Aligning Language Models with Human Feedback without Reinforcement Learning

PIC2 - Master in Computer Science and Engineering
Instituto Superior Técnico, Universidade de Lisboa

Martim Filipe Almeida Santos — 95638*
martim.santos@tecnico.ulisboa.pt

Advisor: André Filipe Torres Martins, Francisco Saraiva de Melo

Abstract Large language models (LLMs) are characterized by their remarkable ability to learn extensive world knowledge and generate human-like text across diverse applications. However, these often contain misleading and toxic content, emphasizing the need to align them with human values and preferences to ensure more useful and secure AI systems. A widely employed strategy in numerous prominent models, including OpenAI's GPT-3.5 and GPT-4, involves Reinforcement Learning from Human Feedback (RLHF). While this method has demonstrated impressive outcomes, RLHF's complexity, instability, and sensitivity to hyperparameters challenge its empirical success and usability across various real-life scenarios. In this work, we intend to deeply study and compare the different novel RL-free approaches that focus on overcoming the drawbacks of RLHF, while demonstrating competitive performance in multiple language tasks. Finally, we propose a novel alternative method that combines existing approaches by leveraging their strengths, aiming to outperform these approaches in the language tasks of dialogue, summarization, and machine translation.

Keywords — LLMs · Fine-tuning · Human Feedback · Human preferences · Alignment · RLHF · RL-free Approaches · Supervised Fine-Tuning · NLP · Efficiency · Complexity · Stability

[*] I declare that this document is an original work of my own authorship and that it fulfills all the requirements of the Code of Conduct and Good Practices of the Universidade de Lisboa (https://nape.tecnico.ulisboa.pt/en/apoio-ao-estudante/documentos-importantes/regulamentos-da-universidade-de-lisboa/).

Contents

1 Introduction

Large Language Models (LLMs) represent a significant stride in Artificial Intelligence, particularly in the realm of language processing. These deep learning models are characterized by their remarkable ability to learn extensive world knowledge and generate human-like text across diverse applications. Prominent examples such as the famous OpenAI's GPT series (including GPT-3.5 and GPT-4), Google's PaLM, and Meta's LLaMA, stand out due to their extensive neural networks, containing billions of parameters. This architectural magnitude empowers them to capture and learn language intricacies, encompassing patterns, syntax, and semantics on a large scale. This proficiency is attained through the utilization of enormous datasets during training, coupled with their powerful computational resources. Essentially, LLMs excel in predictive tasks, foreseeing the next word or token in phrase or sequence, showcasing their versatility in tasks like translation, summarization, and even creative content generation.

Their full potential is further realized when fine-tuned for specific tasks. This process consists of exposing the pre-trained language model to task-specific data, allowing it to adjust internal parameters and refine its performance on specific tasks. The model is then tailored to generate more accurate and contextually relevant outputs for targeted applications. One approach consists of fine-tuning the pre-trained models in a supervised manner using labeled data, known as supervised fine-tuning (SFT). A specific strategy that has garnered attention is Instruction Tuning (Wei et al., 2021). By leveraging datasets enriched with instructions and corresponding human-generated completions, this approach has demonstrated improved model usability and generalization (Chung et al., 2022).

While this strategy has led to markedly improved performance, this approach still encounters challenges that extend to various applications. One of the primary issues lies in the misalignment between this fine-tuning objective, typically focused on maximizing the log probability of human high-quality demonstrations, and the nuanced expectations associated with generating high-quality and harmless outputs (Stiennon et al., 2020). Additionally, models may inadvertently amplify biases from the training data, leading to potentially low quality and harmful outputs, limiting their overall effectiveness and acceptance.

Another challenge arises in the expense associated with collecting new supervised samples for practical applications, especially when expert involvement is necessary to produce high-quality data (Dong et al., 2023). Furthermore, these models may also struggle to generate content that is contextually appropriate or aligns with societal values, limiting their overall effectiveness and acceptance.

Addressing the above-mentioned challenges is important for the practical applicability and real-world utility of these models. This entails going beyond conventional supervised fine-tuning approaches to ensure a safer and more harmonious alignment with the nuanced aspects of human expectations and preferences. **Incorporating human preferences** into large language models is essential for the ethical and effective deployment of such systems. It ensures that the generated content is not only accurate but also aligns with societal values, mitigating the risk of inadvertently producing harmful or biased outputs. Additionally, the integration of human feedback contributes to continuous model improvement, fostering a dynamic and adaptive system that can evolve with changing linguistic nuances and user expectations.

