CS60216: Assignment 1

Implementing RLHF and DPO

[**Deadline : 3rd March, 2025**]

IMPORTANT INSTRUCTIONS

- **Plagiarism:** We will be employing strict plagiarism checking. If your code matches with another student's code, all those students whose codes match will be ***awarded zero marks*** without any evaluation. Therefore, it is your responsibility to ensure you neither copy anyone's code nor anyone is able to copy your code.
- **Code error:** If your code doesn't run or gives an error while running, marks will be awarded based on the correctness of logic. If required, you might be called to meet the TAs and explain your code.
- **Python library restrictions:** You can use simple python libraries like numpy, os, sys, collections, timeit, etc. You can also use Trainer class to help you with setting up the training loop. However, YOU CANNOT USE ANY SPECIALIZED TRAINING LIBRARIES like RewardTrainer or DPOTrainer. If your code is found to use any such library, you will be awarded zero marks for this assignment without any evaluation.

SUBMISSION INSTRUCTIONS

Submit the following files :

Assignment1_<ROLL_NO>_code.ipynb
Assignment1_<ROLL_NO>_reward_model.pt
Assignment1_<ROLL_NO>_rlhf_trained.pt
Assignment1_<ROLL_NO>_dpo_trained.pt
Assignment1_<ROLL_NO>_report.pdf
README.txt

In a zipped file named : **Assignment2_<ROLL_NO>.zip (or tar.gz)**

Your README.txt file should contain the following information -

1. **[Mandatory]** Mention your **Roll Number** on the first line of your README.
2. **[Mandatory]** Any specific library requirements to run your code and the specific Python version you are using.
3. **[Mandatory]** Provide details of your approach with implementing both of the methods.
4. **[Optional]** Any other special information about your code or logic that you wish to convey.

IMPORTANT: PLEASE FOLLOW THE EXACT NAMING CONVENTION OF THE FILES AND THE SPECIFIC INSTRUCTIONS IN THE TASKS CAREFULLY. ANY DEVIATION WILL RESULT IN DEDUCTION OF MARKS. PLEASE NOTE THE EVALUATION GUIDELINES ON PAGE 5.

This assignment is on implementing Reinforcement Learning with Human Feedback (RLHF) and Direct Preference Optimization (DPO) to generate non-harmful answers when prompted with harmful and stereotypic questions.
You have to use the Python programming language for this assignment.
The total marks for this assignment is 50.

Dataset: For this assignment, you will be using the *CulturalKaleidoscope_Preference* Dataset [1]. The relevant files will have to be downloaded from the given link -
Safe-GenAI_Assignment_Data

[1] Banerjee, S., Layek, S., Shrawgi, H., Mandal, R., Halder, A., Kumar, S., Basu, S., Agrawal, P., Hazra, R. and Mukherjee, A., 2024. Navigating the Cultural Kaleidoscope: A Hitchhiker's Guide to Sensitivity in Large Language Models. NAACL 2025

Task A (Implementing RLHF)

Train a reward model to predict human-like preference scores.
Steps:

1. Fine-tune a small transformer-based model, **bert-base-uncased** (https://huggingface.co/google-bert/bert-base-uncased), to predict the preference scores.
2. Train on the preference dataset using the Negative Log-Likelihood loss based on the Bradley-Terry model described below:

$$loss(\theta) = -E_{(x,y_w,y_l)\sim D}\left[log\left(\sigma\left(r_\theta\left(x,y_i\right) - r_\theta\left(x,y_j\right)\right)\right)\right]$$

3. Save the trained reward model for later use.

Use Proximal Policy Optimization (PPO) to fine-tune a language model using the reward model.
Steps:

1. Load **gpt2-medium** (https://huggingface.co/openai-community/gpt2-medium) as the base language model.
2. Define a reinforcement learning loop where the model generates responses and the reward model assigns scores.
3. Use PPO to optimize the language model based on the reward scores and a reference model. Use the frozen parameter version of the **gpt2-medium** (https://huggingface.co/openai-community/gpt2-medium) (i.e the frozen base model) as the reference model. **You can use PPOTrainer from trl library**.
4. Log the improvement in generated responses over training epochs.

Task B (Implementing DPO)

Use Direct Preference Optimization (DPO) to fine-tune the language model on the preference dataset.
Steps:

1. Load **gpt2-medium** (https://huggingface.co/openai-community/gpt2-medium) as the base language model.
2. Implement DPO loss function and optimize the model accordingly. **Do not use DPOTrainer in this case.**

$$loss(\pi_\theta; \pi_{ref}) = -E_{(x,y_w,y_l)\sim D}\left[log\ \sigma\left(\beta\ log\frac{\pi_\theta(y_w|x)}{\pi_{ref}(y_w|x)} - \beta\ log\frac{\pi_\theta(y_l|x)}{\pi_{ref}(y_l|x)}\right)\right]$$

3. Use the frozen version of **gpt2-medium** (https://huggingface.co/openai-community/gpt2-medium) as the reference model.
4. Log the improvement in generated responses over training epochs.

Task C (Performance Comparison)

Compare the performance of the 2 methods : RLHF and DPO
Compare and contrast RLHF and DPO based on sample efficiency, response quality and computation cost.

Steps:

1. Generate responses for the harmful or stereotype-triggering questions in the test split using the RLHF-trained and the DPO-trained models. **Both of the models will be fine-tuned versions of gpt2-medium**.
2. Evaluate them quantitatively (using automated metrics – **BLEU** and **ROUGE**). Use the **"more_preferred"** column from the dataset as the reference answer. **You can use library functions for calculating BLEU and ROUGE scores.**
3. Write a short report summarizing your findings, and do a contrast study of the 2 methods (RLHF and DPO) based on sample efficiency, response quality and computation cost.

EVALUATION GUIDELINES

1. Task A **[20 marks]**
    a. Preprocess the dataset for training the reward model : **4 marks**
    b. Correctly training the reward model : **8 marks**
    c. Correctly implementing RLHF using PPO : **8 marks**
2. Task B **[10 marks]**
    a. Correctly defining DPO loss : **5 marks**
    b. Correctly implementing DPO : **5 marks**
3. Task C **[15 marks]**
    a. Evaluation using BLEU : **5 marks**
    b. Evaluation using ROUGE : **5 marks**
    c. Final report : **5 marks**
4. README **[5 marks]**
5. Deductions :
    a. Plagiarism : **-50 marks**
    b. Using libraries that are not allowed : **-50 marks**
    c. Not following naming conventions : **-2 marks for every violation**
    d. Small bugs in code, etc. that are beyond the overall logic of the code workflow, e.g., not following the input format specification for running code or anything else that does not fall into the marking scheme above: **-2 marks for every violation**
