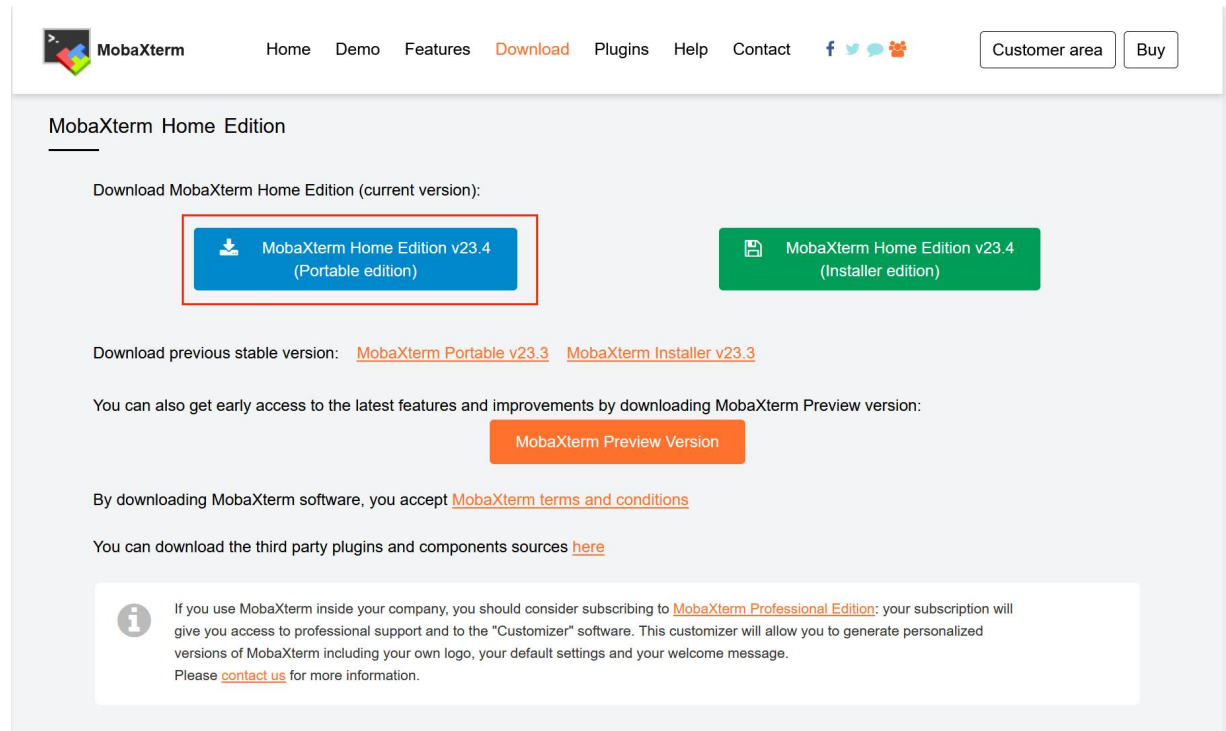


一、BERT 环境配置

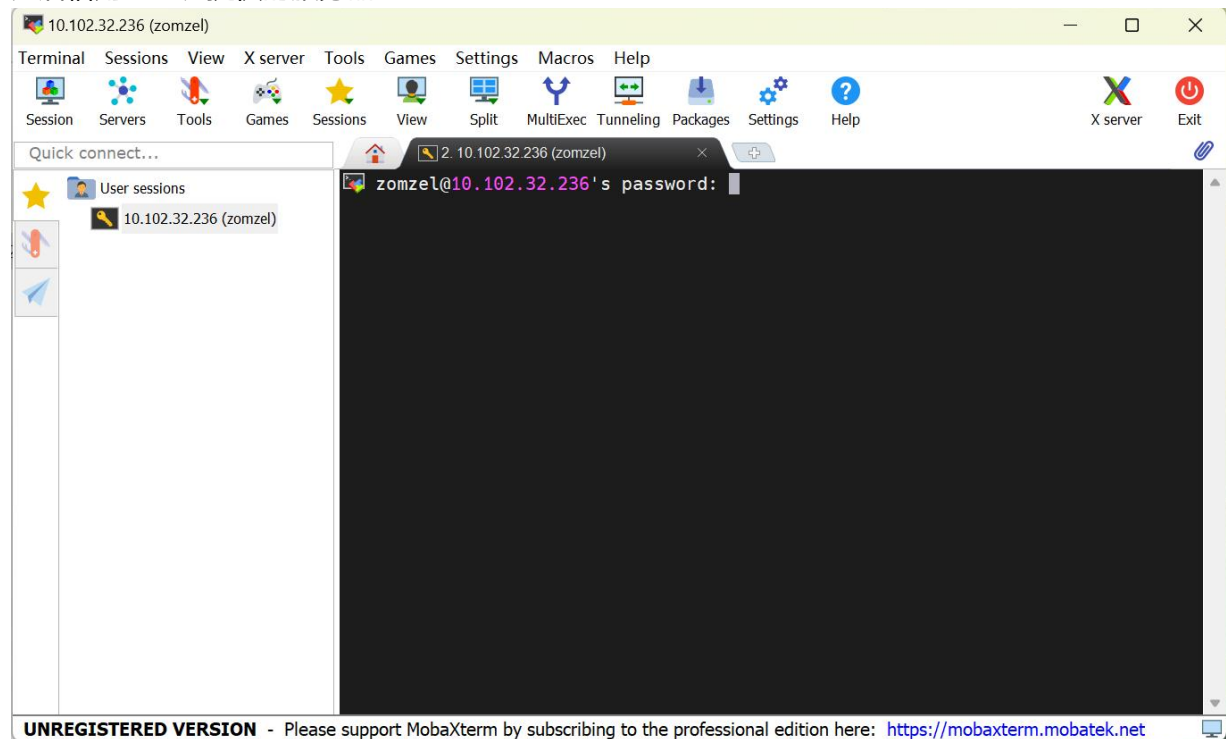
1、远程服务器部署

(1) MobaXterm

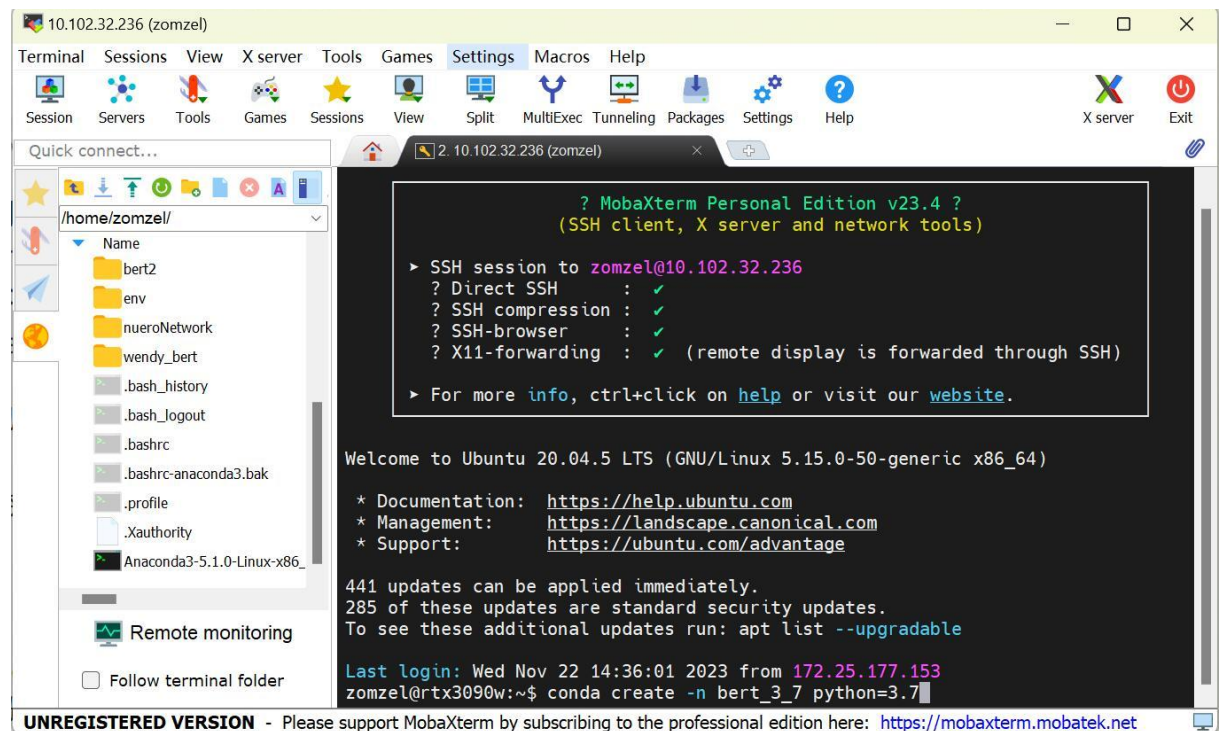
