Preferences vs Rewards

There are problems where providing *exact scalar rewards*

- is harder than *ranking* potential outputs.

For example

- I may prefer chocolate to vanilla
- but I can't quantify how much more

Technically

- rewards form a total order
    - a reward has a magnitude
    - *all* rewards can be compared and ordered
- preferences form a partial order
    - we can order *some* pairs of outputs
    - without providing a magnitude

      Good > Bad

      Big > Small

      Good > Small ? Small > Good ?

Problems related to aligning the *style* of an LLM's output is a case of preferences.

- multiple answers may be "correct"
- but one answer may be "preferred"

For example

Prompt: "How do I change a tire?"

- **Reply A:** An accurate step-by-step answer.
- **Reply B:** A brief, incomplete answer.

Both replies are "correct" but the first is subjectively better.

An example of *Preference Data* is a triple

$$(x, y^+, y^-)$$

- input x
- the preferred output y^+
- the non-preferred output y^-

The case for preferences

Scenario	Why Preference Data?	Typical Example
RLHF & LLM alignment	Human feedback easier as comparisons	Choosing better LLM output
Hard-to-define or subjective “success”	Preference judgments more reliable	Dialogue, safety, style
Biased or noisy scalar rewards	Preferences less affected by outliers	Creative tasks, open-ended
Interpretability needs	Preferences can include rationales	Transparent value alignment
DPO-style methods	Direct optimization over preferences	Pairwise/choice based loss

In this module, we explore Reinforcement Learning for Preference Data.

Example of a Preference Dataset

Here is a link (https://huggingface.co/datasets/HuggingFaceH4/ultrafeedback_binarized) to the UltraFeedback dataset.

It is used to train an Assistant to be

- Helpful
    - answers the user's prompt; doesn't evade or decline
- Honest
    - gives a truthful answer

The methodology for constructing it is given here (https://huggingface.co/datasets/HuggingFaceH4/ultrafeedback_binarized#dataset-card-for-ultrafeedback-binarized).

- The authors gather a number of prompts across multiple domains.
- An AI assistant is asked to proved multiple responses to a prompt
- A second AI assistant
    - critiques the response based on, e.g., the Helpfulness criteria
    - gives a numerical evaluation
- Which creates a ranking of the responses to a prompt

The "binarized" dataset that we viewed

- selects the highest ranked answer as "Chosen"
- randomly selects the other responses as "Rejected"

Note the use of AI feedback rather than Human Feedback.

In [2]:

```
print("Done")
```

```
Done
```