LLM with RL

By Sameer and Yassin

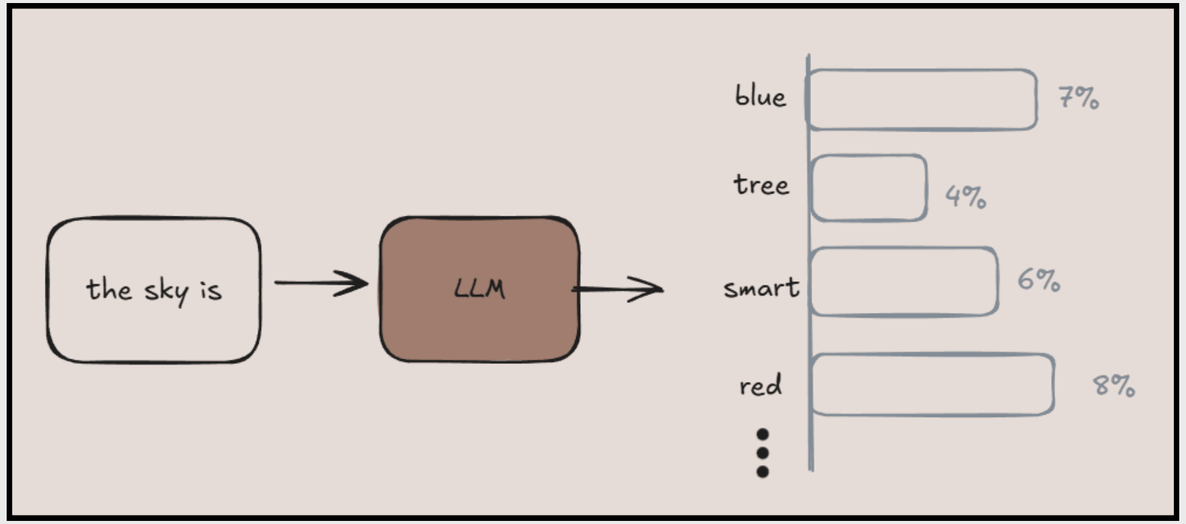
OUTLINE

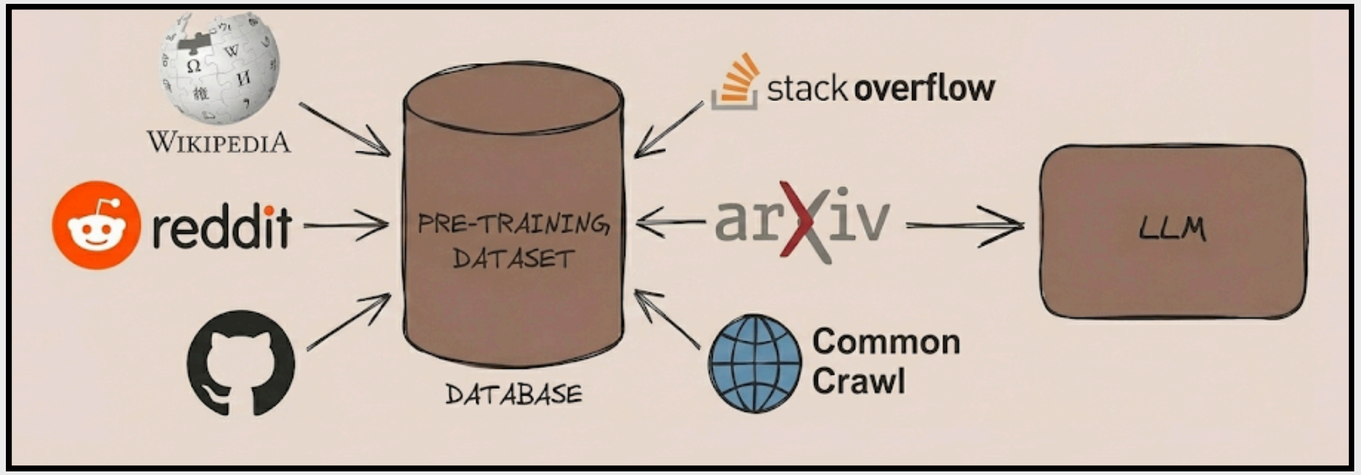
1. LLM training pipline
2. Reinforcement Learning
3. Implementation example

