DL-NLP Assignment2 Report

本实验报告为 2024 年秋季学期《基于深度学习的自然语言处理》课程第二次作业实验报告，作者为 2200017416 康子熙，本报告所有代码均可以在我的 github 仓库中找到。

一、 Task 1

所有代码在我的 github 仓库中可见，代码成功运行并能够应用于后续任务中进行训练，并且训练取得了合理的结果，表明函数正确性。具体实现不在报告中提及。

二、 Task 2

2.1 Training Options

根据 notion 上的提示，我使用 transformer 库的一些函数与类，完成了基于 Bert、RoBerta、SciBert 三种预训练模型，在 restaurant_sup、acl_sup、agnews_sup 三个下游任务上各进行了五次微调，共 45 次实验。我选择了 sklearn 库提供的 micro_f1, macro_f1, accuracy 的指标作为 evaluation 的指标。五次 run 分别使用的随机种子是：42，20，13，5，2。

在训练过程中，除了和 load 的 huggingface 上的模型 config 保持一致之外，我在训练时调整 batch size=8，显存占用大小约为 3G，每次微调在每个数据集上训练共 5 个 epoch。

2.2 Experiments

在三个数据集上，三种模型都取得了比较稳定且可观的效果。然而在一些种子上存在训练无法正常降低 loss 的可能性，在确定没有其他问题的情况下，推断是随机种子选择导致的问题，而不是模型及训练方法的问题。

表 1　Performance on restaurant_sup

Model	Mean Accuracy	Std. Accuracy	Mean F1	Std. F1
SciBert	0.82625	0.00767	0.73647	**0.01153**
RoBerta	0.78196	0.07750	0.61405	0.20310
Bert	**0.84411**	**0.00738**	**0.76396**	0.01181

实验结果说明，在这三个数据集的微调上，Bert 和 SciBert 都取得了很好的效果，且效果好于 huggingface 提供的 RoBerta 模型。RoBerta 模型虽然在 acl_sup 数据集上没有

表 2　Performance on acl_sup

Model	Mean Accuracy	Std. Accuracy	Mean F1	Std. F1
SciBert	**0.83165**	0.02624	**0.74732**	0.03678
RoBerta	0.66331	0.07637	0.40275	0.19903
Bert	0.75827	**0.01656**	0.62685	**0.03602**

表 3　Performance on agnews_sup

Model	Mean Accuracy	Std. Accuracy	Mean F1	Std. F1
SciBert	0.91211	**0.00399**	0.91073	**0.00411**
RoBerta	0.91974	0.00526	0.91813	0.00531
Bert	**0.92605**	0.01126	**0.92457**	0.01152

取得很好的效果，但在另外两个数据集上的效果也同样可观。随机种子在微调时可以通过影响数据打乱等方式影响训练和测试的结果，对训练结果的影响往往并不是不可忽略的。

同时，根据实验结果，RoBerta 的训练结果较为不稳定，SciBert 和 Bert 都可以取得较为稳定的结果，这可能和选取的数据集有关，也有可能和模型的 scale 较小有关，没有办法得到和论文相似的效果。

