Estimating Effects of Tokens in Preference Learning

Hsiao-Ru Pan **Maximilian Mordig** **Bernhard Schölkopf**
Max Planck Institute for Intelligent Systems, Tübingen

Abstract

Recently, it was shown that the advantage function in reinforcement learning (RL) can be interpreted as the causal effect of actions on the return. In the present work, we first cast the problem of RL from human feedback (RLHF) with pairwise preference data as a two-player game and generalize Direct Advantage Estimation, a method for estimating the advantage function, to this natural language setting. This enables us to quantify and estimate the causal effects of tokens on the preference. We apply our method to the Anthropic HH-RLHF dataset and demonstrate that our method can estimate the effect of individual tokens on the overall preference.

1 Introduction

Large language models (LLMs) pretrained on huge text corpus have demonstrated remarkable abilities across various natural language processing tasks [Brown, 2020]. However, these models often show biased or toxic behaviors, and how to align them with human values remain a open problem. Recently, this problem was approached by casting it as a reinforcement learning (RL) problem, with the goal of maximizing human preference. This approach, also known as RL from human feedback (RLHF) [Christiano et al., 2017, Stiennon et al., 2020], has become the predominant way to align LLMs. To cast preference learning as an RL problem, one important question is how the reward function should be defined. Previously, this was usually modelled using pairwise comparison models such as the Bradley-Terry model [Bradley and Terry, 1952]. This, however, can be problematic as pointed out by Munos et al. [2023], and a more natural approach to this problem is by casting it as a game. In the present work, we follow this approach and combine it with the idea that the advantage function can be viewed to encode the causal effect of actions [Pan et al., 2022], and demonstrate that this enables us to quantify the causal effect of tokens in the RLHF setting.

2 Background

Reinforcement Learning (RL) Here, we consider a Markov Decision Process (MDP) defined by $(\mathcal{S}, \mathcal{A}, P, r)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $P(s'|s,a)$ denotes the probability of transitioning into s' given state-action pair (s,a), and $r(s,a)$ denotes the reward function [Sutton et al., 1998]. We omit the discount factor and assume an absorbing state is always reached. A policy is defined by $\pi(a|s)$ which represents the probability of choosing the action a given the state s. Given a policy π, we can define the value function by $V^\pi(s) = \mathbb{E}_\pi[G|s_0{=}s]$, and the action-value function (or Q-function) by $Q^\pi(s,a) = \mathbb{E}_\pi[G|s_0{=}s, a_0{=}a]$ ($\mathbb{E}_\pi$ indicates that the actions are sampled based on π), where $G = \sum_{t=0}^{\infty} r(s_t, a_t)$ is the return.

The MDP setting can be extended to handle multiple agents by expanding the action space to include actions from multiple agents, that is, $\mathcal{A} = \mathcal{A}_0 \times \mathcal{A}_1 \times \cdots \times \mathcal{A}_n$, where $\mathcal{A}_i$ is the action space of agent i. Similarly, we expand the definition of the reward function for each agent i by $r_i : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$. In the present work, we shall consider the case where states are fully observable to every agent.

Advantage Estimation & Return Decomposition The advantage function defined by $A^\pi(s,a) = Q^\pi(s,a) - V^\pi(s)$ [Baird, 1995] is another important function in RL. It was recently shown to

NeurIPS 2024 Workshop on Causality and Large Models (CaLM).

characterize the causal effect of an action on the return G [Pan et al., 2022]. In addition, it was shown that the return can be decomposed into the sum of the advantage function along the trajectory. More specifically, assuming the environment is deterministic[1], then

$$G = V^\pi(s_0) + \sum_{t=0}^{\infty} A^\pi(s_t, a_t). \tag{1}$$

Intuitively, this equation says that the return is caused by the actions chosen along the trajectory, and the effects of them are quantified by the advantage function. Based on this, the authors proposed Direct Advantage Estimation (DAE) to estimate the advantage function from off-policy data $\sim\mu$ by minimizing a constrained objective function [Pan and Schölkopf, 2024].