这里使用了免费便携版，解压后可以直接打开。



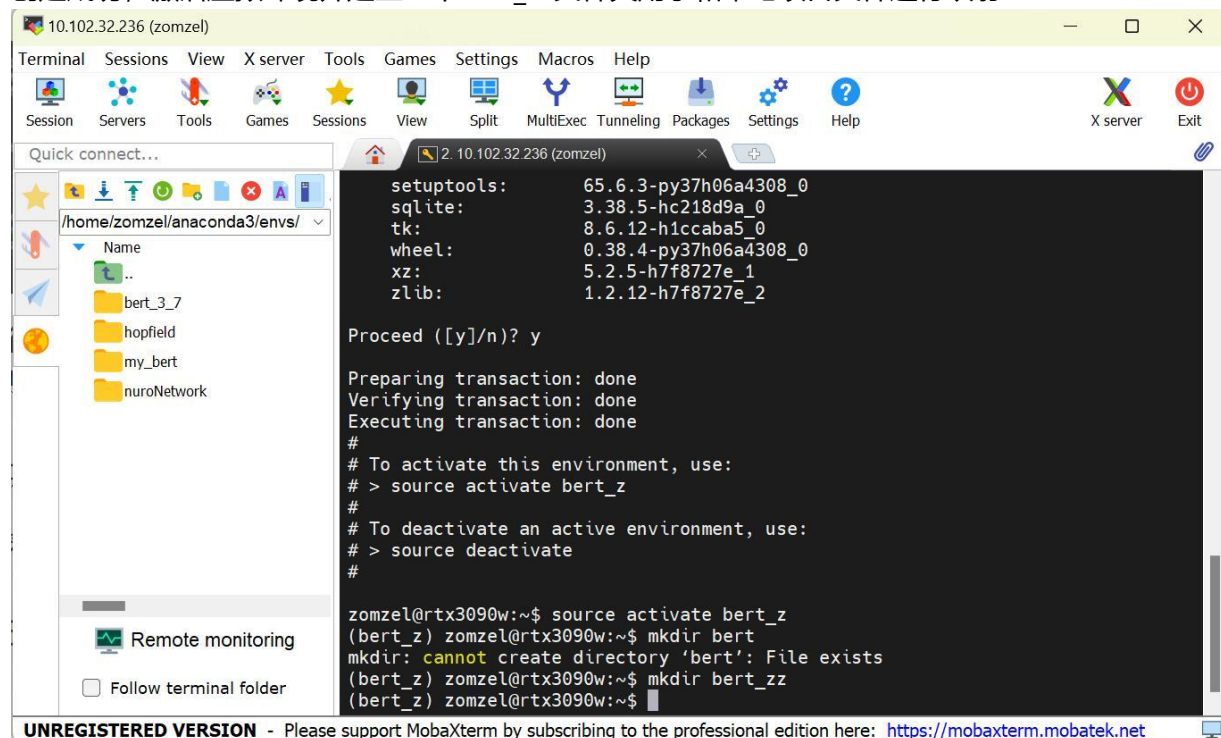
然后借用了组长提供的服务器：



连接成功，并开始创建环境：

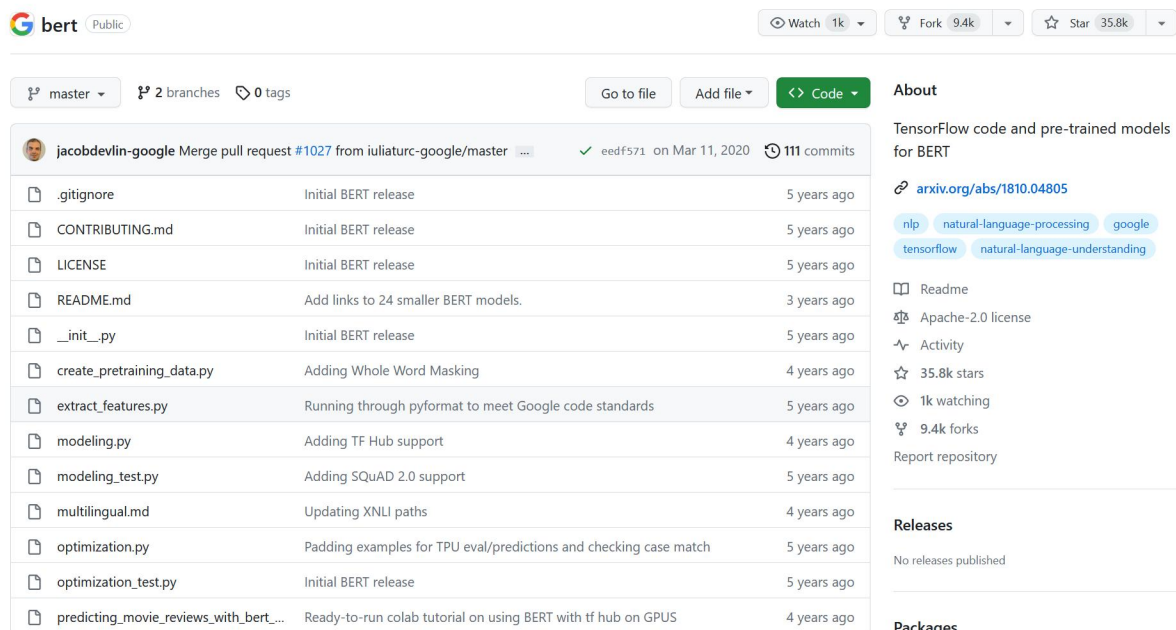


创建成功，激活虚拟环境并建立一个bert_zz文件夹用于和本地项目文件进行映射：