在我的实验中，由于 sklearn 库的 micro F1 和 Accuracy 点数重复，且由于国内的网

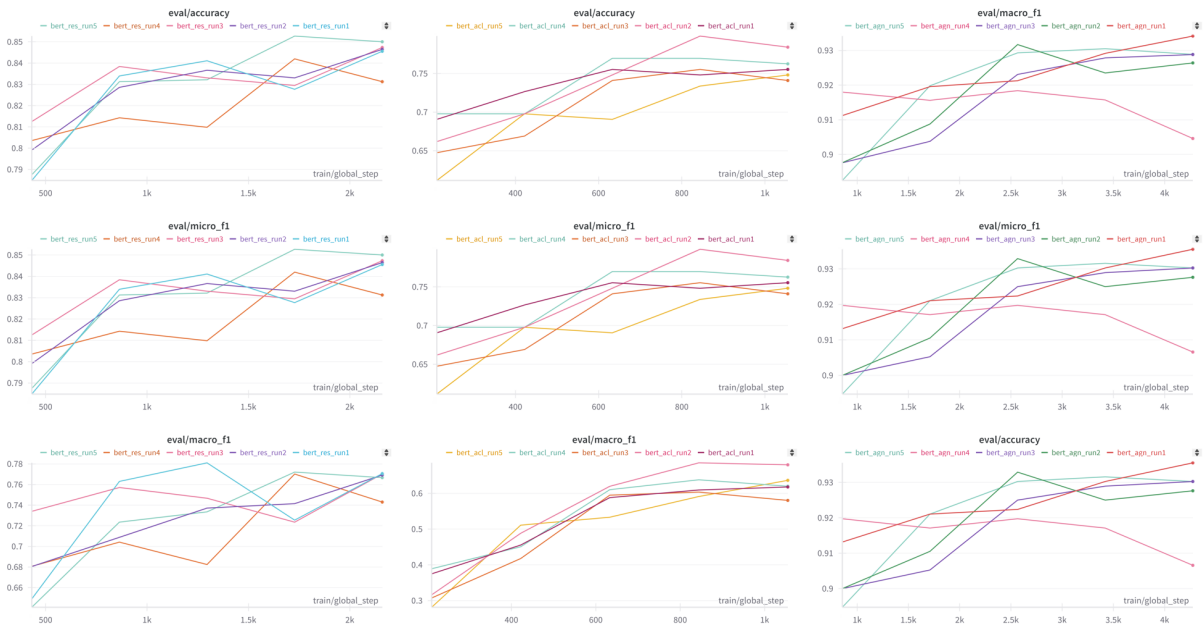

图 1　Bert Base 模型在三个下游任务上微调的效果

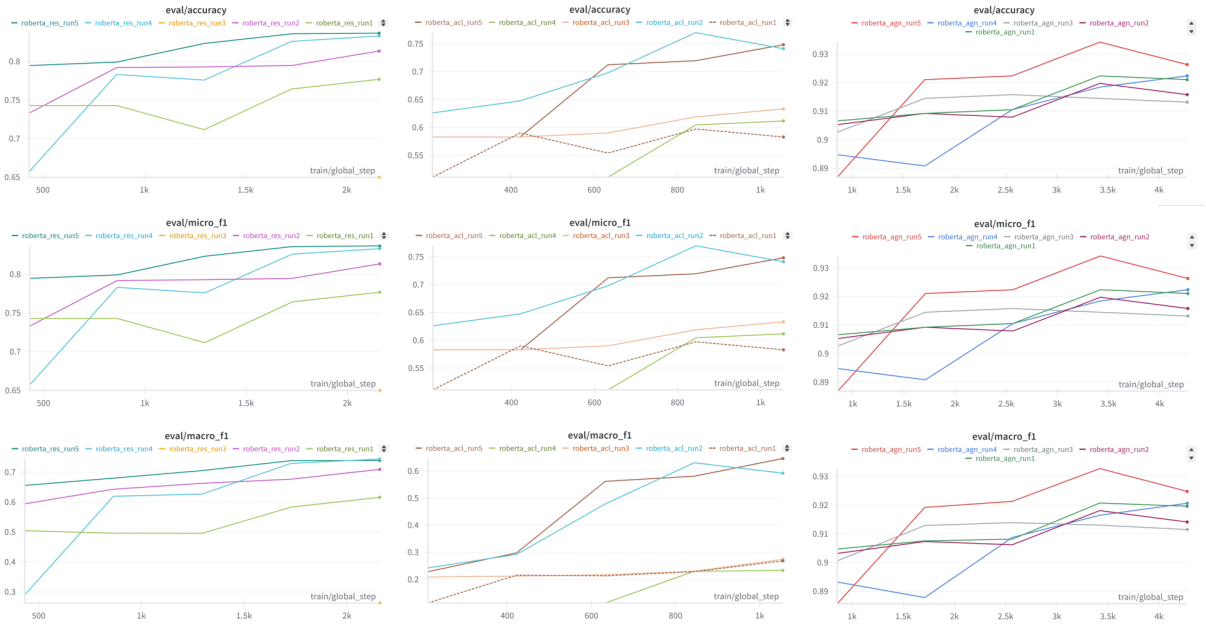

图 2　RoBerta Base 模型在三个下游任务上微调的效果

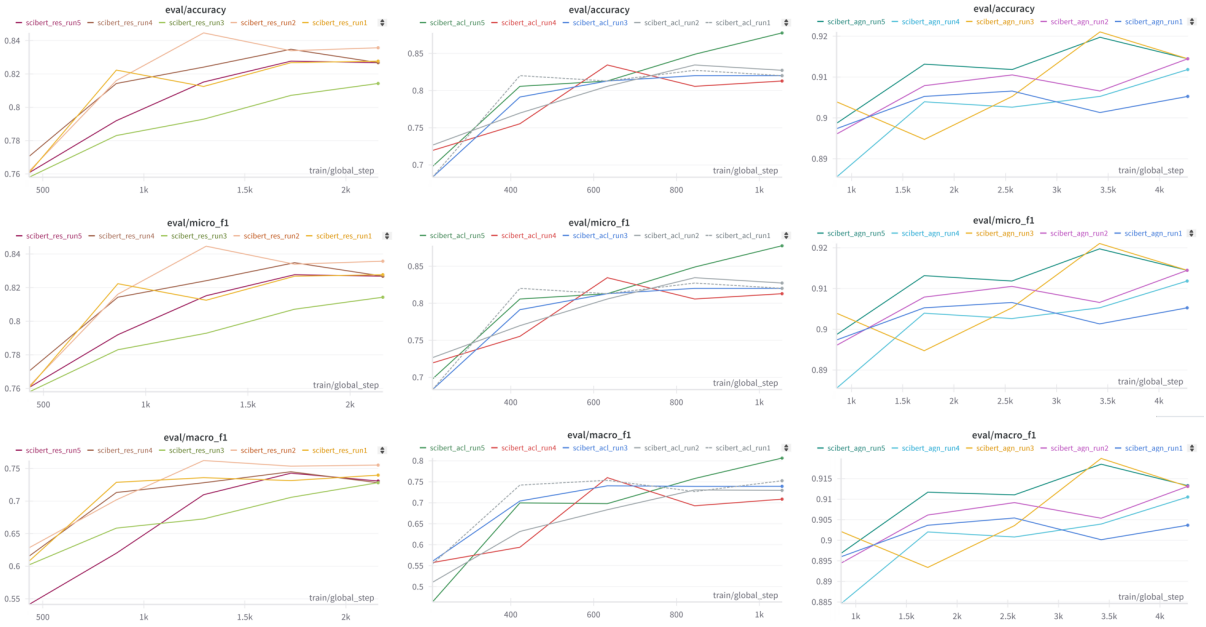

图 3　SciBert Base 模型在三个下游任务上微调的效果

络原因无法加载 huggingface 提供的 evaluation metrics，所以在表格中仅显示 macro F1 的结果。

三、 Task 3

3.1 Adapter 的选择

再明确任务以及要求后，我选用了 adapters 库中的 bottleneck adapter 作为 adapter 训练。具体可参考BnConfig 函数说明。由于算力原因，仅在随机种子 42 上进行了实验。在相同的 options 下，实验显存占用约为 2G，节省了 33%，但是相对的运行时间变成了不适用 adapter 的 3-4 倍。不过这个运行时间并不具有实际参考意义，因为我换用了 AdapterTraining 的训练方法，无法确定是否有其他因素影响了模型的训练时间。

如果不使用 Adapter 的方法，参数以浮点（32 位）存储，每个参数需要 4 字节。微调 3B 参数的模型，则参数总内存 = 3B × 4 字节 = 12B 字节。使用 Adam 优化器，通常需要每个参数两个状态变量，优化器状态总内存 = 2 × 12B 字节 = 24B 字节。激活所需的内存粗略估计为训练时参数的内存，估计为 12B 字节，则总共需要 48B 个字节，即 48G 显存。

