一、BERT 简介

BERT是一种预训练语言表示的方法，在大量文本语料（维基百科）上训练了一个通用的“语言理解”模型，然后用这个模型去执行想做的NLP任务。BERT比之前的方法表现更出色，因为它是第一个用在预训练NLP上的无监督的、深度双向系统。

无监督意味着BERT只需要用纯文本语料来训练，这点非常重要，因为海量的文本语料可以在各种语言的网络的公开得到。

预训练表示可以是上下文无关的，也可以是上下文相关的，而且，上下文相关的表示可以是单向的或双向的。上下文无关模型例如word2vec或GloVe可以为词表中的每一个词生成一个单独的“词向量”表示，所以“bank”这个词在“bank deposit”（银行）和“river bank”（岸边）的表示是一样的。上下文相关的模型会基于句子中的其他词生成每一个词的表示。

BERT建立在最近的预训练相关表示工作之上——Semi-supervised Sequence Learning, Generative Pre-Training, ELMo,和ULMFit——但是关键是这些模型都是单向的或浅双向的。这以为意味着每个词之和它左边或右边的词相关。例如，在句子“I made an bank deposit”中，“bank”的单向表示只基于“I made a”而没有“deposit”。一些以前的工作也有结合了单独的左上下文和右上下文的，但只是用了一种简单的方式。BERT同时用左边和右边内容表示“bank”——“I made a … deposit”——从一个深度网络非常底层就开始了，所以说，他是deeply bidirectional（深层双向）的。

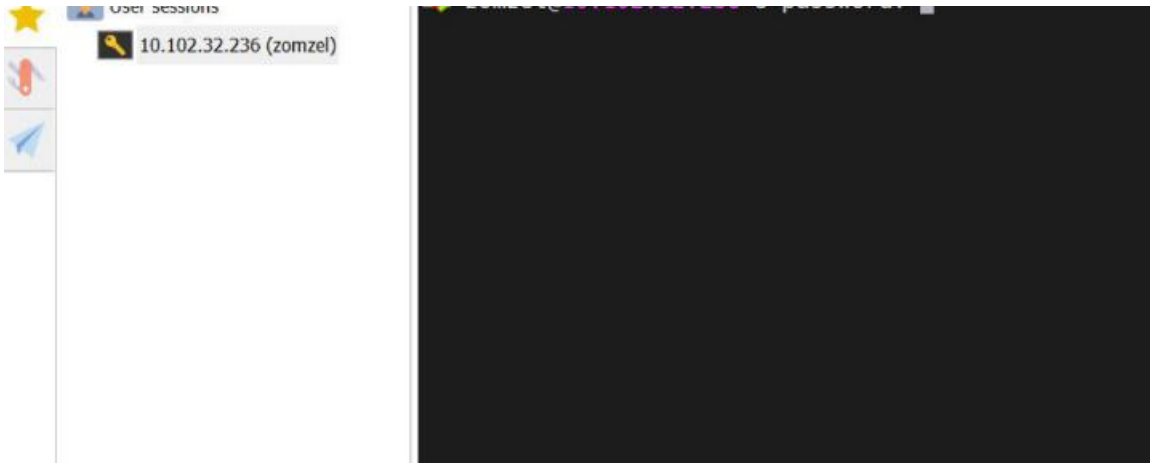
二、 BERT 环境配置

1、远程服务器部署（电脑跑不动，用远程服务器跑 ）

(1) MobaXterm

mobaXterm 用于远程服务器的连接， 官网下载免费的就行直接可用

借用组长服务器



使用 conda 创建运行 bert 的虚拟环境:
conda create -n bert_3_7 python=3.7
激活虚拟环境:
source activate bert_3_7

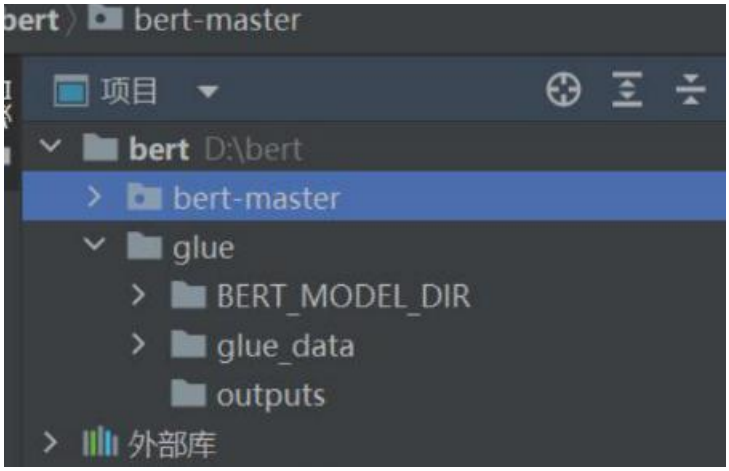
(运行需要的包不急着下)