Collecting human feedback is then essential to align the model's behavior with human preferences, although it presents challenges in balancing harmlessness and helpfulness. Harmlessness involves avoiding undesirable outputs, while helpfulness is gauged through task-related feedback. Some works explicitly divide the problem of alignment of these models into improving its helpfulness and increasing its harmlessness. The format of feedback, whether numerical, ranking-based, natural language, or other types, also has a significant influence on the expressivity and ease of collection, thereby shaping the model's ability to enhance its performance.

To completely **leverage human feedback**, it is also important to guarantee that the model captures the nuanced details, rankings, and preferences of human evaluators. For instance, when using a reward function to evaluate the outputs generated by the model, we might face some challenges since the reward model itself is a learned proxy for human preferences. If the reward model is not accurate or lacks granularity, human preferences, and details might not be fully captured. Another important aspect is to guarantee that the models are cost-effective in terms of training and data efficiency, avoiding excessive complexity and considering memory-friendliness, eliminating the necessity to load multiple LLMs for training. Furthermore, some naive approaches might focus on positive examples, or preferred responses, without adequately considering negative examples,

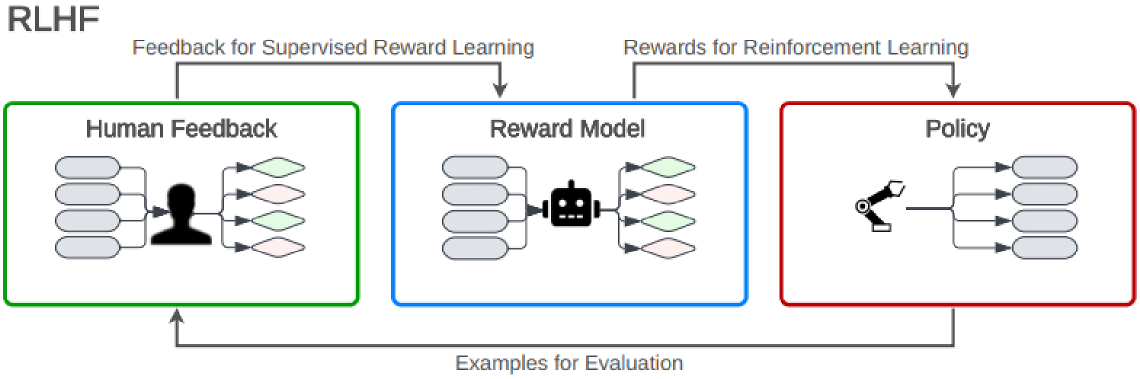

RLHF
Feedback for Supervised Reward Learning
Rewards for Reinforcement Learning
Human Feedback
Reward Model
Policy
Examples for Evaluation

Figure 1: Reinforcement Learning from Human Feedback. Gray, rounded boxes correspond to outputs (e.g., text), and colored diamonds correspond to evaluations. Figure taken from Casper et al. (2023).

which represent undesired responses. This imbalance can compromise the modeling of human preferences considering the possible valuable information that the undesired responses might contain.

Reinforcement Learning from Human Feedback (RLHF) marks a significant departure from traditional methods in refining large language models, introducing a systematic approach that incorporates human preferences and evaluations. This proves especially valuable in various natural language processing tasks such as abstractive English text summarization, where judging summary quality is complex without human judgments. The approach involves training models by refining their understanding through a process called supervised fine-tuning, where they adjust and improve based on specific examples, and incorporate feedback from humans to improve how the models generate responses, essentially guiding the model toward generating more desirable outputs. This evaluative information is then utilized to establish a reward model that reflects human preferences, where the initial model is incentivized or "rewarded" when it produces outputs that align with the desired criteria. Reinforcement learning techniques, such as Proximal Policy Optimization (PPO), are then applied to maximize this estimated reward without drifting too far from the original model. This process enhances the models' capabilities by shaping their learning trajectory to improve the quality and appropriateness of their generated responses. Figure 1 depicts RLHF's three key steps: collecting human feedback, fitting a reward model, and optimizing the policy with RL. This methodology addresses challenges related to context understanding, deciphering hidden meanings, and aligning outputs with ethical considerations and user preferences.

Despite the advantages of RLHF in refining large language models, there are notable drawbacks that warrant consideration, namely its increased complexity, possible instability, and hyperparameters sensitivity. As we delve into the complexities of RLHF, the primary objective remains the alignment of language models with human preferences, balancing maximally useful responses but also delivering safe, accurate, and non-harmful outputs.