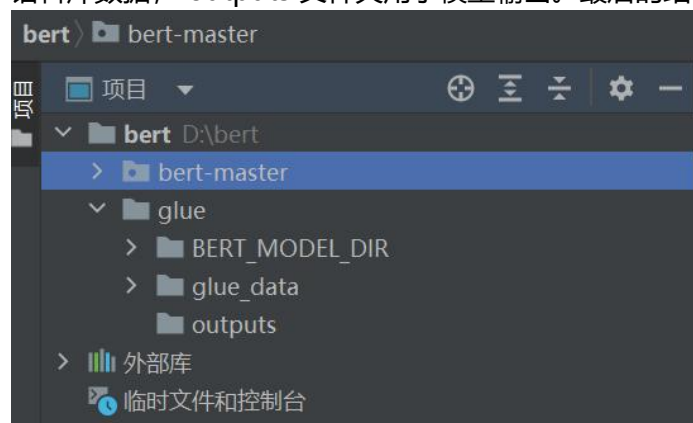


2、BERT 项目代码和模型 git 到本地，在 pycharm 中打开

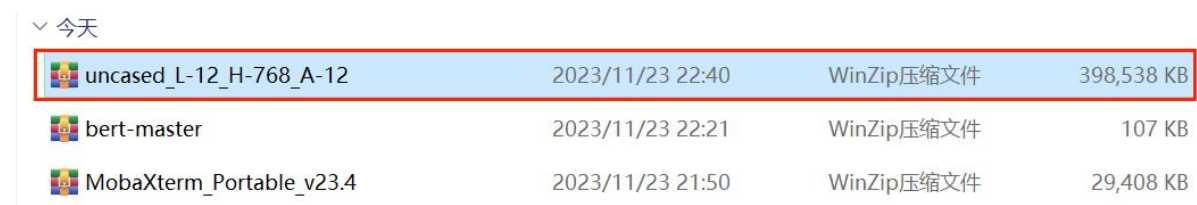
(1) 下载 bert 源码



(2) 本地新建一个文件夹 bert 用于存放项目文件，将压缩包中的bert-master解压到这个文件夹中，在 bert 文件夹中新建一个文件夹 glue 用于存放模型文件和训练测试数据，在 glue 文件夹中新建文件夹 BERT_MODEL_DIR 用于存放模型文件，新建文件夹 glue_data 用于存放训练语料库数据，outputs 文件夹用于模型输出。最后的结构如下图所示：



