Course Project

Code of Honor. All external resources used in the project, including research papers, open-source repositories, datasets, and any content or code generated using AI tools, e.g., ChatGPT, GitHub Copilot, Claude, Gemini, must be *clearly cited* in the final submission. The final report must also include *a clear breakdown of individual group member contributions.* Any lack of transparency in the use of external resources or in reporting group contributions will be considered academic dishonesty and will significantly impact the final evaluation.

Topic	Using RL to Finetune a Language Model
Category	Applications of RL

Objective Design and implement a simplified pipeline for RL-based finetuning of language models (LMs). Starting from a small pretrained LM, e.g., DistilGPT2, you will implement a policy optimization method (PPO) to improve LM outputs according to a reward model, e.g., sentiment classifier. The project aims to compare supervised finetuning against RL-based one, and to analyze the effect of different reward models and modifications.

Motivation Reinforcement Learning with Human Feedback (RLHF) is central to the training of modern large language models such as ChatGPT [2]. While full-scale RLHF requires massive compute and human-labeled datasets, a simplified version can be implemented with smaller models and proxy reward functions. This project provides hands-on experience with how PPO [1] can be used to steer a language model's output, giving students practical exposure to one of the most impactful applications of RL in natural language processing.

Requirements The final submission should address the following requirements while the details can be freely decided by the group members.

1. Implementation: in this respect, you should
   - load a small pretrained LM, e.g., DistilGPT2 or GPT2-small,
   - implement PPO to update the LM based on rewards from a reward model, and
   - compare RL-based finetuning with supervised finetuning (baseline).
2. Reward modeling: for RLHF you need to use a proxy reward function such as:
   - a pretrained sentiment classifier, e.g., DistilBERT sentiment model, or
   - lexical rules, e.g., penalize repetition, reward fluency.

   To ensure originality, the groups must modify the basic reward design in some ways, e.g.:
   - add custom lexical constraints,
   - combine multiple reward signals, or
   - train or use a reward model on a different sentiment dataset.

3. Evaluation: the final project should report key evaluation of the implemented algorithms in the modified environment. In this respect, the results should
   - evaluate reward improvements (average reward per generated text),
   - compare outputs qualitatively (examples before vs after RL finetuning), and
   - measure diversity and fluency of outputs.
4. The results should be elaborated through
   - ablation experiments, e.g., without entropy regularization, with different reward functions, and
   - providing discussions on trade-offs between supervised vs RL-based finetuning.

Milestones The following milestones are to be accomplished through semester.

1. Literature Review and Setup
   - Review RLHF and PPO in the context of LM training.
   - Select a small pretrained LM and finalize reward model choice.
2. Implementation
   - Implement supervised finetuning baseline.
   - Implement PPO-based RL finetuning.
   - Train models on short prompts and validate learning.
3. Evaluation and Analysis
   - Collect and plot reward improvements over training.
   - Compare supervised vs RL finetuned models.
   - Perform ablation experiments.
4. Final Report and Presentation

Submission Guidelines The main body of work is submitted through Git. In addition, each group submits a final paper and gives a presentation. In this respect, please follow these steps.

- Each group must maintain a Git repository, e.g., GitHub or GitLab, for the project. By the time of final submission, the repository should have
  - Well-documented codebase
  - Clear `README.md` with setup and usage instructions
  - A `requirements.txt` file listing all required packages or an `environment.yaml` file with a reproducible environment setup
  - Demo script or notebook showing sample input-output
  - *If applicable,* a `/doc` folder with extended documentation
- A final report (maximum *5 pages*) must be submitted in a PDF format. The report should be written in the provided formal style, including an abstract, introduction, method, experiments, results, and conclusion.
  Important: Submissions that do not use template are considered *incomplete.*
- A 5-minute presentation (maximum *5 slides including the title slide*) is given on the internal seminar on Week 14, i.e., *Dec 1 to Dec 5,* by the group. For presentation, any template can be used.

Final Notes While planning for the milestones please consider the following points.

1. You are encouraged to explore innovative approaches as long as the core objectives are met.
2. PPO should be implemented efficiently with small batch sizes to keep training feasible.
3. Full RLHF pipelines are extremely compute-intensive. In this project, feasibility is ensured by restricting to small models (e.g., DistilGPT2), short sequences, and limited training steps.
4. Computing resources must be managed carefully. The groups should plan shorter experiments (e.g., a few thousand update steps).
5. Vocabulary size, sequence length, and batch size should be reduced to keep memory use reasonable.
6. The goal is **not** to train a high-performing LM, but to *demonstrate how PPO can shape LM outputs* under a reward signal.
7. Teams are expected to manage their computing needs and are advised to perform early tests to estimate runtime and training feasibility. As graduate students, team members can use facilities provided by the university, e.g., ECE Facility. Teams are expected to inform themselves about the limitations of the available computing resources and design accordingly.

References

[1] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[2] Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.