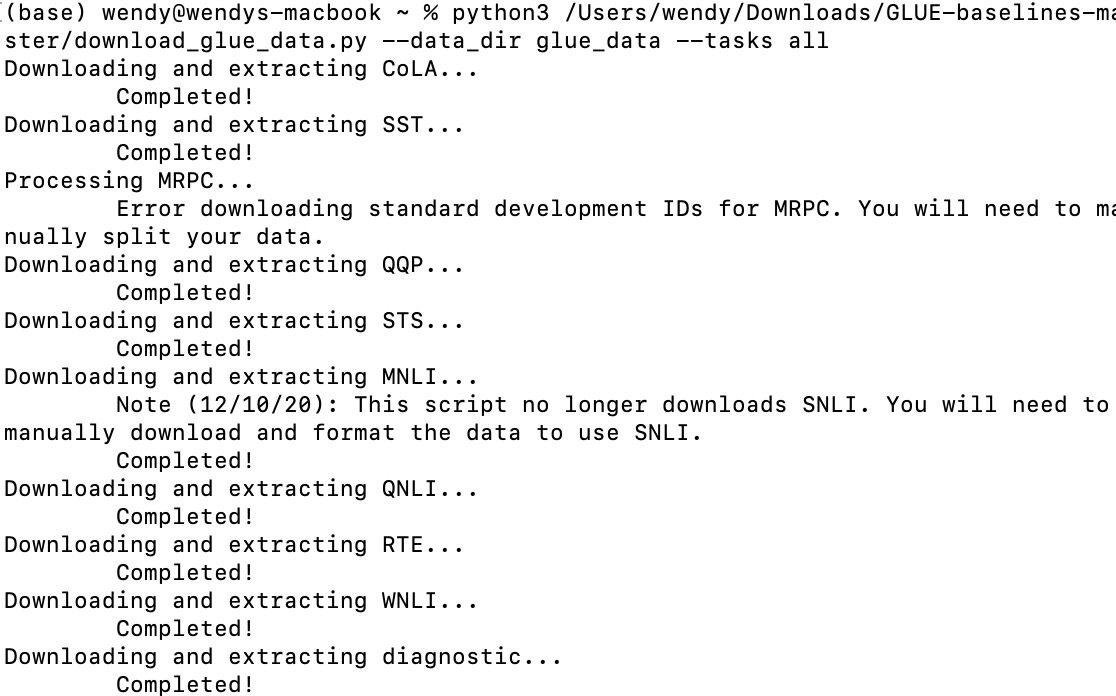
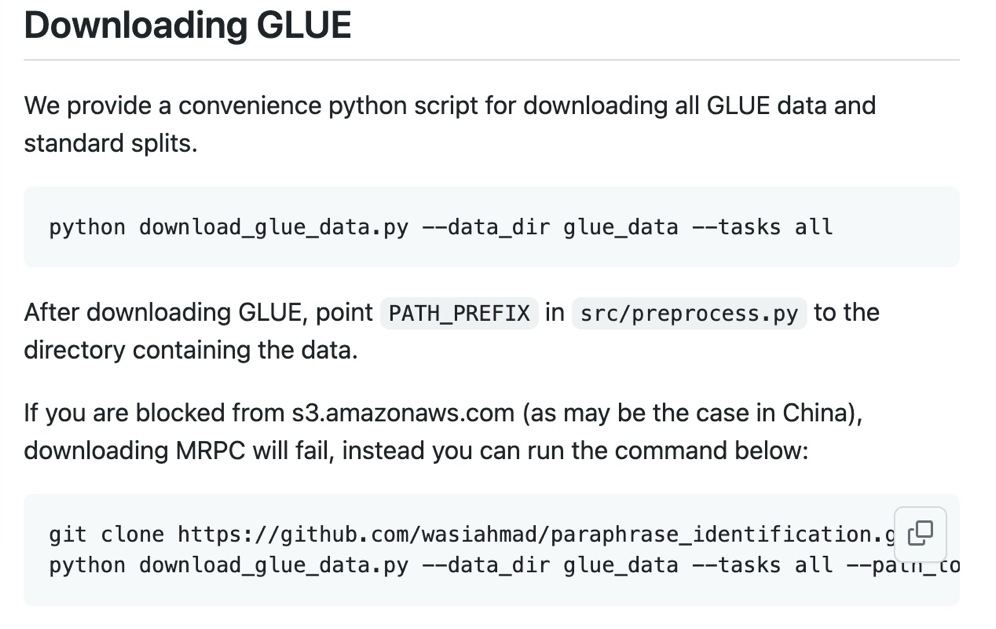
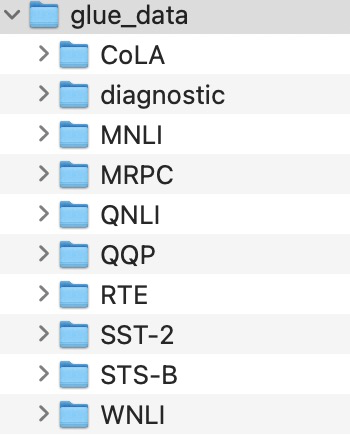
1. 下载BERT项目代码、模型及数据文件到本地，在pycharm中打开
2. 下载BERT源码及模型文件