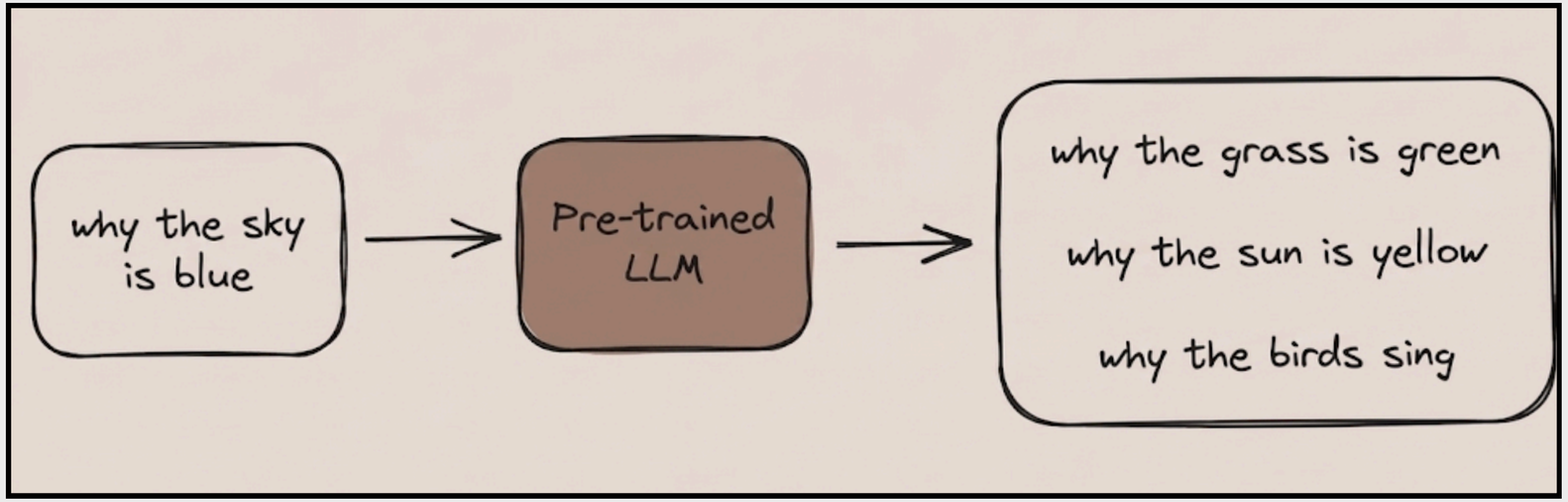
LLM training stages

1) Pre-training

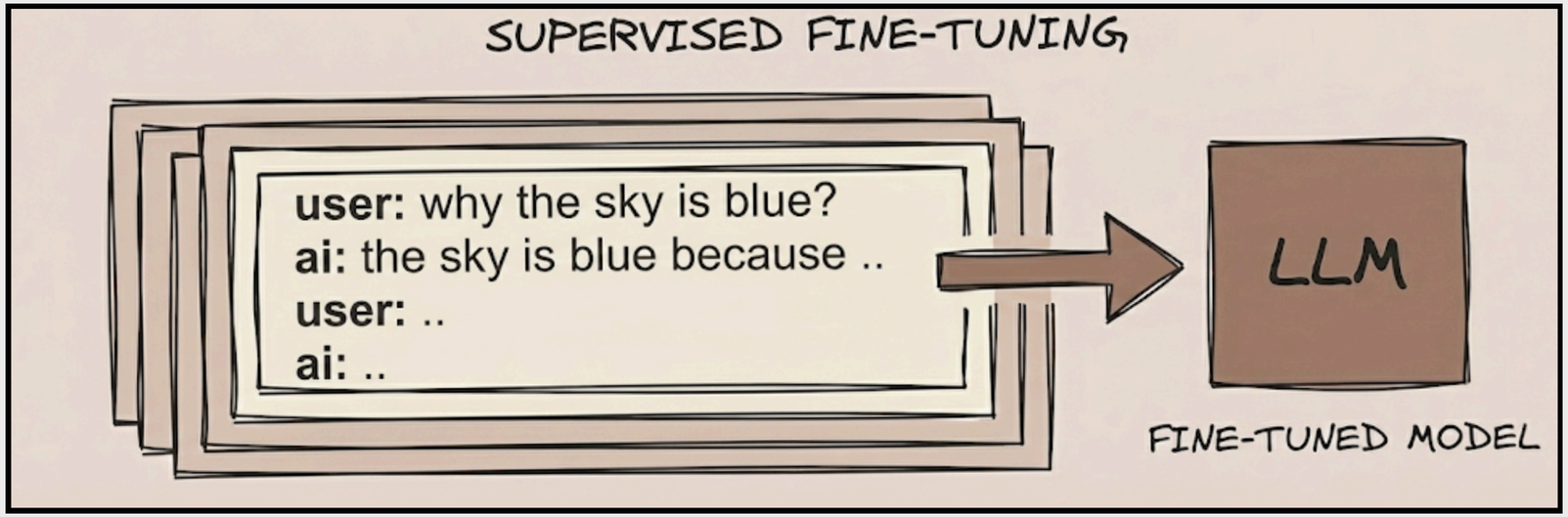
- guessing next token
- Code helps reasoning