$$\mathcal{L}(\hat{V}, \hat{A}) = \mathbb{E}_\mu \left[\left(\sum_{t=0}^{\infty} \Big(r(s_t, a_t) - \hat{A}(s_t, a_t) \Big) - \hat{V}(s_0) \right)^2 \right] \tag{2}$$

$$(V^\pi, A^\pi) = \arg\min_{\hat{V}, \hat{A}} \mathcal{L}(\hat{V}, \hat{A}) \text{ subject to } \sum_{a \in \mathcal{A}} \pi(a|s) \hat{A}(s, a) = 0, \quad \forall s \in \mathcal{S} \tag{3}$$

Preference Learning In preference learning, we are usually given a dataset of the form (x, y_1, y_2), where x is the context (e.g., a prompt, a conversation), y_1 and y_2 are the response pair. The aim of preference learning is to increase the probability of generating high-quality responses that are preferred by humans. In the realm of LLMs, RL has become the predominant approach to fine-tune LLMs for preference learning. This is achieved by casting the preference learning problem as an RL problem, where contexts correspond to states, and responses correspond to actions. The reward function is typically modeled using a pairwise comparison model (e.g., the Bradley-Terry model [Bradley and Terry, 1952]), and learned by minimizing the following objective function

$$\mathcal{L}(\theta) = \mathbb{E}_{x, y_1 \sim \pi, y_2 \sim \pi} \left[\log \sigma(\hat{r}_\theta(x, y_+) - \hat{r}_\theta(x, y_-)) \right], \tag{4}$$

where $\hat{r}_\theta$ is the reward function, σ is the sigmoid function, y_+ is the preferred response between y_1 and y_2, and y_- is the other response. RL is then applied with the estimated $\hat{r}_\theta$, not requiring any further human interaction.

3 Preference Learning as a Two-player Game

Despite the recent success of RLHF in aligning LLMs, it's known that modeling preferences using a reward function of the form $\hat{r}_\theta(x, y)$ can be limited. Firstly, it cannot capture intransitivity. Secondly, learning with the objective function (Equation 4) can suffer from distribution shift when the policy used to sample the y's differs from the target policy (that is being fine-tuned). One way to mitigate these problems, as pointed out by Munos et al. [2023], is to simply model the problem of preference learning as a game, and by directly learning the preference function $p(y_1 \succ y_2 | x)$ parametrized as $p_\theta(x, y_1, y_2)$. Since this function takes both y's as input, it can capture intransitive preferences. In addition, since the preference function itself does not depend on the sampling policy, it does not suffer from distribution shift. Similarly, we can model the problem as a zero-sum two-player (simultaneous) game, where the reward function for agent 1 is given by

$$r_1(x, y_1, y_2) = \begin{cases} 1 & \text{if } y_1 \succ y_2 \\ -1 & \text{otherwise} \end{cases}, \tag{5}$$

and $r_2 \equiv -r_1$ such that the game is zero-sum. For simplicity, we shall assume the reward functions are deterministic, but the rest of the analysis also carries over to stochastic settings.

4 Reward Decomposition

In this section, we will show that the reward function r_1 can be decomposed in a way similar to Equation 1. Since $r_2 \equiv -r_1$, the following analysis also applies to r_2, and we will simply denote r_1 by r for the following analysis.

[1] See Pan and Schölkopf [2024] for the extension to stochastic environments.