<https://github.com/google-research/bert>

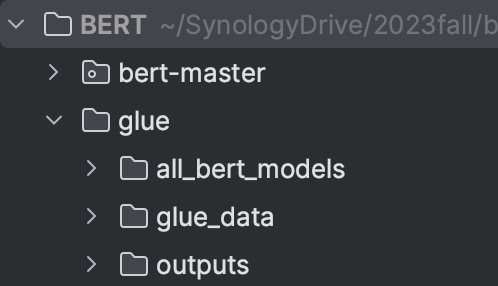
我把上面24个都下了，还顺带下了中文的模型。

1. 本地新建BERT文件夹存放项目文件，包括bert-master源码文件夹，glue文件夹存放模型文件all\_bert\_models、测试数据glue\_data及模型输出outputs。
2. 下载测试数据集GLUE(General Language Understanding Evaluation), 这是一个广泛使用的自然语言处理（NLP）评估基准，旨在评估语言理解模型在多个任务上的性能。

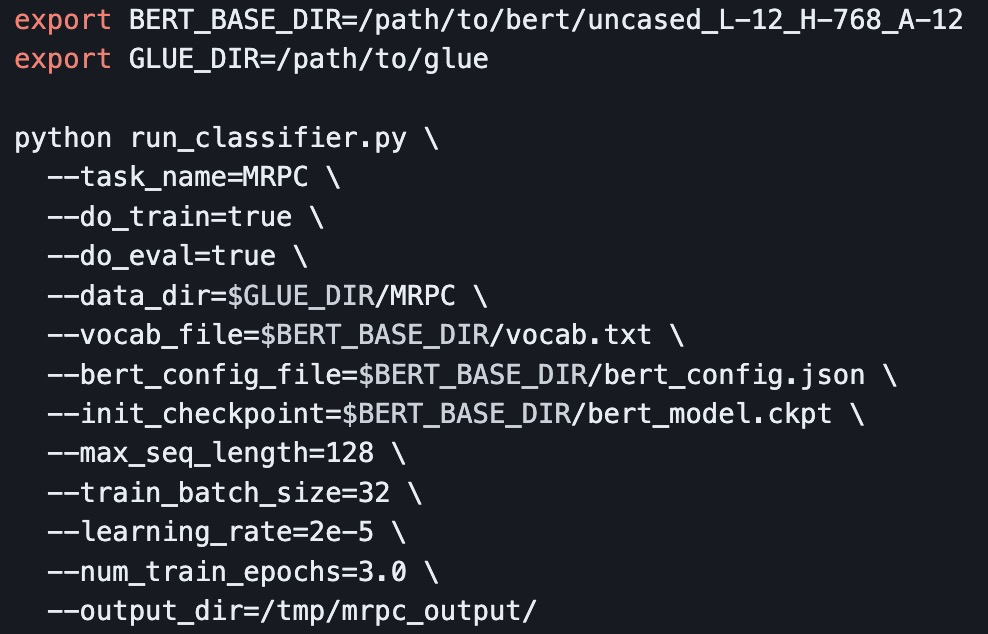
<https://github.com/nyu-mll/GLUE-baselines>