In the pursuit of effectively incorporating human preferences via feedback and advancing the development of more sophisticated models, it is crucial to simultaneously maintain simplicity, stability, decreased sensitivity to hyperparameters, efficiency, scalability, and parallelizability. This concurrent focus ensures the enhancement of empirical success and usability of these models across in various real-life applications. Thus, the evolving landscape of natural language processing beckons a delicate balance between addressing the drawbacks inherent in RLHF and charting a course toward models that align seamlessly with human preferences, mitigate harmful outputs, and excel in practical and real-world scenarios.

The exploration of advanced architectures such as **DPO** (Direct Preference Optimization) (Rafailov et al., 2023), **RSO** (Statistical Rejection Sampling Optimization) (Liu et al., 2023), **PRO** (Preference-based Reinforcement Optimization) (Song et al., 2023), and **RRHF** (Rank-based Reward Modeling with Human Feedback) (Yuan et al., 2023) underscores a strategic shift in the pursuit of effective language model training. Traditional RLHF methods often involve the intermediary step of learning a reward model using pairwise preference comparisons before training the policy. These newer paradigms deviate from this norm, aiming for more direct optimization based on human preferences while maintaining efficiency, simplicity, and robustness. Furthermore, they substantiate their viability through competitive performance in comparison to RLHF methods across various tasks.

DPO, for instance, simplifies the process by skipping the step of learning a reward model, demonstrating effectiveness comparable to existing PPO-based RLHF methods in sentiment generation, summarization, and single-turn dialogue tasks. RSO improves DPO by sourcing preference data from the target optimal policy using rejection sampling, enabling a more accurate estimation of the optimal policy. PRO extends the pairwise Bradley-Terry (Bradley and Terry, 1952) comparison to accommodate arbitrary-length preference rankings, teaching the LLM to prioritize responses that closely align with human preferences and fully utilize the richness of human rankings. RRHF introduces a rank-based approach, further refining the alignment process. More information about these models in Section 2.6.

1.1 Work Objectives

Despite these innovative strides and the rapid development of alternative approaches, some problems and challenges persist. One common hurdle is efficiently handling diverse preferences and ensuring effective exploration during training, in various generation tasks.

Specifically, DPO may struggle with limited exploration during training, potentially converging to suboptimal solutions. Moreover, its effectiveness on larger models is uncertain and its performance in tasks such as machine translation is not fully evaluated. DPO's lack of a reward model also constrains its ability to sample preference pairs from the optimal policy, therefore RSO was proposed to address this limitation. However, RSO also lacks studies in larger models, large-scale decoding samples, and other language generation tasks besides summarization and dialogue and non-human feedback. Importantly, DPO and RSO only contemplate pairwise preferences, which may not fully capture the richness of human ranking preferences.

On the other hand, PRO gain from self-bootstrapping is lower compared to adding high-quality outputs generated by other LMs (e.g. ChatGPT) to the preference ranking sequence and improvement with self-bootstrapping may not be as significant as adding high-quality external responses. Results also show that the performance also depends on the quality of the model giving the responses to increase the length of the rankings. PRO might also struggle with noisy or inconsistent preference feedback.

In this work, we aim to explore how can advanced architectures for training language models based on human preferences be effectively combined to improve their results and better align with human preferences while avoiding the drawbacks of traditional Reinforcement Learning from Human Feedback (RLHF) pipelines. In pursuit of this objective and to establish a robust theoretical foundation for this work, we carried out an extensive literature analysis regarding Reinforcement Learning from Human Feedback (RLHF), human preferences, and prevalent as well as alternative and innovative modeling approaches.

1.2 Expected Contributions

In line with the continuous efforts to explore alternatives for RLHF, we propose to:

1. Compare thoroughly the already developed models, mainly in the dialogue (single-turn and multi-turn), machine translation, and summarization tasks.

2. Examine how benchmarks for assessing model performance in these tasks align with human preferences for usefulness and safety.

3. Propose and implement a novel model that uses similar backbone models and aims to outperform the existing ones in the referred tasks, by combining the advantages of each model and aligning better with human preferences.

2 Background and Related Work

In recent years, the field of natural language processing (NLP) has undergone a revolutionary transformation with the introduction of Large Language Models (LLMs), particularly those pre-trained on massive datasets.