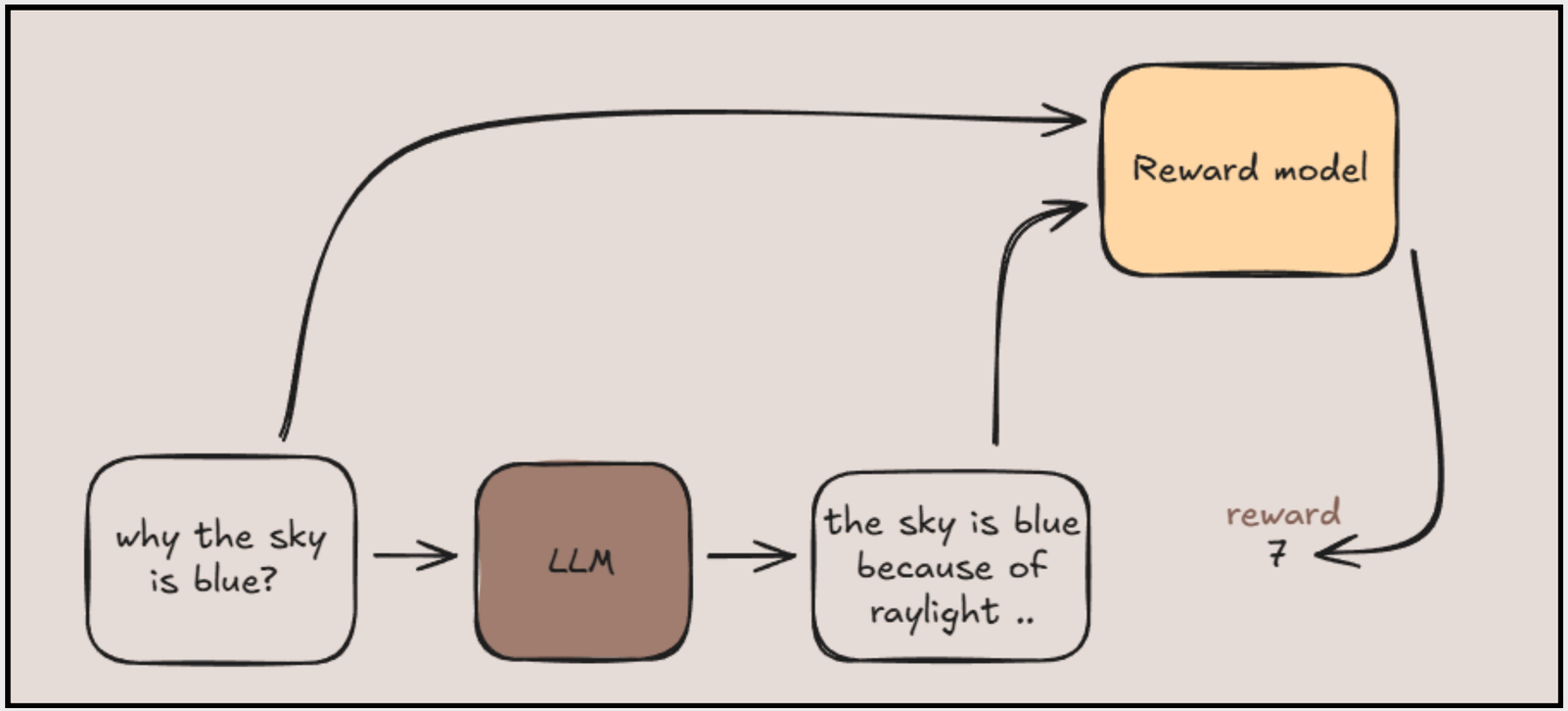


2) Supervised fine-tuning



3) Reinforcement Learning

- previous steps are only imitations

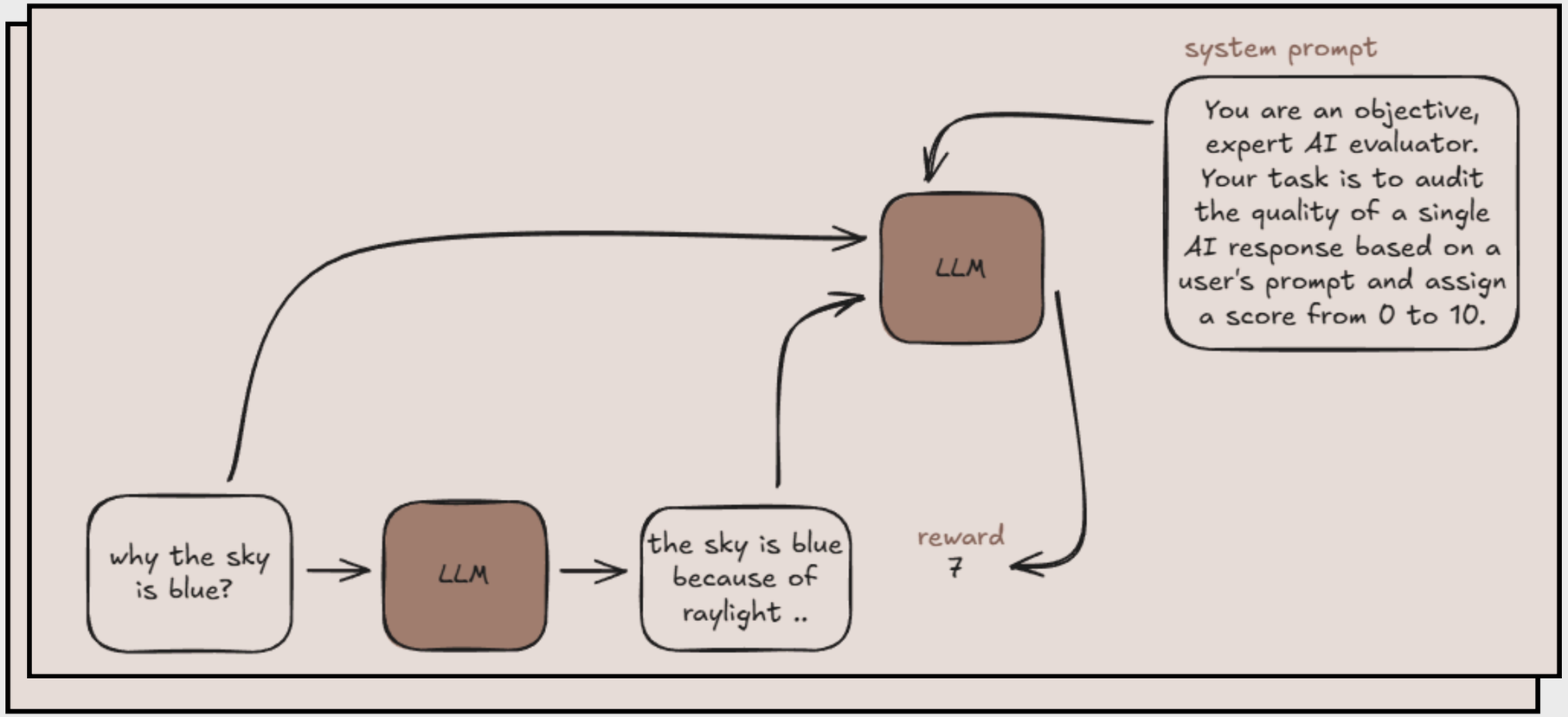


3) Reinforcement learning

- training approaches:
  - RLAIF (RL with AI feedback)
  - RLHF (RL with human feedback)
  - RLVR (RL with verifiable rewards)
- ~~DPO instead of RL~~