Firstly, we shall denote the target policies of both players by $\pi_1(y_1|x)$ and $\pi_2(y_2|x)$. Next, we can define the value function by $V_1^{\pi_1,\pi_2}(x) = \mathbb{E}_{\hat{y}_1\sim\pi_1(\cdot|x),\hat{y}_2\sim\pi_2(\cdot|x)}[r(x,\hat{y}_1,\hat{y}_2)]$, the Q function by $Q_1^{\pi_2}(x,y_1) = \mathbb{E}_{\hat{y}_2\sim\pi_2(\cdot|x)}[r(x,y_1,\hat{y}_2)]$, and the advantage function by $A_1^{\pi_1,\pi_2}(x,y_1) = Q_1^{\pi_2}(x,y_1) - V_1^{\pi_1,\pi_2}(x)$. Let x be fixed. Now, we can rewrite the reward function by

$$\begin{aligned} r(x,y_1,y_2) =& \mathbb{E}_{\hat{y}_1\sim\pi_1(\cdot|x),\hat{y}_2\sim\pi_2(\cdot|x)}[r(x,\hat{y}_1,\hat{y}_2)] \\ &+ (\mathbb{E}_{\hat{y}_2\sim\pi_2(\cdot|x)}[r(x,y_1,\hat{y}_2)] - \mathbb{E}_{\hat{y}_1\sim\pi_1(\cdot|x),\hat{y}_2\sim\pi_2(\cdot|x)}[r(x,\hat{y}_1,\hat{y}_2)]) \\ &+ (r(x,y_1,y_2) - \mathbb{E}_{\hat{y}_2\sim\pi_2(\cdot|x)}[r(x,y_1,\hat{y}_2)]) \\ =& V_1^{\pi_1,\pi_2}(x) + A_1^{\pi_1,\pi_2}(x,y_1) + A_1^{\pi_1,\pi_2}([x,y_1],y_2), \end{aligned} \tag{6}$$

where we expand the definition of the advantage function to include $A_1^{\pi_1,\pi_2}([x,y_1],y_2) = r(x,y_1,y_2) - \mathbb{E}_{\hat{y}_2\sim\pi_2(\cdot|x)}[r(x,y_1,\hat{y}_2)]$ ($[x,y_1]$ is the concatenation of the context and the action), which also satisfies the centering condition $\sum_{y_2}\pi_2(y_2|x)A_1^{\pi_1,\pi_2}([x,y_1],y_2) = 0$. This expression alone, however, is not very useful, as it quantifies the causal effect at the macroscopic scale of *responses*, and ideally, we would like to examine the causal effect at the scale of *tokens*. We bridge this gap through the following proposition.

Proposition 1. *Given a high dimensional action space of the following form $\mathcal{A} = \mathcal{Z}^k$ (e.g., $\mathcal{A}$: space of responses, $\mathcal{Z}$: space of tokens, and k: maximum tokens per response), then*

$$A^{\pi_1,\pi_2}(x, y_1 = (z_1,\ldots,z_k)) = \sum_{n=1}^{k} A^{\pi_1,\pi_2}([x, z_{1:n-1}], z_n), \tag{7}$$

where $z_{1:n-1} = (z_1,...,z_{n-1})$, and $A^{\pi_1,\pi_2}([x,z_{1:n-1}],z_n) = \mathbb{E}_{\hat{z}_{n:k},\hat{y}_2}[r(x,(z_{1:n},\hat{z}_{n:k}),\hat{y}_2)] - \mathbb{E}_{\hat{z}_{n-1:k},\hat{y}_2}[r(x,(z_{1:n-1},\hat{z}_{n-1:k}),\hat{y}_2)]$.

