Natural Language Processing and Large Language Models

Corso di Laurea Magistrale in Ingegneria Informatica

Lesson 21

Reinforcement Learning from Human Feedback

Nicola Capuano and Antonio Greco

DIEM – University of Salerno

Outline

- Reinforcement Learning from Human Feedback (RLHF)
- Transformers trl library
- Try it yourself

Reinforcement Learning from Human Feedback (RLHF)

Reinforcement Learning from Human Feedback (RLHF)

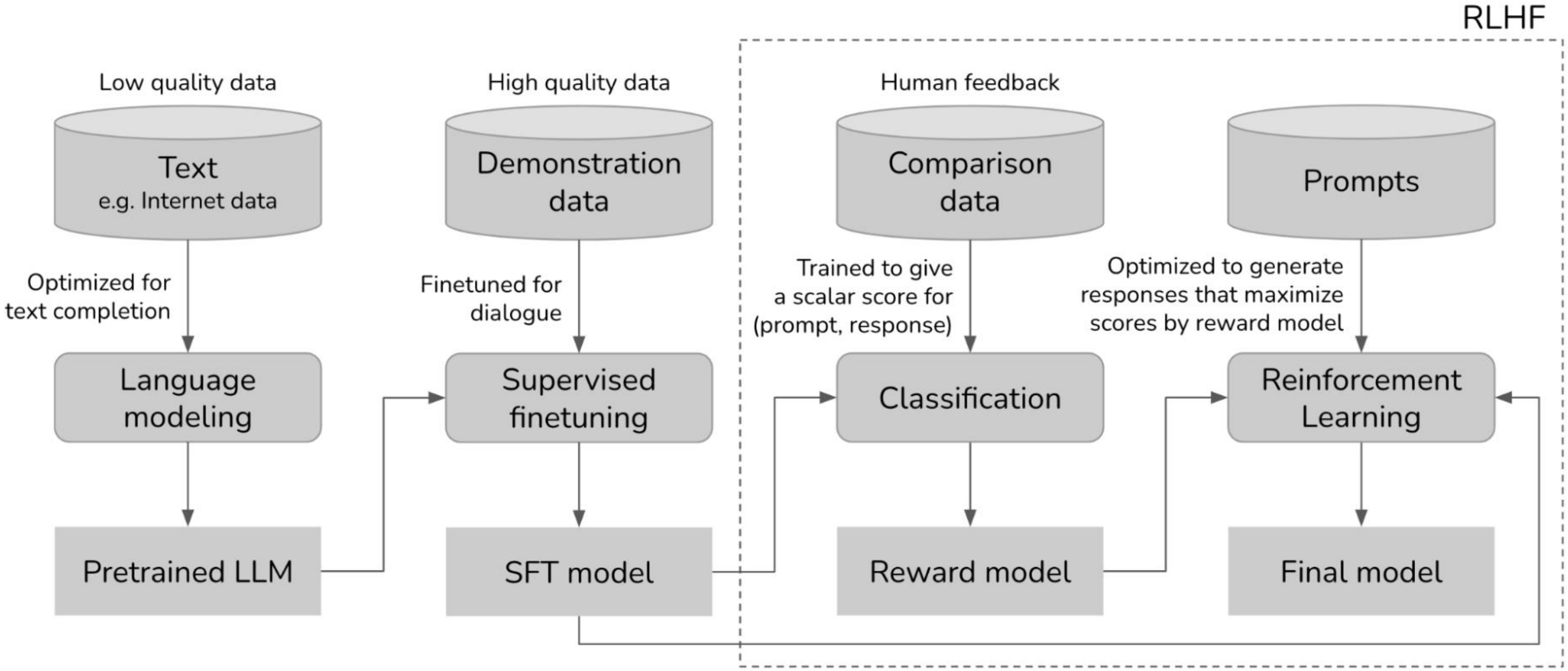
What is RLHF?

- A technique to improve large language models (LLMs) using human feedback as guidance.
- A strategy to balance model performance with alignment to human values and preferences.

Why RLHF?

- It may be a possible strategy to ground the focus of the LLM
- It can enhance safety, ethical responses, and user satisfaction.