对于 Adapter 节省了多少空间，事实上，实验已经证明了 Adapter 只需要 2G 的显存就可以训练，而它的显存大小对于 Bert Base 还是 Bert Large 来说是一样的，也就是说，Adapter 节省了至少 10 倍的显存空间，对于大模型的微调来说尤其有效。

Model	Task	Mean Accuracy	Mean F1
Adapter	restaurant_sup	0.81696	0.69319
	acl_sup	0.82321	0.69791
	agnews_sup	0.92500	0.92353
RoBERTa	restaurant_sup	0.78196	0.61405
	acl_sup	0.66331	0.40275
	agnews_sup	0.91974	0.91813

表 4　Comparison of Adapter and RoBERTa models on RES, ACL, and AGN tasks

3.2 Experiments

实验的结果和预期结果有些不同，应用 adapter 后的模型微调得到的结果比原来微调的效果更好。我分析这可能和我对于 bottleneck adapter 的设置和模型大小有关。一方面是三种预训练模型参数量较小，微调才占用 3G 显存，实际上已经和使用 BnConfig 添加的默认 Adapter 大小相近了，所以 adapter 也可以得到相近甚至更好的效果。如果继续增加模型的 scale，Adapter 就很有可能无法得到和微调全部参数相近的结果了。

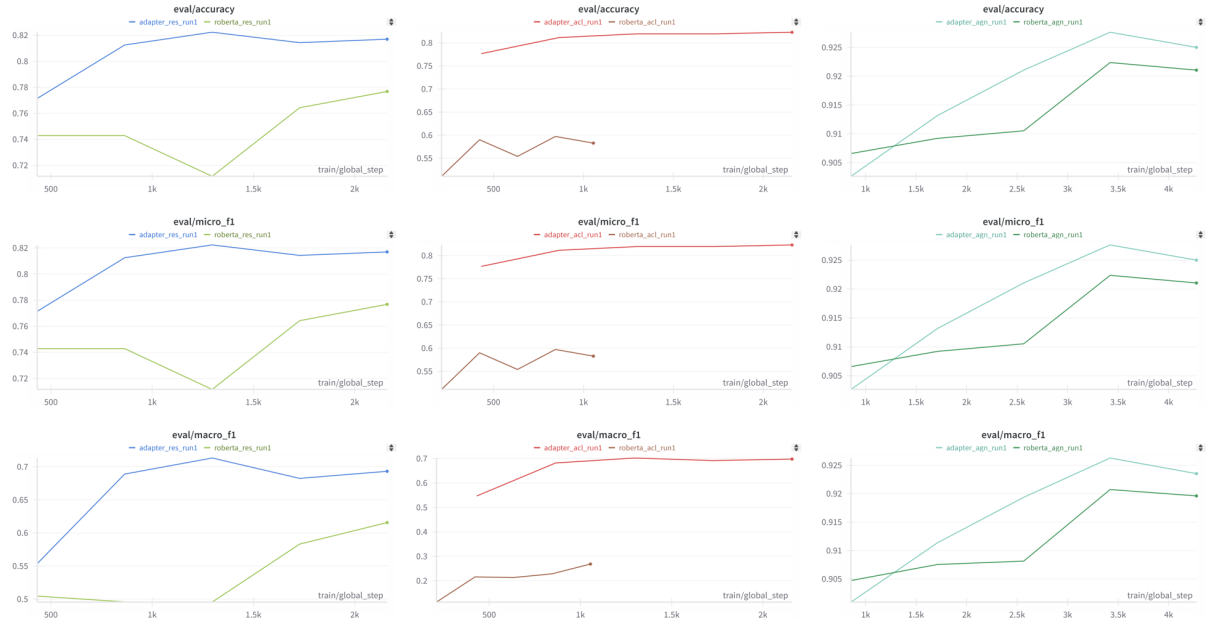

图 4　Adapter 与 RoBerta 在三个下游任务上微调的效果对比