NLP Transformer研修班 项目2

项目描述与目标

我们在上一个项目中系统学习了基于Transformer模型的自然语言理解模块和自然语言生成模块。并且我们也实现了基于检索的对话生成模块。 但是，虽然目前检索式NLG模型在对话应用中已经展现出了巨大的价值，其依然存在很多不足。 检索式NLG模型的最大不足在于其所生成的回复只能来自于其预制的问答库，该模型本身无法生成新的答案。 在本项目中，我们会带领大家实现基于Transformer模型的生成式NLG模型。 虽然这种类型的NLG模型存在一定的不可控性，但是随着预训练模型技术的发展，我们已经可以得到非常令人振奋的生成效果。 甚至在某些特定领域，基于预训练Transformer模型所构建的文本生成已经开始创造巨大的商业价值（如文本摘要，代码生成，心理咨询等）。 在本项目中，我们使用对话预料，带领大家从0构建一个自己的对话生成模型。 本项目中介绍的文本生成模型技术适用于几乎所有的文本生成任务，包括摘要生成，机器翻译，代码生成等。 这些应用最大的不同（几乎是唯一的不同）是所使用的训练数据不同。完成本项目的学员完全有能力在特定的数据集上独立开发和优化上述文本生成模型。

项目预期结果

本项目主要目的是实现基于Transformer模型的文本生成模块。 通过本项目的练习，你将掌握如下技能：

1. 熟练掌握Transformer模型的结构，包括：

- 多头自注意机制的实现
- Transformer模型中各个子模块的组织形式

2. 在原有Transformer结构上做修改，实现考虑多输入的Transformer模型。

项目数据描述

LCCC：包含超过一千万个session的闲聊对话。

项目使用工具

1. Python
2. pytorch

项目预期结果

1. 实现基于预训练Transformer的生成式NLG模型。
2. 了解文本生成模型的常用技术，包括：

- 自回归语言模型中的MLE loss
- 文本生成模型中的各种解码技术
- 使用数据并行的方式利用多张GPU训练模型

3. 有能力独立开发和优化基于预训练模型的文本生成模块。

项目的整个框架

- 数据目录：这一部分包含用于训练对话模型的数据，我们提供了一些数据的样本，详见`data/toy_data.txt`文件。但是完整的LCCC数据集需要自行下载和处理，数据下载地址为：`https://github.com/thu-coai/CDial-GPT`。
- 模型目录：这一部分包含模型的定义以及用于定义数据集和损失函数计算方法的脚本。具体代码详见`model`目录。
- 训练脚本：这一部分包含训练入口脚本以及用来做inference的基本，具体代码在本项目的根目录中。

项目文件说明

本项目的根目录中主要包含如下文件：

- `train.py`：训练脚本的入口文件，负责解析输入参数，调用模型训练过程
- `infer_beam.py`：使用beam search的方法，在已经训练好的模型上，预测相应输入所对应的对话回复。
- `infer.py`：使用greedy decoding的方法，在已经训练好的模型上，预测相应输入所对应的对话回复。
- `interact.py`：可以使用本脚本与已经训练好的模型进行动态交互。
- `config.json`：训练模型时的示例配置文件。
- `run_infer.sh`：使用多张GPU，做批量化的模型推理。
- `run_training.sh`: 启动分布式训练的脚本。

本项目的模型目录中包含如下文件：

- `dataset.py`：数据读取部分的实现文件，定义了如何将训练数据转换为Pytorch的输入格式。
- `loss.py`：实现Label Smoothing loss。
- `model_multi_input.py`：实现了生成模型的推理算法，以及辅助计算logits的工具函数。
- `optim.py`：实现了Adam优化器。
- `text.py`：实现了一个自己的tokenizer，基于字级别的tokenizer。
- `trainer_multi_input.py`：实现了Trainer类，定义了训练细节。
- `transformer_module.py`：实现了transformer模型的某个变种：考虑多个输入的transformer模型（参考https://aclanthology.org/P19-1608.pdf）。
- `utils.py`：模型训练过程中所需使用到的各类辅助函数

代码运行环境

代码环境最低运行要求：

- Python >= 3.6
- pytorch >= 1.11.0

本项目代码在Python3.9, pytorch==1.11.0环境中测试通过。

请在开始本项目前安装依赖项：

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

并下载预训练好的中文GPT模型，模型的下载地址为：`https://pan.baidu.com/s/1nTdAqsCYC_9BO5RNB89p0Q?pwd=y4af`。将下载好的模型放在

chinese_gpt_original目录中。

TODO List

为了完成以下任务，我们需要逐步熟悉、掌握Pytorch框架，所以请大家在完成每个模块时先查阅一下Pytorch的文档， 弄清楚要实现的模块是做什么的以及如何使用。

- **任务1**：实现读取数据的脚本，补全model/dataset.py文件中的DialogDataset类。
- **任务2**：实现Label Smoothing方法，补全model/loss.py文件中的LabelSmoothingLoss类。
- **任务3**：实现计算多头注意力的代码，补全model/transformer_module.py文件中的MultiheadAttention.foward方法。
- **任务4**：补全训练脚本，即model/trainer_multi_input.py文件中的Trainer._eval_train方法。

至此，你应该已经可以通过指定tain.py脚本训练出一个基于GPT模型的中文对话模型了，在开始训练前，你应该先下载预训练好的中文GPT模型（使用本次项目中所提供的模型）。 然后将下载好的模型放在某个目录中，并修改config.json文件中的vocab_path和cgpt_parameters_dir参数项，使其指向该目录。

建议在开始训练前将config.json文件放置在一个独立的项目文件夹中。然后使用如下命令训练模型

```
python trian.py \
    --config {where is config} \
    --gpu {which GPU to use} \
```

或者是使用如下命令在多个GPU上训练模型：

```
bash run_training.sh # remember to modify the path to the config file in this
script
```

注意：本项目所使用的数据需要自定下载和处理，并且模型的训练时间较长，建议使用多张GPU。