**实验三 预训练和微调方法**

**一、实验目标**

本实验旨在通过预训练和微调（fine-tuning）的方法，训练一个Transformer模型，使其能够完成一个与世界知识相关的任务——预测名人的出生地。通过这个实验，学生将能够理解预训练模型如何在没有直接提供任务相关数据的情况下，学习并利用世界知识，以及预训练对模型性能的提升作用。此外，学生还将学习如何通过微调将预训练模型应用于特定任务，并分析模型在预训练和微调过程中的表现，从而深入理解预训练模型在自然语言处理中的重要性和应用方式。

**二、实验要求**

**1.实验环境**

本地开发环境推荐使用Python 3.8及以上版本，安装PyTorch、TensorBoard等必要的库；训练环境需要6GB以上显存，若无相关硬件资源，可以使用Google Colab或Kaggle平台进行模型训练，以充分利用GPU资源。

Kaggle平台使用教程：

https://blog.csdn.net/weixin\_42426841/article/details/143591586

**2.代码实现**

使用提供的wiki.txt文件作为预训练数据，通过实现CharCorruptionDataset类中的getitem()函数，完成对文本的片段掩盖（span corruption）任务的定义。片段掩盖任务是随机选择文本中的一段字符，并将其替换为特殊的标记（sentinel），模型需要根据这些标记还原被替换的字符。

在run.py文件中填写预训练部分的代码，以实现模型在片段掩盖任务上的预训练。预训练过程预计需要大约1小时的时间。

修改run.py文件中微调部分的代码，使其能够处理预训练后的模型。具体来说，如果在命令行参数中提供了预训练模型的路径，则在微调之前加载该预训练模型。

使用birth\_places\_train.tsv文件作为微调数据，对预训练后的模型进行微调，使其能够完成预测名人出生地的任务。

**3.模型评估**

在微调完成后，使用birth\_dev.tsv文件作为开发集，对模型进行评估，并将预测结果写入磁盘。

使用birth\_test\_inputs.tsv文件作为测试集，对模型进行评估，并将预测结果写入磁盘。

预期模型在开发集上的准确率至少达到15%，在测试集上的准确率也应达到类似的水平。

**三、实验步骤**

**1. 环境搭建**

安装Python 3.8及以上版本，并配置好PyTorch环境。克隆实验代码仓库，安装必要的依赖库。

1. **数据准备**

下载并准备wiki.txt文件作为预训练数据。下载并准备birth\_places\_train.tsv、birth\_dev.tsv和birth\_test\_inputs.tsv文件作为微调和评估数据。

1. **预训练实现**

在src/dataset.py中实现CharCorruptionDataset类的getitem()函数，完成片段掩盖任务的定义。在run.py中填写预训练部分的代码，实现模型在片段掩盖任务上的预训练。运行预训练命令，开始预训练过程，并记录预训练的时间和资源消耗。

1. **微调实现**

修改run.py中微调部分的代码，使其能够加载预训练模型。使用birth\_places\_train.tsv文件作为微调数据，对预训练后的模型进行微调。运行微调命令，开始微调过程，并记录微调的时间和资源消耗。

1. **模型评估**

使用birth\_dev.tsv文件作为开发集，对微调后的模型进行评估，并将预测结果写入磁盘。使用birth\_test\_inputs.tsv文件作为测试集，对微调后的模型进行评估，并将预测结果写入磁盘。分析模型在开发集和测试集上的表现，记录准确率、预测结果示例等信息。

**四、实验报告**

提交实验报告，包括预训练和微调过程中的代码实现、模型训练的时间和资源消耗、模型在开发集和测试集上的表现（如准确率、预测结果示例等），以及对实验结果的分析和总结。在报告中分析预训练对模型性能的影响，比较预训练模型和未预训练模型在任务上的表现差异，并探讨预训练模型在自然语言处理任务中的优势和局限性。

**五、实验截止时间及提交方式**

实验截至时间为4月28日晚24：00，请同学们将整个项目打包，和实验报告一起，[发送到splab\_nlp\_2025@163.com](mailto:%E5%8F%91%E9%80%81%E5%88%B0766051282@qq.com), 文件命名格式为“姓名-学号-预训练和微调方法.zip”