创建一个文件夹用于与本地项目文件进行映射
mkdir bert

2、 BERT 项目代码和模型 git 到本地，在 pycharm 中打开

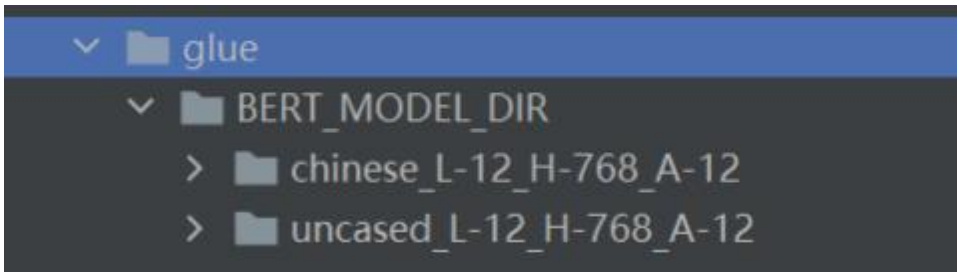
(1) 下载 bert 源码

(2) 本地新建一个文件夹 bert 用于存放项目文件



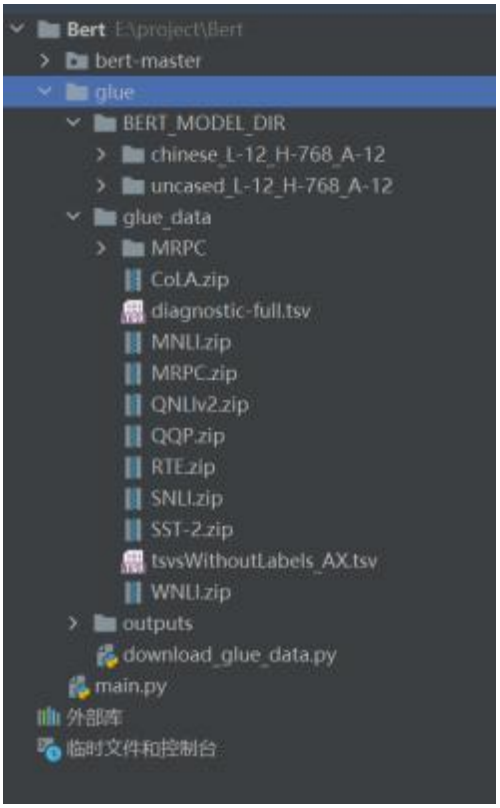
(3) BERT 的 github 主页下载预训练模型

下载 Base 基础的预训练模型就行， 下载后将压缩包解压到刚刚建立的 BERT_MODEL_DIR 文件夹中， 如下图所示 (这里多下载了一个 chinese 的不 需要管他)：