(3) 下载base的预训练模型并解压到glue中的BERT_MODEL_DIR文件夹中：



(4) 下载语料库

将 glue_data 解压到 glue 下的glue_data文件夹中（我这里将之前建立的替换掉了，之前不建立这里直接解压也是可以的），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。

3、运行配置

在 pycharm 中部署远程服务器，让服务器来跑下载的代码。这里根据组长提供的教程操作，完成了服务器的连接、映射、解释器等配置。

三、使用 BERT 预训练模型进行 Sentence 分类任务

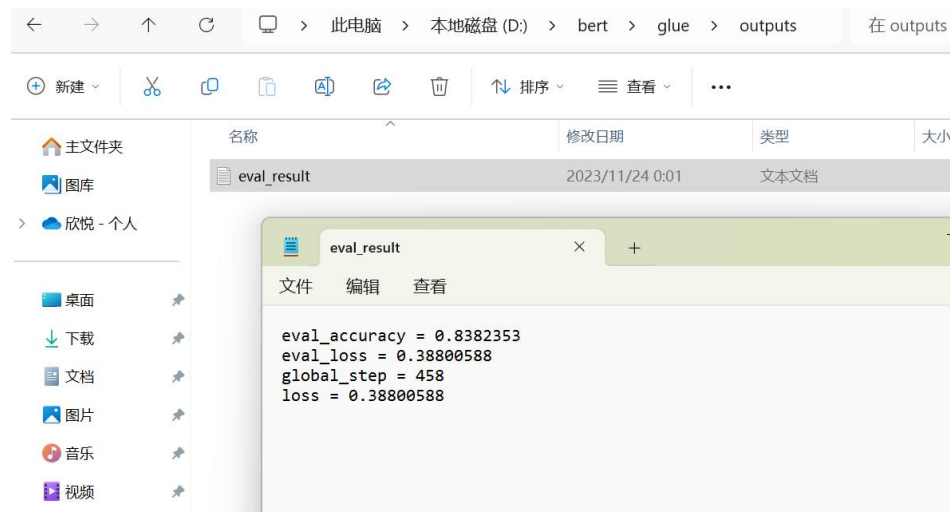
1、在远程服务器上的命令行进入 bert_zz 文件夹，此时源码文件全部上传到了该文件夹，包括 requirements.txt 文件，使用命令：

```
pip install -r requirements.txt
```

来下载所有必须的依赖包，这里很顺利地安装了所有依赖包，但忘记截图了.....

2、配置运行参数。在 pycharm 上进行运行的配置。得到模型并进行了测试后即可进行训练了。

3、运行run_classifier.py进行训练，训练结束后得到如下结果，模型训练成功。



4、使用训练模型和测试数据进行句子分类的预测。

最后在/outputs/mrpc_output 文件夹中找到预测输出 test_result.tsv

其中每一行代表一个样本属于哪个类的概率：

	run_classifier.py	test_result.tsv
1	0.44913915	0.5508608
2	0.4414468	0.55855316
3	0.4481929	0.5518071
4	0.44897732	0.5510227
5	0.448736	0.551264
6	0.44533116	0.55466884
7	0.44664755	0.5533524
8	0.44980532	0.5501947

四、总结

1、BERT 预训练模型可以很好的完成语言的分类和预测任务，不仅可以进行语句的分类还可以预测 mask 掉部分词的语句从而还原原本的句子，也可以预测一句话的下一句是什么。

2、BERT 模型具有良好的鲁棒性和可扩展性，可以很容易的通过微调将模型适应于各种语言任务。