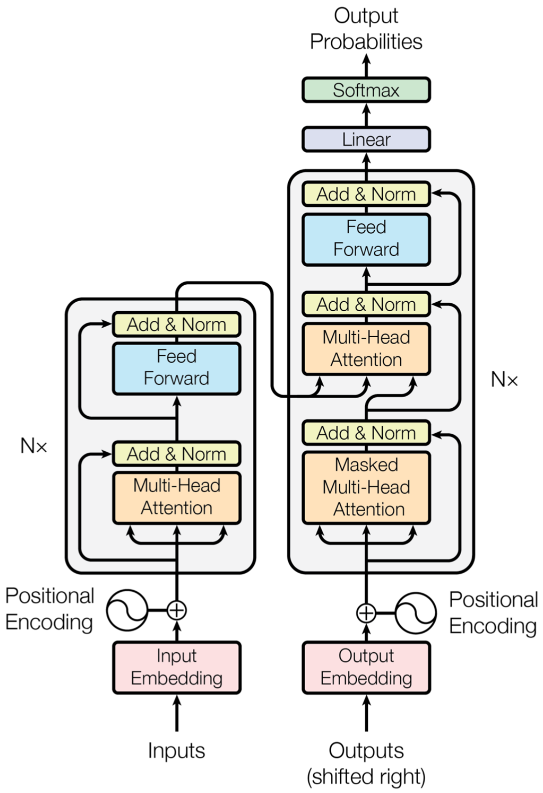


Figure 2: Transformer architecture. It consists of two main components: the encoder (left) and the decoder (right). This figure was taken from Vaswani et al. (2017).

2.1 Large Language Models (LLMs)

Language Models consist of artificial neural networks designed to understand and generate human-like language. This involves ascertaining the probability of a specific word within the model's vocabulary occurring, given all previous words. The subsequent predicted word in the sequence is typically determined by selecting the one with the highest probability. Formally, the probability of a sequence of words $y_1, y_2, ..., y_T$ given an input context x can be expressed using the chain rule of probability:

$$P(y_1, ..., y_T \mid x) = \prod_{t=1}^{T} P(y_t \mid x, y_1, ..., y_{t-1}) \tag{1}$$

A Large Language Model (LLM) is an advanced type of Language Model with the distinguishing feature that is their magnitude. They are usually constructed on enormous neural networks that contain billions of parameters, enabling them to learn language patterns, syntax, and semantics on an extensive scale. LLMs achieve this prowess through the utilization of massive datasets during training, coupled with powerful computational resources. They employ the Transformer architecture, depicted in Figure 2, which utilizes layers with Multi-head Attention mechanisms combined with Feed-Forward networks, enabling efficient capture of contextual relationships in sequential data.

Notable examples of LLMs include OpenAI's GPT models (e.g. GPT-3.5 and GPT-4, used in ChatGPT), Google's PaLM (used in Bard), and Meta's LLaMA.

Regarding the latest advancements in the field of developing new and smaller backbone models, Mistral 7B (Jiang et al., 2023) is engineered for superior performance and efficiency, leveraging grouped-query attention (GQA) for faster inference, coupled with sliding window attention (SWA) to effectively handle sequences of arbitrary length with a reduced inference cost. Mistral 7B outperforms Llama 2 13B across all evaluated benchmarks, and Llama 1 34B in reasoning, mathematics, and code generation. Similarly, Zephyr-7B (Tunstall et al., 2023), an aligned version of Mistral-7B, excels in intent alignment without human annotation. Leveraging AI Feedback data, distilled Fine-tuning (dSFT), and distilled Direct Preference Optimization (dDPO), it sets the state-of-the-art on chat benchmarks for 7B parameter models and requires no human annotation. In particular, results on MT-Bench (Zheng et al., 2023) show that Zephyr-7B surpasses Llama2-Chat-70B, the best open-access RLHF-based model.

2.2 Pre-training

Pre-training is the initial stage in the development and training process of LLMs, where the model learns from vast and diverse text data. During this phase, the model engages in self-supervised learning, capturing the contextual relationships and semantic nuances present in the data. This extensive pre-training on massive datasets enables LLMs to capture a broad spectrum of linguistic knowledge making them versatile and capable of handling various downstream natural language processing tasks effectively.

2.3 Fine-Tuning

Fine-tuning is an important process that adapts pre-trained models to specific tasks, leveraging the knowledge acquired during unsupervised pre-training. This approach enables efficient transfer learning, allowing models to specialize in particular domains or applications by exposing them to task-specific datasets and information. The benefits of fine-tuning include increased data efficiency, faster training cycles, and the incorporation of domain-specific knowledge. Ultimately, fine-tuning optimizes model performance, making it well-suited for diverse downstream tasks and facilitating the integration of advanced language understanding into various applications.