(4) 下载语料库数据

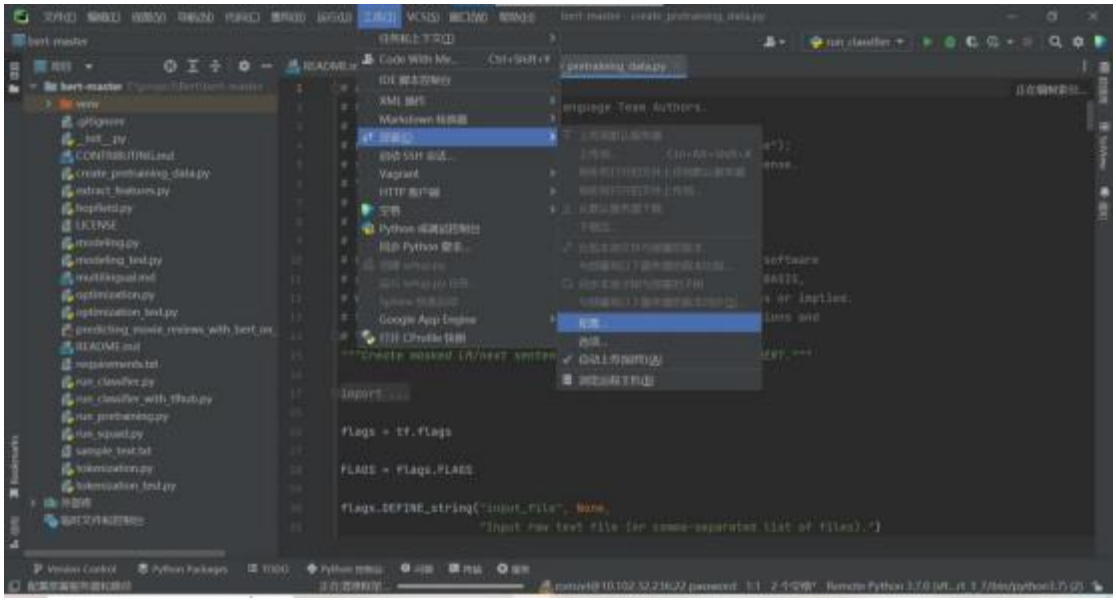
将 glue_data 解压到 glue 文件夹中 (好像之前创建 glue_data 文件夹多余 了) 然后将 glue_data 中的 MRPC 压缩包解压到当前文件夹。至此所有文件 准备完毕,如下图所示:



3、 运行配置

在 pycharm 中部署远程服务器，让远程服务器来跑我们的代码

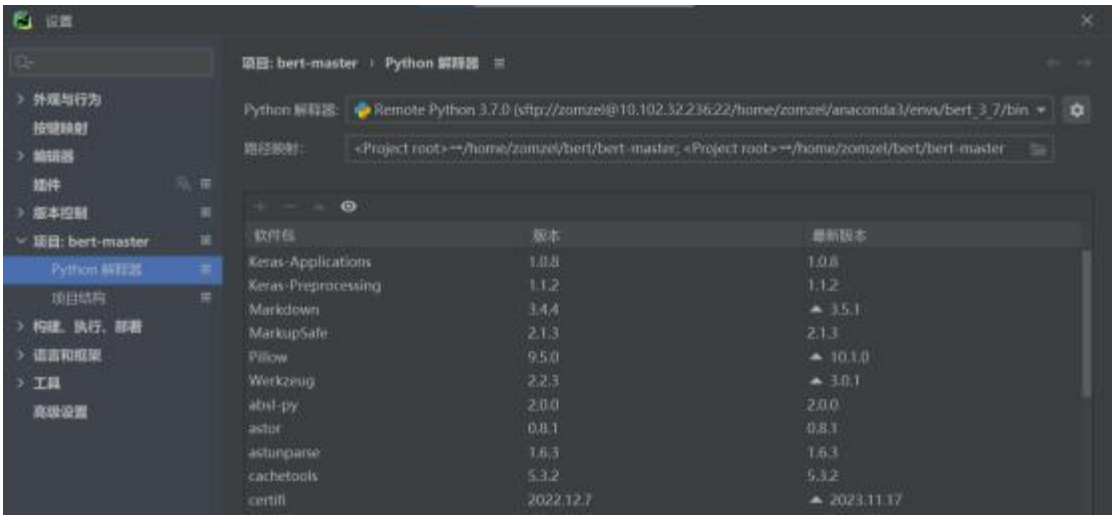
(1) 选择工具->部署->配置



(2) 点加号进行添加部署，选择 SFTP 类型，将服务器命名为 i_BERT(随意)，点 击确定进行下一步的配置。

(3) 配置解释器

为项目代码配置解释器， 文件->设置->项目->解释器



点击小齿轮添加解释器， 选择 ssh 解释器和现有的远程服务器配置

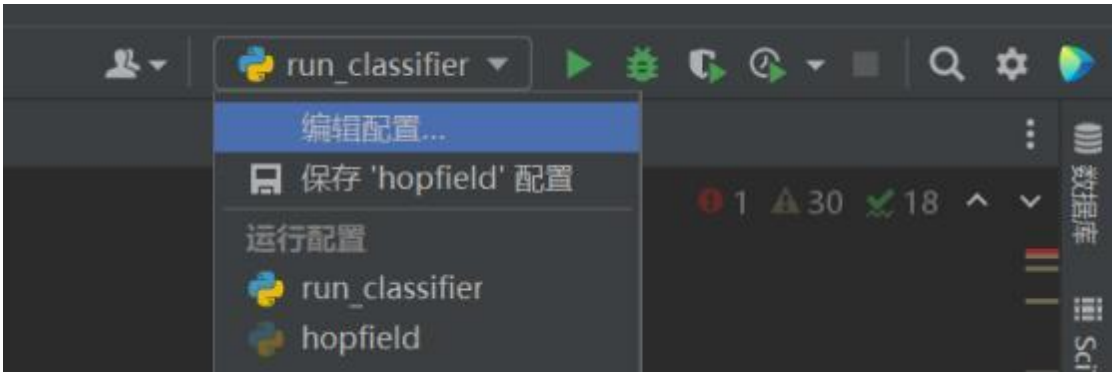
三、使用 BERT 预训练模型进行 Sentence 分类任务

1、在远程服务器上的命令行进入 bert 文件夹，此时源码文件全部上传到了该文件夹，包括 requirement.txt 文件，使用命令：

pip install -r requirement.txt

来下载所有必须的依赖包，可能有些包会下载失败，再用 pip 手动安装就行，如果 还失败可以尝试下载不同的版本

2、配置运行参数。在 pycharm 上进行运行的配置。

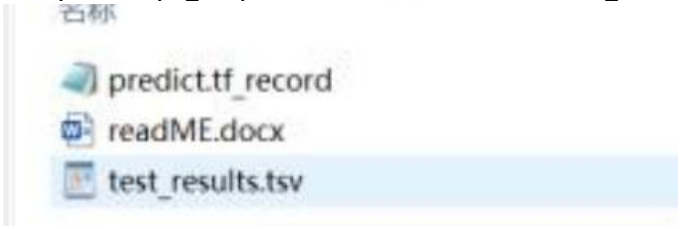


运行文件选择 run_classifier,py,然后输入运行的形参，直接复制下面的配置丢入形参配置

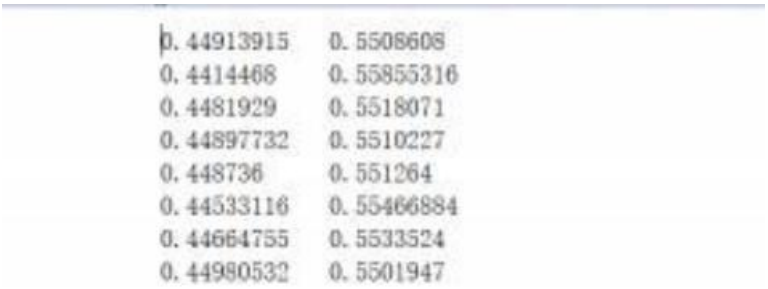
3、然后就可以直接运行 run_classifier.py 进行训练
在 outputs 文件夹能找到 eval_result.txt 文件， 内容如下:

模型训练成功！ 正确率和损失值与官方给出的数据相差不大。

4、使用训练模型和测试数据进行句子分类的预测。
使用测试任务的参数配置重新跑 run_classifier.py 进行预测。(时间比训练短一些) 在 /outputs/mrpc_output 文件夹中找到预测输出 test_result.tsv



其内容如下:



0.44913915	0.5508608
0.4414468	0.55855316
0.4481929	0.5518071
0.44897732	0.5510227
0.448736	0.551264
0.44533116	0.55466884
0.44664755	0.5533524
0.44980532	0.5501947

四、 总结

1. 多功能性：BERT预训练模型展现出卓越的多功能性。它不仅能够成功进行文本分类和预测任务，还能够执行其他复杂任务。例如，它具备填充掩码（Masked Language Model，MLM）的能力，可以识别并还原被掩盖的词汇，从而还原原本的句子。此外，BERT还具有下一句预测（Next Sentence Prediction，NSP）的功能，可以推测一句话的下一句是什么，从而理解文本的连贯性和逻辑关系。

2. 鲁棒性和可扩展性：BERT模型在不同语言和多种任务上都表现出强大的鲁棒性。通过微调，我们可以轻松地将模型适应于各种语言任务，而无需从头开始重新训练。这种可迁移性使得BERT成为一个通用的自然语言处理工具，可以应用于各种应用领域，包括文本分类、命名实体识别、机器翻译等。此外，研究人员还可以尝试其他预训练模型，以探究它们在不同语言和任务上的效果，从而进一步扩展自然语言处理的研究领域。