在pycharm中把这个项目打开后，长这样：

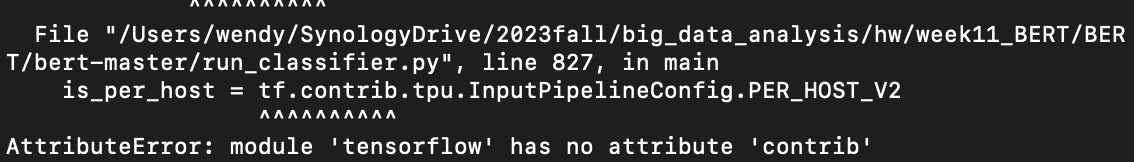


1. 使用BERT模型对sentence分类
2. 训练分类器



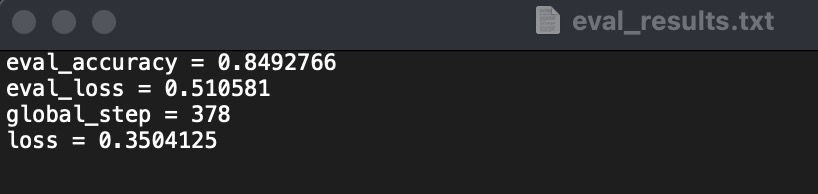
把路径换成自己的文件夹即可。

但是在本机上跑的时候，出现了类似如下的TensorFlow版本不兼容的各种问题：



故尝试conda了一个虚拟环境配置TensorFlow1.0。

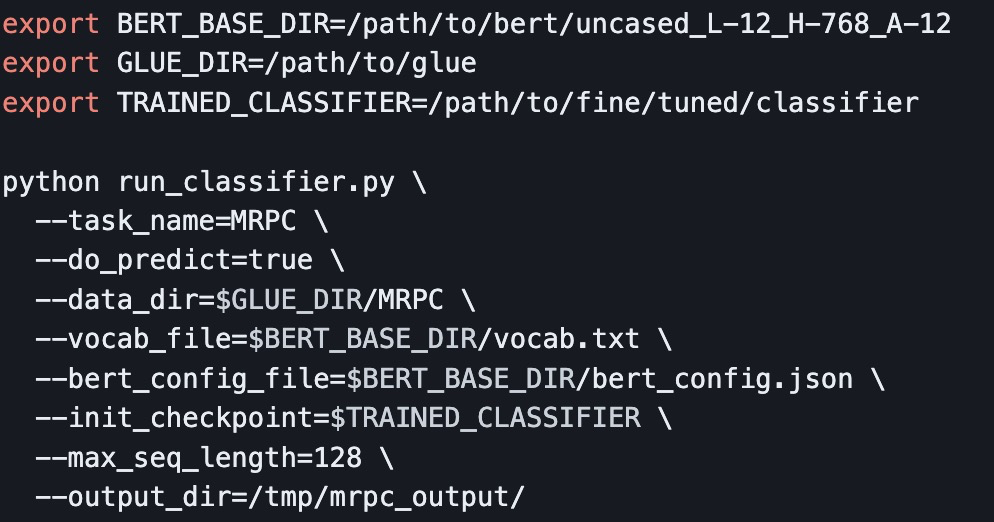
重新运行run\_classifier.py，训练分类后得到一个txt文件：



说明Dev集的准确率为85%左右

1. 使用分类器进行预测

终端中运行如下命令：



同样把export的路径换成自己的文件夹路径。

运行结束后，可以在outputs文件夹中找到test\_result.tsv。



1. 总结
2. BERT是一种预训练的NLP模型，可以从全称“Bidirectional Encoder Representations from Transformers“看出其特点，也就是使用MLM（Masked Language Model）和NSP（Next Sentence Prediction）进行预训练，捕捉句子中的上下文信息，并生成高质量的词向量表示。

在预训练阶段，BERT能学习到丰富的语言表示，并能在下游的NLP任务中（如文本分类，句子关系判断等）得到应用。注意这个预训练模型需要在特定任务的输出层进行微调，以适应不同任务的需求。

Hugging Face Transformers库提供了各种预训练的语言模型，包括BERT、GPT、RoBERTa等，还支持多种深度学习框架，如 TensorFlow、PyTorch 等，并提供了简单易用的API来微调这些模型。如果后续想继续使用BERT的话，可以用Hugging Face Transformers将其应用于不同框架。

1. pycharm社区版没有提供Deployment>Configuration>Add New Configuration的功能，还是需要下载professional版。不过目前的项目在本机上运行其实是足够的。