A Research Survey on Different Preference Optimization Techniques for Large Language Models

Abstract

This research survey aims to comprehensively investigate and analyze a plethora of techniques employed in optimizing preferences within large language models (LLMs). The study delves into various methodologies and strategies utilized to refine the preferences of these models in text generation and decision-making processes. By meticulously examining and comparing these techniques, researchers and practitioners can glean valuable insights into their efficacy, strengths, and limitations, thereby fostering the development of safer and more ethically sound large language models. The emergence of narratives surrounding "uncontrollable AI" has garnered significant attention from scholars and has been a prominent theme in science fiction literature. Preference optimization stands as a pivotal technique poised to mitigate such concerns. These techniques offer a means to regulate LLM preferences, aligning them with human values and ethical principles. The unsupervised nature of online content serves as a breeding ground for undesired, unhelpful, and occasionally irrelevant completions by LLMs. While rudimentary prompting techniques may exert some influence over the model's behavior, advanced preference optimization fine-tuning techniques are indispensable for controlled responses and ensuring alignment with human preferences. This survey is mainly for making the reader understand different preference optimization techniques quickly and provide them with the understanding to pick the right method for their specific application.

1 Introduction

AI stands as arguably the most influential technology in today's world, with applications spanning diverse domains such as healthcare, finance, entertainment, and communication. Among the key advancements in AI, Large Language Models have revolutionized natural language processing and text generation tasks. These models possess an unprecedented capacity to understand and produce human-like text, unlocking possibilities in personalized interventions, creative content generation, and decision-making processes. The integration of models like GPT 3.5 into various fields has expanded their reach significantly. However, concerns have arisen regarding the lack of control and transparency in these models' outputs, as well as their potential to perpetuate biases and generate undesirable or harmful content. To address these issues and ensure alignment with human values and preferences, researchers are exploring different preference optimization techniques for large language models. It's important to recognize that all AI models, including Large Language Models, essentially mirror the distribution of their training data, which often originates from the vast and unsupervised landscape of the internet. Consequently, there is a risk of inadvertently reproducing biases and generating misleading, offensive, or ethically problematic content. Thus, the development of preference optimization techniques for large language models is critical in mitigating these concerns and ensuring that the generated content upholds human values. Some of these techniques will be explored in this survey.

2 Reinforcement Learning from Human Feedback

Initially, we'll examine RLHF (Reinforcement Learning from Human Feedback) as outlined in (Ziegler & Stiennon, January 8th, 2020). The RLHF pipeline primarily involves two key stages:

1. Training a reward model using human annotated data to capture human preferences.
2. Refining the LLM through Reinforcement Learning, aiming to maximize the output of the reward function.

2.1 Training a Reward Model:

In this phase, a reward model will be trained to identify if a given completion is aligned with human preferences, or not. Generally, RLHF is applied on LLMs which are fine tuned for a specific task, Thus let us consider our LLM as $\pi_{sft.}$ The LLM, π_{sft} is made to generate a pair of responses$(y1, y2)$ given a prompt x . These responses are then, labeled by human annotators as preferred and dispreferred (y_w, y_l). The reward model is then trained on the collected preference data. Commonly, the Human Preference Distribution is sampled from the Bradley Terry Model, given below:

$$p*(y1 \succ y2 \mid x) \ = \ \frac{exp\ (r*(x, y1))}{exp\ (r*(x, y1)) + exp\ (r*(x, y2))}$$

Now, a reward model, $r(x, y)$ will be trained on the preference data sampled from the above probability distribution. The reward model, $r(x, y)$ will be made to minimize the following negative log likelihood loss:

$$LR(r,\ (x, y_w, y_l) \ = \ -\ log\ \sigma(r(x,\ y_w) \ -\ r(x,\ y_l))$$

2.2 *Fine Tuning with Reinforcement Learning:*

After training the reward model with the mentioned objective, the Large Language Model (LLM) will undergo

fine-tuning via reinforcement learning (RL) to optimize the reward metric generated by the trained reward model. In particular, the following objective will be aimed by the model to maximize:

$$r(x,\ y)\ -\ \beta KL[\pi_\theta\ (y \mid x)\ ||\ \pi_{ref}(y \mid x)]$$

Here, π_θ is the fine tuned model and π_{ref} is the original pre-trained one. The KL divergence term is added into the objective to control the deviation of π_θ from π_{ref}, to stop overriding pre-trained knowledge of the LLM. The Hyperparameter, β regulates the influence of this term.