2.3.1 Supervised Fine-Tuning (SFT)

Supervised fine-tuning involves refining a pre-trained language model using labeled data tailored for a specific task. This process aims to improve the model's performance on targeted objectives by leveraging task-specific labeled datasets, containing input examples and their corresponding desired outputs.

Instruction fine-tuning (Wei et al., 2021) Instruction fine-tuning is a specialized technique to tailor large language models to perform specific tasks based on explicit instructions. While traditional supervised fine-tuning involves training a model on labeled task-specific data, instruction fine-tuning goes further by incorporating high-level instructions or demonstrations to guide the model's behavior. Through exposure to a variety of instructions, the model develops robust generalization skills, thereby improving its capacity to generate accurate responses that align with human-like instruction formats.

2.3.2 Distillation

An alternative method to supervised fine-tuning (SFT), referred to as distilled SFT (dSFT), leverages a teacher language model to generate instructions and responses, eliminating the need for an external dataset. The distilled SFT process, usually following the self-instruct protocol (Wang et al., 2022), involves using a set of seed prompts, denoted as $x_1^0, \ldots, x_n^0$ designed to cover diverse topical domains. The dataset is constructed through an iterative self-prompting approach, where the teacher responds to an instruction and refines it based on the response. For each x^0, the response y^0 is sampled from $\pi_T(\cdot \mid x^0)$, and then refined by a new instruction (using a prompt for refinement), $x_1 \sim \pi_T(\cdot \mid x^0, y^0)$. This iterative process results in a final dataset $C = \{(x_1, y_1), \ldots, (x_n, y_n)\}$. Distillation is achieved through supervised fine-tuning,

$$\pi_{\text{dSFT}} = \max_{\pi} \mathbb{E}_{(x,y)\sim C} \log \pi(y \mid x). \tag{2}$$

2.3.3 Fine-Tuning Challenges

The efficacy of the fine-tuning process is not immune to challenges, especially when deployed in intricate tasks such as dialogue, text summarization, or machine translation. The crux of the issue lies in the misalignment between the fine-tuning objective — usually centered around maximizing the log probability of human demonstrations — and the nuanced expectations, associated with generating high-quality outputs. Supervised fine-tuning only teaches LLMs about the best responses and cannot provide fine-grained comparisons to suboptimal ones. This misalignment becomes particularly conspicuous when the task requires a nuanced perception of context and the synthesis of information.

Although advantageous, the fine-tuning process also encounters the inherent difficulty of distinguishing between critical and less significant errors. Despite its benefits, the models face the challenge of dealing with the complexities of subtle language nuances, where the difference between a precisely crafted output and a less optimal one can be subtle yet impactful. The challenge intensifies as models, during fine-tuning, tend to assign probability mass to all human demonstrations, including those of low quality. This arises from the fact that the data used to train these models is generally scraped from the Internet, often containing noise, social biases, toxicity, and errors.

This inclination to treat all examples equally can compromise the model's ability to discern and prioritize the finer nuances necessary for generating high-quality and safe answers. For the output to be both harmless and truly helpful, a model should generate outputs considering both these parameters.

Moreover, the fine-tuning process contends with the challenge posed by distributional shifts during sampling. These shifts can introduce anomalies and discrepancies that were not present in the original training data, leading to a potential degradation in overall performance.

Addressing all these challenges is important for the practical applicability and real-world utility of large language models. It necessitates methodologies that move beyond traditional fine-tuning objectives, seeking to align model outputs more closely with human expectations and preferences.

2.4 Leveraging Human Feedback

2.4.1 Feedback Format

When considering human feedback, it is important to decide in what *format* this feedback must be collected (Fernandes et al., 2023). The choice of feedback format has implications on its expressivity, the ease of its collection, and how we can use it to improve the models. In particular, the complexity of the format is an important factor: simpler formats are often easier to collect and use as part of the training/decoding process but contain less information than more "complex" formats. Simpler formats might not be able to capture important information for improving the system. The choice of format also affects how easy it is for humans to provide feedback, its consistency/agreement, and the level of rationality of said feedback (Ghosal et al., 2023).

Numerical feedback, which takes an input and an output and returns a single score is one of the simplest feedback formats to collect and use (Kreutzer et al., 2018; Liu et al., 2018; Shi et al., 2020). Although easy to leverage, it suffers from some limitations: depending on the complexity of the generation task, reducing feedback to a single score might generally be a challenging and vaguely defined task for humans. This leads to a costly collection process, problems of subjectivity and variance, and difficulties in distinguishing between outputs of similar quality.