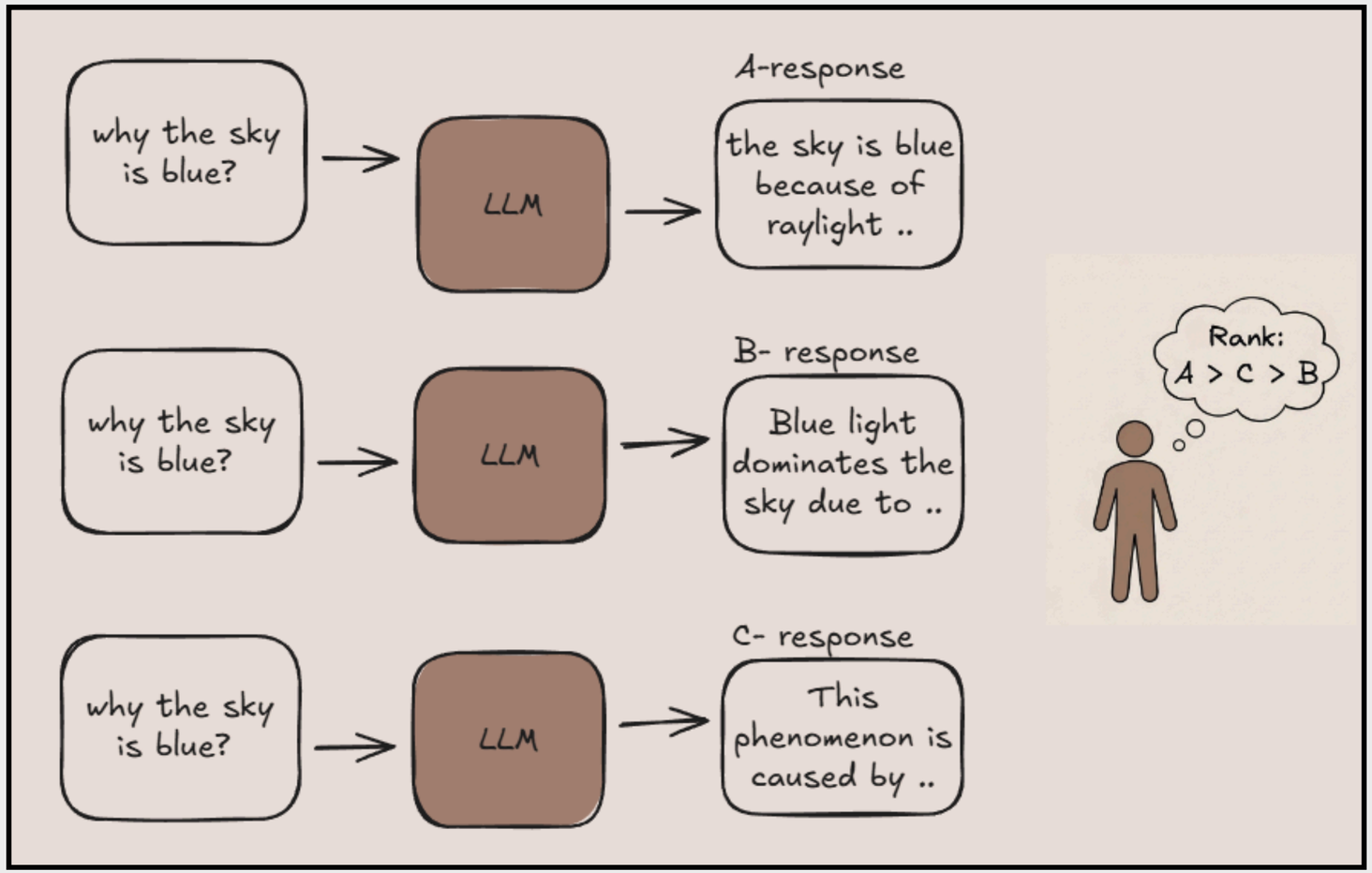
3.1) RLAIF

- learning from itself



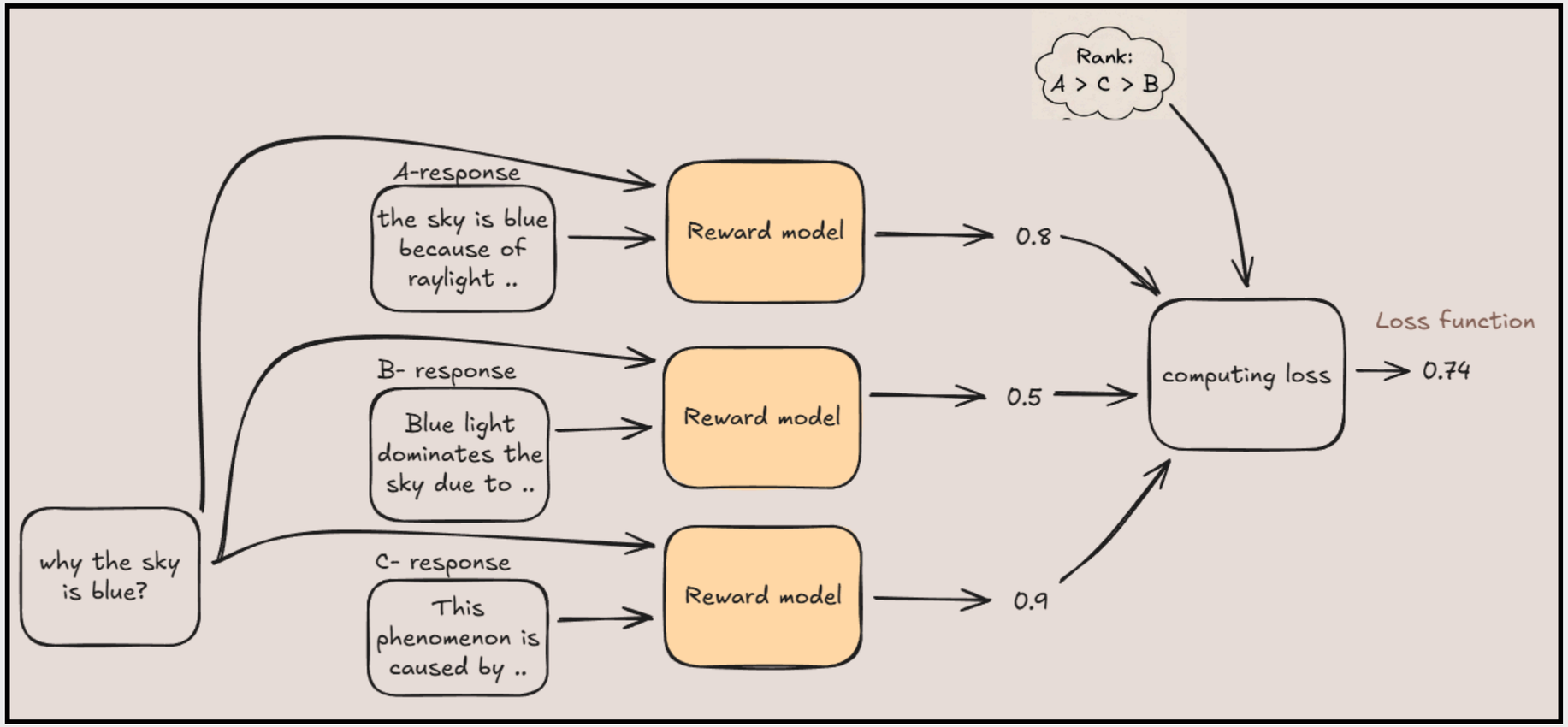
3.2) RLHF

- human ranking responses



3.2) RLHF:

- training the reward model



Project implementation

Thank you

Appendix: Loss for reward model

Let's assume your model currently outputs these raw scores (logits) for the inputs:

- $r_A = 0.8$
- $r_C = 0.5$
- $r_B = 0.9$ *(Note: The model is currently wrong about B, as B should be lowest)*

Here is exactly how the computer calculates the loss for this batch:

1. Form the Pairs

From **A > C > B**, valid pairs (Winner vs. Loser) are:

1. **A vs C** (Winner: A)