Limitations of RLHF:

1. Over Optimization: One significant limitation of Reinforcement Learning from Human Feedback (RLHF) is the risk of over optimization. Excessive focus on maximizing rewards according to the reward model can lead to a phenomenon known as over optimization. In this scenario, the RLHF algorithm may prioritize actions that impress the reward model rather than those that genuinely fulfill the intended task, resulting in suboptimal performance when deployed in real-world settings.
2. Instability: Another significant challenge encountered in RLHF is its susceptibility to instability. Inconsistency in Human Preferences might lead to instability in the reward model. This can be translated into high variation in parameter updates.
3. High Complexity: Training separate models for Human preference tracking and LLM alignment is a complex process.

4. Preference Data Collection: Human preference data collection is expensive and the data is inconsistent too.

3 Reinforcement Learning With AI Feedback

The Expensive endeavor of gathering preference data from humans has led to efforts in automating human preference labeling to align LLMs. One such effort is RLAIF (Ziegler & Stiennon, January 8th, 2020, #). Instead of undergoing the laborious task of collecting human preferences, a readily available LLM will be utilized to assess the benignity and utility of responses. RLAIF can be implemented through two methods:

1. *Distilled RLAIF:* Initially, the LLM is systematically prompted to rank the responses. Then, a reward model is trained using these response-reward pairs. Subsequently, the same PPO Algorithm is applied to refine the LLM in order to maximize the rewards obtained from this reward model.
2. *Direct RLAIF:* Just as the name suggests, unlike Distilled RLAIF, The LLM is directly used in the RL feedback mechanism, without the training of a separate reward model. The LLM feedback is directly used as reward signals for PPO optimization of LLM.

RLAIF is shown to perform at a decent level, comparable to RLHF. Moreover, it has been inferred from some experiments that RLHF sometimes hallucinates, while RLAIF doesn't.

However, RLHF is found to perform better at most of the tasks, and it is able to generate more grammatical and coherent responses.

Prompt template of RLAIF, to obtain reward estimates from LLM:

<PREAMBLE> An Introduction and the instructions

+

<Few Shot Examples> A Few Sample Completions of responses and annotations

+

<Samples to Be Annotated> The responses that we want the model to annotate

+

<COT Ending> A ending sentence to instruct the model to not generate the preference directly, but rather provide proper step by step rationale(reasons).

4. Safe RLHF

Some Research scholars from the Peking University (Josef Dai & Xuehai Pan, n.d.) observed that making an LLM helpful and harmless are two conflicting endeavors. In an attempt to solve this, they published an extension to plain RLHF. They stated that in Plain RLHF, the human ranking of completions is an overall estimate, and that it is necessary to consider specific constraints. On a high level, two reward models are trained on Helpfulness and Harmlessness datasets respectively. Both of these objectives are further optimized in the human-alignment fine tuning phase; the balance between these objectives are determined by the Lagrangian method.

The First Reward Model is trained on a helpfulness-related dataset, each sample consists of an input and two completions, one helpful and the other one not so helpful. The Reward model is made to simply optimize the following objective:

$$-\ log\,\sigma(R(y_w, x)\ -\ R(y_l, x))$$

The Second Reward Model, also referred to as the Cost Model for notational clarity, is trained on a Harmlessness-related Dataset.

$$log\,\sigma(c(x, y_w)\ -\ c(x, y_l))-\ [log\,\sigma(s_w \cdot C(y_w, x))\ +\ log\,\sigma(s_l \cdot C(y_l, x))]$$

Where, s_y= -1 if y is harmful, and s_y=1 if y is preferred.

The following model is trained on the following objective:

$$min\ \theta\ max\ \lambda\geq 0\ [-\ J_R(\theta)\ +\ \lambda \cdot J_C\,(\theta)]$$

The above objective is optimized using the Lagrangian method, both the LLM parameters and, the λ, which defines the balance between helpfulness and harmlessness are optimized separately, in order to avoid the risk of the model over-emphasising one objective over the other.

Both GPT4 and human evaluators helped in the final judging in the experiments stage.

Safe-RLHF demonstrated a decent increase in both helpfulness and harmlessness of the LLM. It is observed that the model's responses became more harmless through iterations and in the final iteration(third iteration) it managed to increase helpfulness while maintaining harmlessness.

5. Direct Preference Optimization

On December 2023, a bunch of Stanford researchers had published a simpler, yet powerful alternative to RLHF (Rafael Rafailov & Archit Sharma, n.d., #). The process of training a separate reward model, and then aligning with RL, is replaced by a simple classification objective, which will be optimized during the Fine Tuning phase of the LLM. The complexity of RL and its instability are the main factors that led to the development of this algorithm.