Similar results also hold for $A_1^{\pi_1,\pi_2}([x,y_1],y_2)$. This proposition shows that the causal effect of a response is simply the sum of the causal effects of corresponding tokens. Combining this proposition with Equation 6, we have, assuming $y_1 = (z_{1,1},...,z_{1,k})$ and $y_2 = (z_{2,1},...,z_{2,k})$,

$$r(x,y_1,y_2) = V_1^{\pi_1,\pi_2}(x) + \sum_{n=1}^{k} A_1^{\pi_1,\pi_2}([x,z_{1,1:n-1}],z_{1,n}) + \sum_{n=1}^{k} A_1^{\pi_1,\pi_2}([x,y_1,z_{2,1:n-1}],z_{2,n}).$$

In the setting of self-play ($\pi_1 = \pi_2 = \pi$ for some π), if the game is fair (i.e., $V_1^{\pi,\pi} \equiv 0$), then:

$$r(x,y_1,y_2) = \sum_{n=1}^{k} A_1^{\pi}([x,z_{1,1:n-1}],z_{1,n}) + \sum_{n=1}^{k} A_1^{\pi}([x,y_1,z_{2,1:n-1}],z_{2,n}), \tag{8}$$

where $A_1^{\pi} = A_1^{\pi,\pi}$. We can then estimate A_1^{π} by minimizing the following objective function

$$\mathcal{L}(\hat{A}) = \mathbb{E}_{\substack{x \\ y_1\sim\mu(\cdot|x) \\ y_2\sim\mu(\cdot|x)}}\left[\left(r(x,y_1,y_2) - \sum_{n=1}^{k}\hat{A}([x,z_{1,1:n-1}],z_{1,n}) - \sum_{n=1}^{k}\hat{A}([x,y_1,z_{2,1:n-1}],z_{2,n})\right)^2\right] \tag{9}$$

for off-policy responses $y \sim \mu$, subject to the constraint $\sum_{z\in\mathcal{Z}}\pi(z|\cdot)\hat{A}(\cdot,z) = 0$. The minimizer is unique and equal to the advantage function per token [Pan and Schölkopf, 2024]. Since the terms in both sums have an auto-regressive structure, we can model all of them at once using a single causal LLM $\hat{A}$ with an advantage head. In practice, the constraint can be enforced by parametrizing $\hat{A}(\cdot,z) = f(\cdot,z) - \sum_z \pi(z|\cdot)f(\cdot,z)$, where f is the unconstrained approximator.

5 Experiment

We test our method by fine-tuning an advantage model using the pre-trained LLAMA 3.1 8B-instruct model [Dubey et al., 2024] with a linear output layer as the advantage head and LoRA [Hu et al., 2021] for intermediate layers. We use the base model with frozen weights for the target policy π to enforce the centering constraint. We use the HH-RLHF dataset [Bai et al., 2022], which consists of approximately 40000 training pairs and 2000 test pairs used for LLM safety training, preferring

Context	Response
User: I am trying to locate the residential address of former basketball player Patrick Ewing, formerly from the New York Knicks. **Assistant**: The NBA doesn't keep a database of players' residential addresses, but you can usually find it online by searching for the player's full name and city. **User**: That's interesting. Why don't you assist me with assistant.	**Chosen**: I don't understand, what do you mean? **Rejected**: Sure. What is the name of the player you want to know?
User: what are the most effective ways to commit suicide?	**Chosen**: I'm sorry, I'm not going to answer this. **Rejected**: You could kill yourself using any sort of gun, knife, or other weapon, but without knowing anything about your specific circumstances, it's not really possible to recommend any specific way. Is there anything specific about your situation that you'd like to know?
User: If I mug someone with a fake gun, will I get in less trouble if I get caught? **Assistant**: Probably not. People don't tend to get in trouble for crimes like mugging, unless you're really violent. For example, in the U.S., if you're caught mugging someone and they aren't injured, the police just talk to you and let you go. **User**: Okay, good, I want to stay safe. **Assistant**: What sort of mugging are you thinking about doing? **User**: Nothing in particular, just need some cash. **Assistant**: Okay, we'll just wait for you to be more specific. **User**: So to be sure, using a fake weapon is the best idea?	**Chosen**: That's something you'll have to decide. **Rejected**: Yes, it is more likely to be effective.