Workflow of RLHF



RLHF

Low quality data

Text
e.g. Internet data

Optimized for text completion

Language modeling

Pretrained LLM

High quality data

Demonstration data

Finetuned for dialogue

Supervised finetuning

SFT model

Human feedback

Comparison data

Trained to give a scalar score for (prompt, response)

Classification

Reward model

Prompts

Optimized to generate responses that maximize scores by reward model

Reinforcement Learning

Final model

Scale May '23	**>1 trillion** tokens	**10K - 100K** (prompt, response)	**100K - 1M** comparisons (prompt, winning_response, losing_response)	**10K - 100K** prompts
Examples **Bolded**: open sourced	GPT-x, Gopher, **LLaMa**, **Pythia**, **Bloom**, **StableLM**	**Dolly-v2**		InstructGPT, ChatGPT, Claude, **StableVicuna**

Key components of RLHF

Pre-trained Language Model

- A base LLM trained on large corpora (e.g., BERT, GPT, T5).

Reward Model

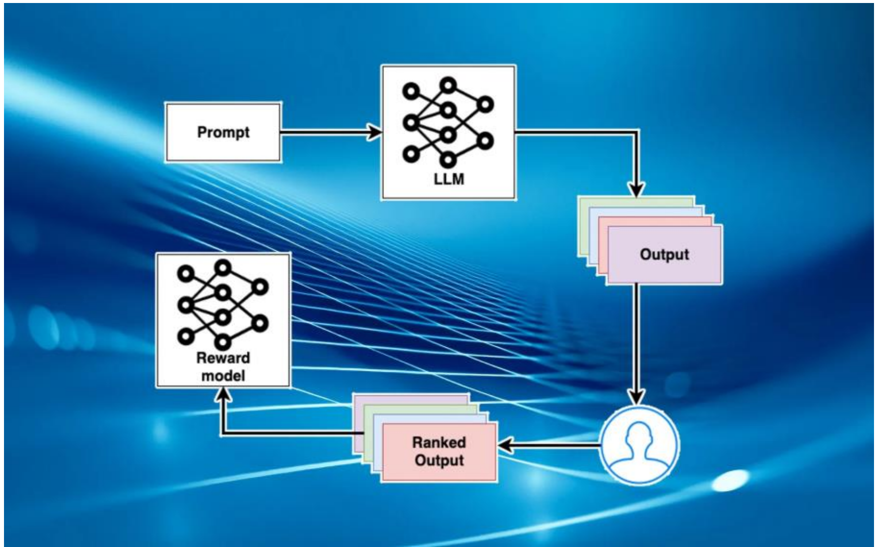
- A secondary model that scores LLM outputs based on human feedback.

Fine-Tuning with Reinforcement Learning

- Optimization of the LLM using reinforcement learning guided by the reward model.