Alternatively, **Ranking-based** feedback involves humans ranking alternative outputs, providing a more nuanced understanding of model performance and often being easier to collect (Chaganty et al., 2018; Ziegler et al., 2019; Stiennon et al., 2020). **Natural Language** feedback, on the other hand, offers detailed insights by capturing specific issues or suggesting improvements, contributing to a richer understanding of model shortcomings, albeit with the challenge of increased complexity in its collection. Additionally, domain-specific feedback types, such as multi-aspect feedback or post-editions, further enhance the potential for refining model behavior (Li et al., 2016; Scheurer et al., 2022; Li et al., 2022). (See Figure 3)

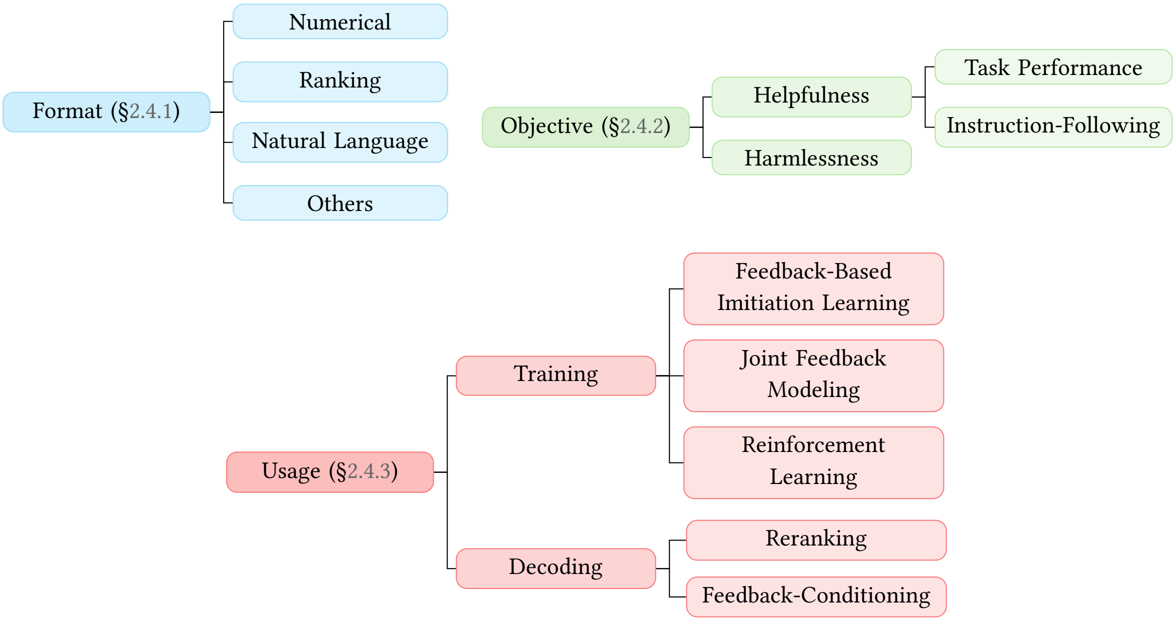


Figure 3: Taxonomy of methods that leverage human-feedback, adapted from an illustration in Fernandes et al. (2023).

2.4.2 Objective

Collecting human feedback is then essential to align the model's behavior with human preferences. Some works explicitly divide the problem of alignment of these models into improving its helpfulness and increasing its harmlessness (Bai et al., 2022). **Harmlessness** is comparatively easier as it primarily involves significant features such as adapting expression styles and maintaining politeness in most conversations. It focuses on mitigating undesirable and harmful outputs or norm-violating content by evaluating and avoiding text toxicity, adherence to safety objectives, addressing ethical concerns, and non-violation of predefined rules. **Helpfulness**, on the other hand, is often gauged through feedback related to task performance, such as the quality of translation or the relevance, consistency, and accuracy of summaries, with the assumption that improved task-related outputs contribute to overall system usefulness.

2.4.3 Usage of Human Feedback

To leverage human feedback and capture nuanced human preferences, training approaches like feedback-imitation learning and joint-feedback modeling appear to optimize model parameters using the collected human feedback information, as shown in Figure 3.

Feedback-based imitation learning involves optimizing language models through supervised learning using positively labeled generations and corresponding inputs or preferred dialogues (Fernandes et al., 2023; Li et al., 2016; Kreutzer et al., 2018; Glaese et al., 2022). Its main disadvantage lies in disregarding generations without positive feedback, potentially overlooking valuable information for the model's optimization.