2. **C vs B** (Winner: C)
3. **A vs B** (Winner: A)

2. Calculate Score Differences

For every pair, subtract the Loser's score from the Winner's score:

- $\Delta_{AC} = r_A - r_C = 0.8 - 0.5 = \mathbf{0.3}$
- $\Delta_{CB} = r_C - r_B = 0.5 - 0.9 = \mathbf{-0.4}$
- $\Delta_{AB} = r_A - r_B = 0.8 - 0.9 = \mathbf{-0.1}$

3. Apply the Sigmoid Function

The sigmoid function $\sigma(x) = \frac{1}{1+e^{-x}}$ converts the difference into a probability (0 to 1). Th
represents "How confident is the model that the winner is actually better?"

- $P(A > C) = \sigma(0.3) \approx \mathbf{0.57}$ (Model is slightly confident A wins)
- $P(C > B) = \sigma(-0.4) \approx \mathbf{0.40}$ (Model thinks B wins; this is low confidence in C)
- $P(A > B) = \sigma(-0.1) \approx \mathbf{0.47}$ (Model thinks B wins; this is low confidence in A)

4. Compute Log Loss

We want to maximize these probabilities, which means minimizing their negative logarithm.
Formula: $\text{Loss} = -\log(P)$

- $L_{AC} = -\log(0.57) \approx \mathbf{0.56}$
- $L_{CB} = -\log(0.40) \approx \mathbf{0.92}$ (Higher loss because model was wrong)
- $L_{AB} = -\log(0.47) \approx \mathbf{0.75}$

5. Total Loss (Average)

$$L_{\text{total}} = \frac{0.56 + 0.92 + 0.75}{3} \approx \mathbf{0.74}$$