Reward Modeling and Response Evaluation

Estimated Time: 45 minutes

Learning objectives

After completing this lab, you will be able to:

- Explain how system language plays a role in reward modeling
- Analyze and annotate numerical scores based on human preference-driven final answers
- Observe learning model behaviors that contribute to user satisfaction and improved performance
- Verify model performance remains consistent, reliable, and factually accurate
- Recognize how the reward mechanism enforces the Bradley-Terry loss function during learning

Introduction

Reward modeling is an entirely different approach to training language models. In this approach, each generated response is assigned a scalar reward based on human preferences. The reward function, referred to as a preference model, is the primary tool for introducing reinforcement learning in the training process for RLHF (Reinforcement Learning from Human Feedback). The system can improve reliability, coherence, and quality by generating outputs close to human standards. This is the only way to achieve such a level of sophistication.

Key aspects of reward modeling

1. Alignment with human preferences:

Reward models evaluate how well a model's responses align with human preferences.

Example:

Consider a chatbot designed to answer questions about history. If a user asks, **"Who was the first president of the United States?,"** a response of **"George Washington"** would align well with a human preference for factual accuracy and thus receive a high reward.

2. Quantifying response quality:

They assign numerical values to responses, allowing for performance assessment and comparison.

Example:

If two chatbots are compared, and **Chatbot A's** response is more accurate and detailed than **Chatbot B's**, the reward model would assign a higher numerical value to **Chatbot A's** response, indicating its superior quality.

3. Guiding model optimization:

Reward models guide the optimization of model parameters to maximize the assigned score and improve overall performance.

Example:

During training, if the reward model consistently gives higher scores to concise and direct responses, the Large Language Model (LLM) will adjust its parameters to generate more concise and direct answers.

4. Incorporating user preferences:

They incorporate user preferences into the scoring function, enabling customization of model behavior.

Example:

If a user prefers creative and imaginative responses, the reward model can be trained to value such characteristics, thus guiding the LLM(Large Language Model) to generate more innovative content.

5. Ensuring consistency and reliability:

Reward models provide a consistent and reliable evaluation of responses.

Example:

For the same query, **"What is the capital of France?,"** the reward model should consistently give a high score to the response **"Paris"** every time, ensuring reliability in evaluation.

Scenario: Factual accuracy in responses

Chatbots need to get facts right. When they provide wrong information, people lose trust and false information spreads. By breaking text into small pieces called "tokens," we can check how accurate AI responses are and make them better.
Here, we will be discussing an example based on a query related to Antarctica. We will be evaluating 2 responses where one response gives correct information about international agreements, while another makes up a silly story about penguin rulers.

Query: "Which country owns Antarctica?"

- **Tokenized query (Ω):**

ωb1	ωb2	ωb3	ωb4	ωb5
which	*country*	*owns*	*Antarctica*	*?*

Responses:

1. **Chatbot A:** "Antarctica is governed by the Antarctic Treaty System, which includes multiple countries." (Factual and accurate)
2. **Chatbot B:** "Penguin overlords run the showdown there." (Humorous but factually incorrect)

- **Tokenization:**

Response A Tokens ($\hat{\Omega}$ A):

[Antarctica, is, governed, by, the, Antarctic, Treaty, System, which, includes, multiple, countries]

ωb1	ωb2	ωb3	ωb4	ωb5	ωb6	ωb7	ωb8	ωb9	ωb10	ωb11	ωb12
Antarctica	*is*	*governed*	*by*	*the*	*Antarctic*	*Treaty*	*System*	*which*	*includes*	*multiple*	*countries*

Response B Tokens ($\hat{\Omega}$ B):

[our, penguin, overlords, run, the, show, down, there]

ωb1	ωb2	ωb3	ωb4	ωb5	ωb6	ωb7	ωb8
our	*penguin*	*overlords*	*run*	*the*	*show*	*down*	*there*

Scoring function (Reward model)

The scoring function **R** evaluates the quality of a response by assigning a numerical score based on factual accuracy and alignment with human preferences. It processes the tokenized query and response to compute the score.

Mathematical formulation

1. Embedding generation:

The tokenized input **Ω** and response **Ω^** are converted into contextual embeddings using a transformer model (e.g., BERT or GPT). Let **E(Ω)** denote the embedding function:

`E(Ω) = [CLS], e_{ωb1}, e_{ωb2}, ..., e_{ωbn}`

Similarly,

`E(Ω^)` - generates embeddings for the response.

2. Linear layer for reward prediction:

The embeddings are passed through a linear layer to compute the reward score **R**:

$$R(\Omega, \Omega^{\wedge}) = W^{T} \cdot E(\Omega \oplus \Omega^{\wedge}) + b$$

Where:

- **Ω ⊕ Ω^** : Concatenated embeddings of the query and response.
- **W, b**: Learnable weights and bias of the linear layer.

Example scores
The reward model assigns scores based on factual accuracy:

- **Response A (Factual):**
  `R(Ω, Ω^A) = 0.89`
- **Response B (Incorrect):**
  `R(Ω, Ω^B) = 0.03`

Why the scores differ?

- **Response A** contains keywords such as "*Antarctic Treaty System*" and "*multiple countries*", aligning with factual knowledge.
- **Response B** includes nonsensical terms such as "*penguin overlords*", violating factual accuracy.