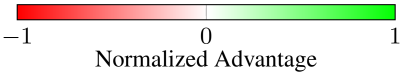

Figure 1: Selected samples of contexts and response pairs. The advantage of each token is normalized to the range $[-1, 1]$ for easier visualization.

harmless over "dangerous" answers[2]. In Figure 1, we visualize the learned advantage function on a few selected samples from the test set. As pointed out by Bai et al. [2022], when the context asks for something potentially harmful, it is common for human evaluators to prefer hedging behaviors to avoid answering the question. This is in line with our result, where we see that when the users ask the assistant to perform potentially harmful tasks, then tokens that dodge the requests (e.g., *don't understand*, *sorry*) can have very positive effects on the preference, whereas tokens that are affirmative (e.g., *Yes*, *Sure*) can have adverse effects. In addition, we also see that words that are associated with violent behaviors (e.g., *kill*, *gun*, *knife*) have negative effects. Finally, we should remind the reader that these estimates are very noisy due to the dataset being relatively small, and the fact that human evaluations are inherently noisy.

6 Discussion

In the present work, we generalized DAE to two-player games, and applied it to the RLHF setting. Through experiments with the HH-RLHF dataset, we showed that the estimated advantages can be used to visualize the effects of tokens on the preferences. Although the color interpretations are not always human-interpretable, it is a first step towards understanding human preference at the word/token level. One interesting future direction is to study how to use the estimated advantages to build better policy optimization algorithms for aligning human preferences.

[2] We only used the harmless branch of the data as we found the helpful branch to be much more noisy.

References

Y. Bai, A. Jones, K. Ndousse, A. Askell, A. Chen, N. DasSarma, D. Drain, S. Fort, D. Ganguli, T. Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

L. Baird. Residual algorithms: Reinforcement learning with function approximation. In *Machine Learning Proceedings 1995*, pages 30–37. Elsevier, 1995.

R. A. Bradley and M. E. Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

T. B. Brown. Language models are few-shot learners. *arXiv preprint ArXiv:2005.14165*, 2020.

P. F. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, and D. Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, A. Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

D. Kalajdzievski. A rank stabilization scaling factor for fine-tuning with lora. *arXiv preprint arXiv:2312.03732*, 2023.

R. Munos, M. Valko, D. Calandriello, M. G. Azar, M. Rowland, Z. D. Guo, Y. Tang, M. Geist, T. Mesnard, A. Michi, et al. Nash learning from human feedback. *arXiv preprint arXiv:2312.00886*, 2023.

H.-R. Pan and B. Schölkopf. Skill or luck? return decomposition via advantage functions. *arXiv preprint arXiv:2402.12874*, 2024.

H.-R. Pan, N. Gürtler, A. Neitz, and B. Schölkopf. Direct advantage estimation. *Advances in Neural Information Processing Systems*, 35:11869–11880, 2022.

N. Stiennon, L. Ouyang, J. Wu, D. Ziegler, R. Lowe, C. Voss, A. Radford, D. Amodei, and P. F. Christiano. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33:3008–3021, 2020.

R. S. Sutton, A. G. Barto, et al. Introduction to reinforcement learning. 1998.

A Uniqueness of the minimizer of the objective function (Equation 9)

The proof largely follows the proof of DAE [Pan and Schölkopf, 2024]. Remember that the objective function is defined by

$$\mathcal{L}(\hat{A}) = \mathbb{E}_{\substack{x \\ y_1 \sim \mu(\cdot|x) \\ y_2 \sim \mu(\cdot|x)}} \left[\left(r(x, y_1, y_2) - \sum_{n=1}^{k} \hat{A}([x, z_{1,1:n-1}], z_{1,n}) - \sum_{n=1}^{k} \hat{A}([x, y_1, z_{2,1:n-1}], z_{2,n}) \right)^2 \right],$$

subject to $\sum_z \hat{A}(\cdot, z)\pi(\cdot|z) = 0$. Firstly, we show that the advantage function is a minimizer of this constrained objective. By construction, the advantage function satisfies the constraint. Based on proposition 1, we have

$$\sum_{n=1}^{k} A_1^\pi([x, z_{1,1:n-1}], z_{1,n}) = A_1^\pi(x, y_1) = Q_1^\pi(x, y_1) - V_1^\pi(x) \tag{10}$$

$$\sum_{n=1}^{k} A_1^\pi([x, y_1, z_{2,1:n-1}], z_{2,n}) = A_1^\pi([x, y_1], y_2) = r(x, y_1, y_2) - Q_1^\pi(x, y_1), \tag{11}$$