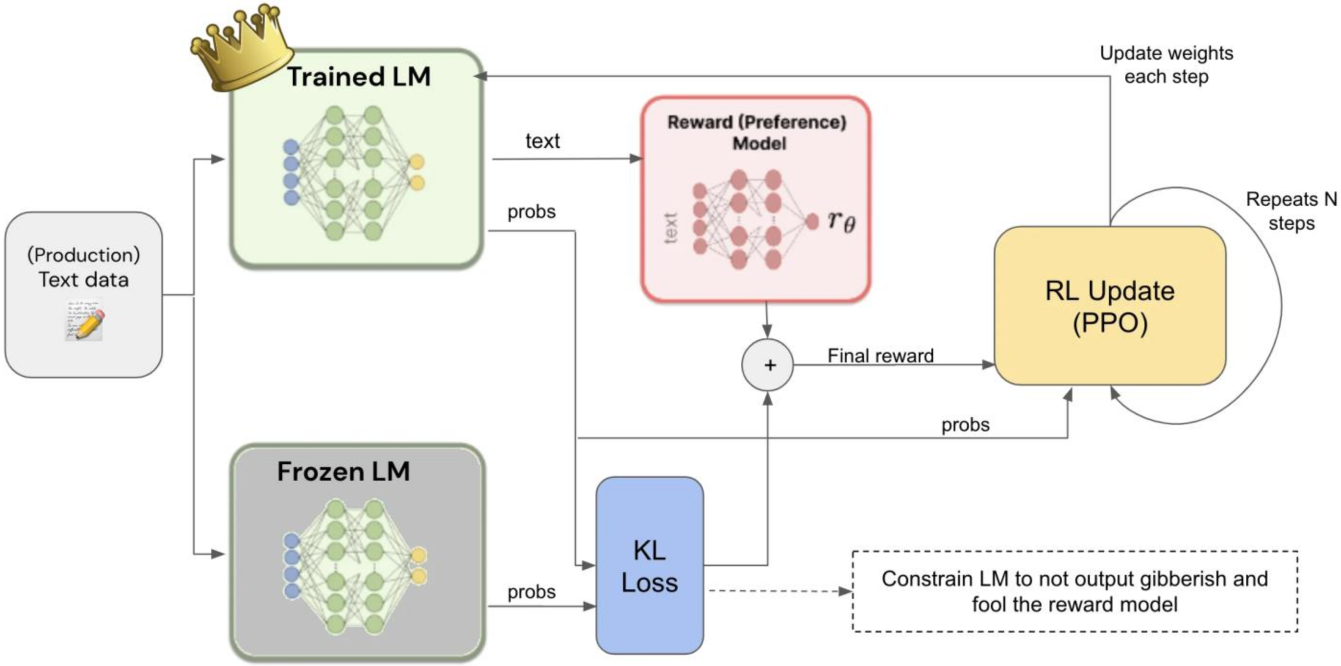
Reward model

- The inputs for training a reward model are:
  - Multiple LLM generated outputs for given prompts
  - Corresponding human rank responses according to their preferences
- The goal is to train a model to predict human preference scores
- The methodology uses a ranking loss function to teach the reward model which outputs humans prefer



Fine tuning with proximal policy optimization (PPO)

- The goal is to align the LLM's outputs with human-defined quality metrics.
  1. Generate responses using the LLM.
  2. Score responses with the reward model.
  3. Update the LLM to maximize reward scores.



Pros and cons of RLHF

Pros

- Iterative Improvement: Possibility to collect human feedback as the model evolves and update the reward model and fine-tune iteratively.
- Improved Alignment: Generates responses closer to human intent.
- Ethical Responses: Reduces harmful or biased outputs.
- User-Centric Behavior: Tailors interactions to user preferences.

Cons

- Subjectivity: Human feedback may vary widely.
- Scalability: Collecting sufficient, high-quality feedback is resource-intensive.
- Reward Model Robustness: Misaligned reward models can lead to suboptimal fine-tuning.

Tasks to enhance with RLHF

- **Text Generation**: RLHF can be used to enhance the quality of text produced by LLMs.
- **Dialogue Systems**: RLHF can be used to enhance the performance of dialogue systems.
- **Language Translation**: RLHF can be used to increase the precision of language translation.
- **Summarization**: RLHF can be used to raise the standard of summaries produced by LLMs.
- **Question Answering**: RLHF can be used to increase the accuracy of question answering.
- **Sentiment Analysis**: RLHF has been used to increase the accuracy of sentiment identification for particular domains or businesses.
- **Computer Programming**: RLHF can be used to speed up and improve software development.

Case study: GPT-3.5 and GPT-4

- The pre-trained models have been fine-tuned using also RLHF.

- OpenAI declares that achieved with RLHF
  - Enhanced alignment
  - Fewer unsafe outputs
  - More human-like interactions.