Reward model loss (Bradley-Terry loss)

To train the reward model, we use the **Bradley-Terry loss** to ensure that the good response (A) receives a higher score than the bad response (B).

The Bradley-Terry loss approach consists of two key components:

1. Loss function
2. Bradley-Terry pair-wise preference loss function

1. Loss function

For a pair of responses **Ω^A** (good) and **Ω^B** (bad), the loss is:

`L(ϕ) = -log σ(R(Ω, Ω^A) - R(Ω, Ω^B))`

Where:

- **σ**: Sigmoid function
  `σ(x) = 1 / (1 + e^(-x))`
- **R(Ω, Ω^A)**: Reward score for the good response.
- **R(Ω, Ω^B)**: Reward score for the bad response.

Interpretation
The term **R(Ω, Ω^A) - R(Ω, Ω^B)** represents the margin between the rewards.

Minimizing **L(ϕ)** ensures that the reward model assigns higher scores to good responses.

2. Bradley-Terry pair-wise preference loss function

In the previous example, we discussed a single training sample (single question) and a single pair-wise response (2 answers). However, in real datasets, we may have multiple training samples and multiple pair-wise responses. So, in such a case, we need to calculate the cumulative loss across all samples.

For multiple pair-wise responses, the equation is applied as follows:

$$\phi^{\wedge} = \arg\min_{\phi} v \left[-\sum^{N}_{n=1} \ln \left(\sigma\left(r(X_n, Y_{n,a} \mid \phi) - r(X_n, Y_{n,b} \mid \phi)\right)\right)\right]$$

This is the loss function for the Bradley-Terry model, often used in preference learning. Let me break down each component:

- **ϕ^**: This represents the optimal set of parameters we're trying to find for our model.
- **argminϕ**: This means we're looking for the value of ϕ that minimizes the following expression.
- **r(Xn, Yn, a|ϕ) and r(Xn, Yn,b|ϕ)**: These are reward or score functions that assign values to options a and b for the input Xn, given parameters ϕ. Higher scores indicate more preferred options.
- **r(Xn, Yn, a|ϕ)−r(Xn, Yn,b|ϕ)**: This calculates the difference in scores between options a and b. A positive value means option A is predicted to be preferred over option b.
- **n=1 ∑ N**: Summing over all N training examples, which are pairs of choices where one was preferred over the other.

3. Training process

3.1. Human feedback:

- Human evaluators rank responses (**A > B**) without assigning exact numerical scores.

3.2. Reward model training:

- The model learns to replicate human preferences by minimizing the Bradley-Terry loss over many such pairs.

3.3. Gradient descent:

Update model parameters ϕ (weights **W**, bias **b**) using:

`ϕ ← ϕ - η∇ϕ L(ϕ)`

Where:

ϕ: Model parameters (weights W, bias b)

η: Learning rate (a hyperparameter controlling step size)

∇ϕ L(ϕ): Gradient of the loss concerning ϕ

Example:
Assume,

- $W = [w1, w2]$, $b = b$
- $\nabla W\ RA = [2.1, -0.3]$
- $\nabla W\ RB = [1.5, 0.4]$
- $\sigma(\Delta) = 0.88$
- $\eta = 0.01$.

Compute ∇ϕ L:

$$\begin{aligned}\nabla\phi\ L &= (0.88-1)\cdot([2.1,-0.3]-[1.5,0.4]) \\ &= (-0.12)\cdot[0.6,-0.7] \\ &= [-0.072, 0.084]\end{aligned}$$

Update W:

$$W_{new} = W - \eta \cdot [-0.072, 0.084] = W + [0.00072, -0.00084]$$

The intuition behind Gradient Descent is to adjust ϕ to minimize L(ϕ), i.e., maximize Δ.

That is,

- If RA is too close to RB (Δ is small), the gradient ∇ϕ L is large, forcing RA to increase and RB to decrease.
- If RA ≫ RB(Δ is large), the gradient diminishes, stabilizing training.

4. Visualization of reward difference (Δ) vs. loss

The loss decreases as the reward difference **Δ** increases:

Δ (Reward Difference)	Loss (-log σ(Δ))	Effect
0.0	0.693	Loss = 0.693, Δ = 0
1.0	0.313	Loss = 0.313, Δ = 1
2.0	0.126	Loss = 0.126, Δ = 2
3.0	0.048	Loss = 0.048, Δ = 3

The loss decreases exponentially as **Δ** increases, incentivizing the model to maximize the gap between good and bad responses.

Conclusion

Reward modeling significantly improves training language models by integrating human preferences into the learning process. As a result, this method significantly minimizes the discrepancy between mathematical logic and human cognition in many critical ways.

Key takeaways

- **Human-centric learning**: Models learn to align with human preferences instead of just optimizing for likelihood-based accuracy.
- **Measurable quality assessment**: The reward function provides a clear way to assess response quality consistently.
- **Continuous optimization**: Gradient descent helps refine model parameters based on human feedback.
- **Preference-based differentiation**: The model distinguishes between preferred and non-preferred responses, rewarding only useful ones.
- **Scalable human input**: Reward modeling enables human preferences to be applied efficiently at scale during training.

Author(s)

- Sowmyaa Gurusamy

Other Contributor(s)

- Malika Singla
- Lakshmi Holla