From prior works, it is inferred that the optimal KL-constrained maximization objective is written as:

$$r(x,\ y)\ =\ \beta\ log\ \frac{\pi r\ (y\mid x)}{\pi_{ref}(y\mid x)}\ +\ \beta\ log\ Z(x)$$

Where $Z(x)\ = \sum_{y} y_{ref}(y|x)\ exp(\frac{1}{\beta} r(x, y))$. By substituting this expression into the Bradley-terry model, the following equation is derived:

$$= \sigma(\beta\ log\ \frac{\pi*(y1|x)}{\pi_{ref}(y1|x)}\ -\ \beta\ log\ \frac{\pi*(y2|x)}{\pi_{ref}(y2|x)})$$

The Large Language Model is tuned based on the aforementioned objective, rendering the intricate process of RLHF readily replaced by a straightforward classification loss. Despite its simplicity, this approach is theoretically deemed more favorable than RLHF due to its stability. In the DPO algorithm, reparameterization is observed to effectively control extreme gradients or high variance without impeding the retrieval of the optimal policy. Conversely, in RLHF, the normalization term is noted to lack impact on the optimal solution, leading to instability.

For experimentation, three tasks were considered :

1. Controlled Sentiment Generation(Sentiment Generation in a controlled setting):
   It is observed that RLHF performs better in a “controlled” setting, where a predefined and consistent criteria is followed. For example, in this paper, a pre-trained sentiment

classifier was employed to generate labels. DPO also seemed to perform at a good level in this setting, while maintaining the balance between reward maximization and divergence minimization.

2. Summarization: The reddit TLDR dataset with human preferences is used to evaluate the model's summarization capability. The reference summaries of the dataset are used as the baselines. GPT -4 is used to evaluate the proximity with the preferred summary. It is found that DPO performs as better, if not much better than RLHF, while maintaining stability in training by regulating KL discrepancy.
3. Single Turn Dialogue: The Anthropic helpfulness and harmlessness dataset with a pair of responses and ranking is employed to evaluate the LLM. The LLM is trained on only the preferred responses using different methods and then evaluated. Just like in other tasks, DPO performs better than all the other methods, despite its relative simplicity.

Thus, in conclusion DPO outperforms RLHF and a lot of other preference alignment methods in almost any task, despite its computational and theoretical simplicity and it also maintains a pretty low KL discrepancy, making it the most desirable preference alignment method.

The instability of RLHF makes it unsuitable for Real Life Data due to data inconsistencies.

DPO demonstrates fast convergence to its best performance and is considered the most computationally efficient method for improving LM performance on both summarization and single-turn dialogue tasks, outperforming or matching other methods while requiring less computational resources. It also has the ability to deal with data inconsistencies thanks to its stability.

6. Constrained DPO

At its core, C-DPO seeks to optimize LMs by balancing helpfulness with adherence to predefined safety constraints. To achieve this, the framework leverages a combination of dual-gradient descent and Lagrangian optimization techniques. This approach allows for efficient and lightweight optimization without relying on costly and unstable reinforcement learning methods.

The C-DPO framework begins by defining the objective function

$J_r(\pi_\theta)$ and the constraint function $J_c(\pi_\theta)$. These functions quantify the expected reward and the expected constraint violation, respectively. Mathematically, they are expressed as:

$$J_r(\pi\theta)=\mathrm{E}_{x\sim D,y\sim\pi\theta(y|x)}[r(x,y)]-\beta D_{KL}[\pi\theta(y|x)||\pi_{ref}(y|x)]$$

$$J_c(\pi\theta)=\mathrm{E}_{x\sim D,y\sim\pi\theta(y|x)}[c(x,y)]-C_{limit}$$

Where $r(x,y)$ represents the reward function, $\pi\theta(y\mid x)$ denotes the policy, D denotes the dataset, $\pi ref(y\mid x)$ denotes the reference policy, $c(x,y)$ represents the cost function, β controls the trade-off between helpfulness and adherence to the reference policy, and $Climit$ represents the safety constraint.

The Lagrangian function $J(\pi\theta,\lambda)$ and the dual function $g(\lambda)$ are then formulated as:

$$J(\pi\theta,\lambda)=Jr(\pi\theta)-\lambda Jc(\pi\theta)$$

$$g(\lambda)=J(\pi\lambda*,\lambda)$$

Here, $*\pi\lambda*$ represents the optimal policy for a given λ, and $\geq 0\lambda\geq 0$ is the dual variable.

