Tutorial 1: SFT Practice

0. Introduction

This tutorial will guide you through the process of conducting a **Supervised Fine-Tuning (SFT)** experiment using the **Qwen series <1B model**. The aim here is to understand the "**data-training-evaluation**" cycle, an essential process for model fine-tuning. Remember, the goal is not to achieve the highest performance possible but to explore and gain a deeper understanding of the workflow.

1. Dataset, Model and Preparation

1.1 Model:

We used the 0.5B qwen-based model as the training model for our assignment. The qwen series is a commonly used open-source model in academia and industry. You can download from HuggingFace.

`Qwen/Qwen2.5-0.5B`

1.2 Dataset:

`Open-Orca/OpenOrca`

2. Preparing data cleaning for SFT

2.1 Intro

Before we can start training, it is important to clean our data. The quality of training data directly influences the effectiveness of our fine-tuning process. We aim to address four primary data quality issues:

1. **Formatting issues:** non-standard roles, missing fields, mismatched dialogue turns
2. **Quality issues:** empty/garbage text, repetition, templated spam, extremely short/extremely long anomalous samples
3. **Consistency issues:** multiple repetitions of the same semantic meaning, conflicting answers to the same prompt
4. ... (You can explore more here to find out more advanced data cleaning tricks) e.g. data distribution, information gains...

3. SFT Train

3.1 Intro

The goal of SFT is to leverage the general language knowledge learned by the base model while tailoring it to specific use cases or tasks. In this experiment, our objective is not only to fine-tune the model but also to understand the entire closed-loop process of data preparation, training, and evaluation in a practical setting. We will pay special attention to the cleaning and preprocessing of the dataset to ensure that high-quality data is fed into the model. Additionally, we will optimize the fine-tuning process by selecting the right hyperparameters and training configuration to ensure the model converges effectively and yields optimal performance.

3.2 Framework and suggested settings

For this experiment, we will be using **llama-factory** as the primary framework for training and fine-tuning the model. `llama-factory` is an efficient framework designed for large-scale model training and fine-tuning, providing support for distributed training, easy integration with the Hugging Face library, and optimized utilities for managing data pipelines.

Basic Settings:

- Base model: `Qwen/Qwen2.5-0.5B`
- Full finetuning

Key Hyperparameters to Tune for Performance:

- **Learning Rate**: Controls how much to adjust the weights with each training step.
- **Batch Size**: The number of samples used in one update cycle.
- **Epochs**: The number of times the model will see the entire dataset.
- **Warmup Steps**: The number of steps to gradually increase the learning rate.
- **Weight Decay**: A regularization technique to prevent overfitting.
- **Max Sequence Length**: The maximum length of the input sequences.
- **Optimizer**: The algorithm used for updating model parameters.
- ...

4. Evaluation

4.1 Intro

Once the fine-tuning is complete, we need to evaluate the model's performance. Evaluation should check:

1. **Data Distribution**: Has the model learned the distribution of the data? This can be verified by observing improvements in loss and perplexity.
2. **Output Behavior**: Has the behavior of the model improved, particularly in terms of generating more accurate or aligned outputs? We will use both manual and automatic metrics to assess this.

4.2 Evaluation details

The final report should include the scores for the following 8 evaluation sets. Please use standard evaluation libraries to report the scores whenever possible.

```
mmlu, arc_easy, arc_challenge, hellaswag, winogrande, truthfulqa_mc2, piqa, boolq
```

(Please do not worry too much about how high or low the actual scores are; the metric scores are not the only indicator for this assignment, so there's no need to be overly concerned about this part.)

4.3 Example

In our example, we provide two evaluation examples:
1. lm_eval for mmlu_test;
2. lm_eval for ARC-E.

Note that `llamafactory-cli eval` feature will be deprecated in later versions of llamafactory. You can try it, but it's not recommended.

4.3.1 Setup and preparation

```
!pip -q install -U "lm_eval[hf]" tiktoken sentencepiece
```

```
import os, json, time
import torch

# TODO: Change it to the output directory after your SFT training is complete.
SFT_MODEL_DIR = "/content/your_sft_model_dir"
OUT_DIR = "/content/lm_eval_results/your_output_subdir"
os.makedirs(OUT_DIR, exist_ok=True)

MODEL_ARGS = f"pretrained={SFT_MODEL_DIR},trust_remote_code=True,dtype=float16"
DEVICE = "cuda:0"
BATCH_SIZE = 4
```

4.3.2 MMLU (5-shot for default)

```
!python -m lm_eval \
  --model hf \
  --model_args "{MODEL_ARGS}" \
  --tasks mmlu \
  --num_fewshot 5 \
  --device {DEVICE} \
  --batch_size {BATCH_SIZE} \
  --apply_chat_template \
  --output_path "{OUT_DIR}/mmlu_5shot.json"
```

4.3.3 ARC-E（arc_easy，0-shot for default）

```
# Cell — ARC-E (0-shot)
!python -m lm_eval \
  --model hf \
  --model_args "{MODEL_ARGS}" \
  --tasks arc_easy \
  --num_fewshot 0 \
  --device {DEVICE} \
  --batch_size {BATCH_SIZE} \
  --apply_chat_template \
  --output_path "{OUT_DIR}/arc_easy_0shot.json"
```

4.4 Detailed Requirement

For reproducibility, there are many key parameters involved in generating or scoring the results during testing. Please ensure these are aligned.

4.4.1 mmlu (MMLU, Knowledge + Reasoning, Multidisciplinary)

> fewshot: 5
> set apply_chat_template True

4.4.2 arc_easy (ARC Easy, scientific common sense reasoning)

> fewshot: 0
> set apply_chat_template True

4.4.3 arc_challenge (ARC Challenge, difficult version)

> fewshot: 25
> set apply_chat_template True

4.4.4 hellaswag (HellaSwag, Common sense completion)

> fewshot: 10
> set apply_chat_template True

4.4.5 winogrande (WinoGrande, Reference resolution/Common sense)

> fewshot: 5
> set apply_chat_template True

4.4.6 truthfulqa_mc2 (TruthfulQA, Anti-illusion/Fact-checking reliability, MC version)

> fewshot: 0
> set apply_chat_template True

4.4.7 piqa (PIQA, Physics General Knowledge Quiz)

> fewshot: 0
> set apply_chat_template True

4.4.8 boolq (BoolQ, Reading Comprehension / True or False Questions)

> fewshot: 0
> set apply_chat_template True

EOF

Please feel free to contact TAs if you have any questions. Thx !!!