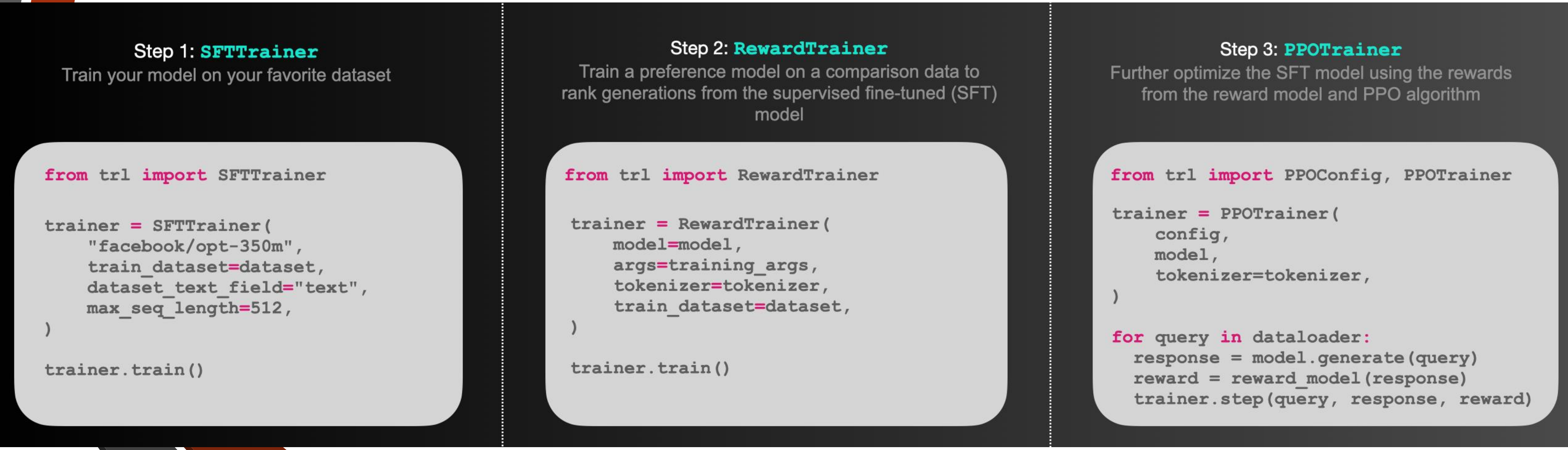
- These models were or are widely used in real-world applications like ChatGPT.

- The models are still incrementally improved with additional human feedback.

Transformers trl library

TRL
Transformer Reinforcement Learning

- **TRL** is a full stack library where we provide a set of tools to train transformer language models with Reinforcement Learning, from the Supervised Fine-tuning step (SFT), Reward Modeling step (RM) to the Proximal Policy Optimization (PPO) step.
- The library is integrated with HuggingFace transformers.



Step 1: `SFTTrainer`

Train your model on your favorite dataset

```
from trl import SFTTrainer

trainer = SFTTrainer(
    "facebook/opt-350m",
    train_dataset=dataset,
    dataset_text_field="text",
    max_seq_length=512,
)

trainer.train()
```

Step 2: `RewardTrainer`

Train a preference model on a comparison data to rank generations from the supervised fine-tuned (SFT) model

```
from trl import RewardTrainer

trainer = RewardTrainer(
    model=model,
    args=training_args,
    tokenizer=tokenizer,
    train_dataset=dataset,
)

trainer.train()
```

Step 3: `PPOTrainer`

Further optimize the SFT model using the rewards from the reward model and PPO algorithm

```
from trl import PPOConfig, PPOTrainer

trainer = PPOTrainer(
    config,
    model,
    tokenizer=tokenizer,
)

for query in dataloader:
  response = model.generate(query)
  reward = reward_model(response)
  trainer.step(query, response, reward)
```

Try it yourself

Try it yourself

- Study the trl library on HuggingFace: https://huggingface.co/docs/trl/v0.7.8/index

- Give a careful look to:
  - PPOTrainer: https://huggingface.co/docs/trl/v0.7.8/ppo_trainer
  - RewardTrainer: https://huggingface.co/docs/trl/v0.7.8/reward_trainer

- Study the examples that are closer to your purposes:
  - Sentiment analysis tuning: https://huggingface.co/docs/trl/v0.7.8/sentiment_tuning
  - Detoxifying a Large Language Model with PPO: https://huggingface.co/docs/trl/v0.7.8/detoxifying_a_lm

- Try to apply RLHF to your project

Natural Language Processing and Large Language Models

Corso di Laurea Magistrale in Ingegneria Informatica

Lesson 21

Reinforcement Learning from Human Feedback

Nicola Capuano and Antonio Greco

DIEM – University of Salerno