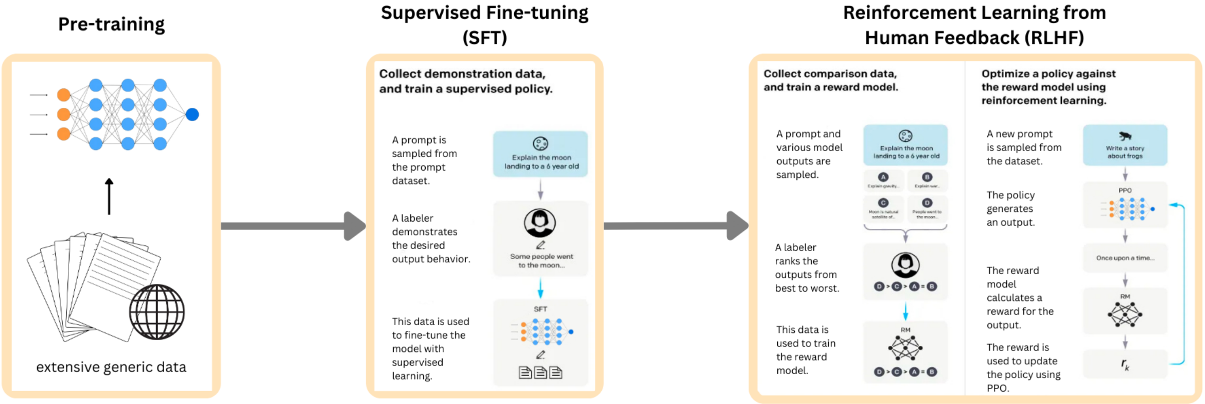
In contrast, **joint feedback modeling** incorporates all collected human feedback directly into model optimization, allowing for diverse feedback formats like natural language. Having $\mathcal{D}$ as the dataset of inputs x, generations y, and human feedback f, this can be achieved by minimizing the following loss of the form

$$\mathcal{L}_i(\theta) = -\log p_\theta(y_i, f_i \mid x_i) \tag{3}$$

Over all examples in $\mathcal{D}$. This loss function factors into

$$\mathcal{L}_i(\theta) = -\log p_\theta(f_i \mid y_i, x_i) + \log p_\theta(y_i \mid x_i) \tag{4}$$

The model can then be trained to predict feedback given to each generation (Li et al., 2016; Hancock et al., 2019), or predict both generations and corresponding human feedback (Xu et al., 2022; Thoppilan et al., 2022). This


Pre-training
extensive generic data
Supervised Fine-tuning (SFT)
Collect demonstration data, and train a supervised policy.
Explain the moon landing to a 6 year old
Some people went to the moon...
SFT
Reinforcement Learning from Human Feedback (RLHF)
Collect comparison data, and train a reward model.
Explain the moon landing to a 6 year old
A
Explain gravity...
B
Explain war...
C
Moon is natural satellite of...
D
People went to the moon...
D > C > A = B
RM
D > C > A = B
Optimize a policy against the reward model using reinforcement learning.
Write a story about frogs
PPO
Once upon a time...
RM
r_k

A prompt is sampled from the prompt dataset.

A labeler demonstrates the desired output behavior.

This data is used to fine-tune the model with supervised learning.

A prompt and various model outputs are sampled.

A labeler ranks the outputs from best to worst.

This data is used to train the reward model.

A new prompt is sampled from the dataset.

The policy generates an output.

The reward model calculates a reward for the output.

The reward is used to update the policy using PPO.

Figure 4: The lifecycle of Large Language Models (LLMs) with distinct stages, including pre-training, supervised fine-tuning (SFT), and Reinforcement Learning from Human Feedback (RLHF).

approach, however, faces the challenge of learning to accurately predict nuanced human feedback but handles problems such as increased complexity and potential training costs associated with diverse feedback formats.

2.5 Reinforcement Learning from Human Feedback (RLHF)

Reinforcement Learning from Human Feedback (RLHF) emerges as a paradigmatic shift to tackle the misalignment challenges prevalent in fine-tuning objectives. RLHF represents a strategic deviation from conventional supervised learning approaches, aiming to train models based on human preferences and evaluations. For instance, in the domain of text summarization, RLHF introduces a methodical approach, leveraging human feedback to navigate the inherent subjectivity of summarization tasks, where judging summary quality is complex without human judgments. The incorporation of human feedback plays an important role in tackling challenges associated with toxicity and harmful content generation. It also integrates user preferences, thereby ensuring ethical and user-aligned AI outputs.