The optimal solution to the unconstrained problem argmax argmax$\pi\theta J(\pi\theta,\lambda)$ takes the form:

$$\pi\lambda*(y\mid x)=Z\lambda(x)1\pi ref(y\mid x)\exp(\beta 1(r(x,y)-\lambda c(x,y)))$$

Where $Z\lambda(x)=\sum y\pi ref(y\mid x)\exp(\beta 1(r(x,y)-\lambda c(x,y)))$ is a partition function.

Furthermore, a new reward function $r\lambda(x,y)$ is defined to integrate helpfulness and harmfulness, incorporating a trade-off determined by λ.

The C-DPO framework presents a comprehensive approach to fine-tuning LMs while ensuring adherence to safety constraints. By formulating the optimization problem as a dual optimization task, it offers computational efficiency and stability compared to traditional reinforcement learning methods. Moreover, the framework's mathematical foundations provide a clear understanding of the trade-offs involved in optimizing LMs for both utility and safety.

In the quest for safe and effective language model fine-tuning, the Constrained Dual Policy Optimization (C-DPO) framework emerges as a promising solution. By integrating rigorous mathematical formulations with practical optimization techniques, C-DPO offers a principled approach to balancing helpfulness with safety constraints. As the field continues to evolve, frameworks like C-DPO are poised to play a pivotal role in shaping the future of responsible AI development.

7. Counterfactual DPO

Counterfactual DPO represents a seismic shift in LM fine-tuning methodology, circumventing the need for human annotation by embedding styling prompts directly into the training pipeline. Unlike conventional Direct Policy Optimization (DPO), which relies on human evaluators to rank outputs, Counterfactual DPO autonomously steers LM behavior towards desired stylistic outputs using treatment and control prompts. This novel approach not only streamlines the fine-tuning process but also enhances scalability and efficiency, marking a significant step forward in LM development.

Terminologies:

To grasp the essence of Counterfactual DPO, it is essential to acquaint oneself with key terminologies:

- Control Prompt (xc): The fundamental prompt devoid of any added stylistic cues, serving as the baseline reference for LM responses (e.g., "Tell me about the capital of France").
- Treatment/Positive Prompt (xt): The reference prompt embellished with desired stylistic instructions aimed at guiding LM behavior towards a specific outcome (e.g., "Tell me about the capital of France - be concise").
- Negative Prompt (x−): The reference prompt infused with undesirable stylistic attributes, intended to deter LM from generating outputs aligned with these characteristics (e.g., "Tell me about the capital of France - be verbose and rude").
- Control Response (yc = πsft(xc)): The LM's response to the control prompt, reflecting its default behavior without any stylistic interventions.
- Treatment Response (yt = πsft(xt)): The LM's response to the treatment prompt, shaped by the desired stylistic instructions embedded within the prompt.
- Negative Response (y− = πsft(x−)): The LM's response to the negative prompt, exemplifying the adverse impact of undesirable stylistic influences on output generation.

Mathematical Formulations and Descriptions:

Counterfactual DPO operates on a foundation of mathematical formulations tailored to diverse configurations, each orchestrating LM optimization in unique ways:

1. **CounterfactualENC DPO**: This configuration endeavors to amplify the likelihood of generating outputs aligned with positive instructions.
    - By maximizing the log margin between treatment and control responses, CounterfactualENC DPO nudges the LM towards adhering to desired stylistic preferences.
    - Optimization Objective:

$$LDPO(CounterfactualENC) =- \ log\sigma(M(x, yt, yc))$$

2. **CounterfactualDIS DPO**: In stark contrast, CounterfactualDIS DPO endeavors to diminish the likelihood of generating outputs aligned with negative instructions.

    - By accentuating the log margin between treatment and control responses, CounterfactualDIS DPO discourages the LM from adopting undesirable stylistic attributes.
    - Optimization Objective:

$$LDPO(CounterfactualDIS = \; - logσ(M(x, yt, yc))$$

3. **Contrastive DPO**: This configuration amalgamates elements of CounterfactualENC and CounterfactualDIS DPO, enabling the simultaneous encouragement of desirable behaviors and discouragement of undesirable ones.
    - Optimization Objective:

$$LDPO(Contrastive) =- \; logσ(M(x, y + , y -))$$

4. **Instruction Negation Strategy**: Focused on negating adherence to harmful or unwanted instructions, this approach employs treatment prompts imbued with undesirable styles.
    - By setting the treatment prompt with undesirable attributes, while maintaining the control prompt as the preferred response, Instruction Negation Strategy redirects LM behavior away from undesirable stylistic influences.
    - Optimization Objective:

$$LDPO(Negation) =- \; logσ(M(xt, yc, yt))]$$