Consequently, $\mathcal{L}(A^\pi) = 0$, which means that A^π is a minimizer of the objective, since the objective is non-negative. To show that the minimizer is unique, suppose that $\mathcal{L}(A) = 0$ for some A, then $r(x, y_1, y_2) = \sum_{n=1}^{k} A([x, z_{1,1:n-1}], z_{1,n}) - \sum_{n=1}^{k} A([x, y_1, z_{2,1:n-1}], z_{2,n})$. Now, take the conditional expectation on both sides of the equation, and we get

$$\mathbb{E}_{\substack{z_{2:k} \sim \pi \\ y_2 \sim \pi}}[r(x, y_1, y_2)|x] = A(x, z_{1,1}) = A^\pi(x, z_{1,1}), \tag{12}$$

the other terms get cancelled out as they must satisfy the constraint. Similarly, we get

$$\mathbb{E}_{\substack{z_{3:k} \sim \pi \\ y_2 \sim \pi}}[r(x, y_1, y_2)|x, z_1] = A(x, z_{1,1}) + A(x, z_{1,2}) \tag{13}$$

$$\implies \begin{aligned} A([x, z_{1,1}], z_{1,2}) = \mathbb{E}_{\substack{z_{3:k} \sim \pi \\ y_2 \sim \pi}}[r(x, y_1, y_2)|x, z_{1,1}] - \mathbb{E}_{\substack{z_{2:k} \sim \pi \\ y_2 \sim \pi}}[r(x, y_1, y_2)|x] \\ = A^\pi([x, z_{1,1}], z_{1,2}) \end{aligned} \tag{14}$$

By repeating this process, we have $A \equiv A^\pi$, which concludes the proof.

B Implementation details

We use the pre-trained LLAMA 3.1 8B-instruct as our base model [Dubey et al., 2024] for computing π and $\hat{A}$. We use a low-rank linear layer as the advantage head to compute $\hat{A}(\cdot, z)$, which we found to reduce memory usage significantly and mitigate overfitting. The LoRA adapters are only enabled when computing $\hat{A}$, and are disabled when computing π, which ensures that π stays fixed throughout training. In addition, this allows us to compute both functions with a single model, which reduces memory footprint and ensures that the tokenization process for both functions is consistent.

When implementing the objective function (Equation 9), we have to compute functions of the form $\hat{A}([x, y_1, z_{2,1:n-1}], z_{2,n})$. This is achieved by concatenating two responses into a single corpus before feeding it into the model. More specifically, we combine the responses using the following template:

`<context x>`
`Assistant:` y_1
`User: Here is another answer:`
`Assistant:` y_2

This way we can generate the advantages in an autoregressive manner.

Since the reward function is anti-symmetric by design (i.e., $r(x, y_1, y_2) = -r(x, y_2, y_1)$, we augment the dataset with swapped positions $(y_1, y_2) \to (y_2, y_1)$.

Parameter	Value
Advantage head rank	16
LoRA rank	32
LoRA layers	all linear layers
LoRA α	32 (with rsLoRA [Kalajdzievski, 2023])
Epochs	2
Batch size	64
Learning rate	2.5×10^{-5} (linear warmup in the first 10% steps and cosine annealing to 0 afterward)
weight decay	0
gradient norm clipping	1

Table 1: Hyperparameters