The application of RLHF extends far beyond summarization, making an impact across various NLP domains. RLHF addresses a spectrum of tasks, including machine translation, dialog generation, image captioning, and sentiment generation. Notably, the scalability of language models in RLHF has increased, now reaching into the realm of hundreds of billions of parameters (Ye et al., 2023; Brown et al., 2020), reflecting a commitment to performance improvement and adaptability across the various NLP challenges.

The idea of RLHF is to align the language models with human preferences and social values by optimizing a reward function that reflects specific human preferences (e.g. moral, helpful, harmless). The process widely employed in recent studies involves a series of well-defined steps. The first stage consists of doing Supervised Fine-tuning (SFT) on the LLM: Labelers furnish the desired behavior's response with t tokens, denoted as $y = y_1, \ldots, y_t$, for a given input prompt, denoted as x. Subsequently, RLHF proceeds to fine-tune a pre-trained LLM using supervised learning (maximum likelihood) on this data, resulting in a model denoted as π_{SFT}:

$$\mathcal{L}_{\text{SFT}} = -\sum_{t} \log P_{\pi_{\text{SFT}}}(y_t \mid x, y_{1,\ldots,t-1}). \tag{5}$$

In the second phase, the SFT model is prompted with prompts x to generate pairs of responses $(y_1, y_2) \sim \pi_{\text{SFT}}(y \mid x)$. These pairs are then presented to human labelers, who express their preferences by indicating a favored answer as y_w, while the other response is denoted as y_l. Specifically, we use the notation $y_w \succ y_l \mid x$ where y_w and y_l denote the preferred and dispreferred answers amongst (y_1, y_2), respectively. The preferences are assumed to be generated by some latent reward model $r^*(x, y)$, which we do not have access to. To model human preferences, the Bradley-Terry (BT) model (Bradley and Terry, 1952) is one popular approach. Bradley-Terry seeks to assign higher scores to preferable responses in comparison to unfavorable ones when presented with the same prompt. The BT model defines the human preference distribution as follows:

$$p^*(y_1 \succ y_2 \mid x) = \frac{\exp(r^*(x, y_1))}{\exp(r^*(x, y_1)) + \exp(r^*(x, y_2))} \tag{6}$$

Subsequently, given a dataset providing comparative data $\mathcal{D} = \{x^{(i)}, y_w^{(i)}, y_l^{(i)}\}_{i=1}^N$, we can then parametrize a reward model $r_\theta(x, y)$ and estimate the parameters via maximum likelihood, minimizing its associated loss. Framing this objective as a binary classification to train the reward model, we get the following negative log-likelihood loss:

$$\mathcal{L}(r_\theta, \mathcal{D}) = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}}\Big[\log \sigma(r_\theta(x, y_w) - r_\theta(x, y_l))\Big], \tag{7}$$

where σ is the logistic sigmoid function, $r_\theta(x, y)$ denotes the scalar output of the reward model for prompt x and response y with parameters θ, and $\mathcal{D}$ represents the dataset of human judgments sampled from p^*.

In the context of Language Models (LMs), the network denoted as $r_\theta(x, y)$ is commonly initialized using the SFT model $\pi_{SFT}(y \mid x)$. This initialization involves the addition of a linear layer on top of the final transformer layer, generating a single scalar prediction for the reward value (Ziegler et al., 2019). To ensure a reward function with lower variance, prior works ensure that the rewards are normalized, hence $\mathbb{E}_{(x,y)\sim\mathcal{D}}\big[r_\theta(x, y)\big] = 0$ for all x.

In the subsequent step, RLHF utilizes the acquired reward function r_θ to provide feedback to the language model. The goal is to train a policy that generates higher-quality outputs as judged by humans (see Figure 5). Specifically, RLHF formulates the following optimization problem:

$$\max_{\pi_\theta} \mathbb{E}_{x\sim D, y\sim\pi_\theta(y \mid x)}\Big[r_\theta(x, y) - \beta D_{KL}(\pi_\theta(y \mid x) \parallel \pi_{\text{ref}}(y \mid x))\Big] \tag{8}$$

Here, β controls the deviation from the base reference policy π_{ref}, namely the initial STF model π_{SFT}. This is done to to maintain generation diversity and prevent from generating of only high-reward but meaningless answers. Moreover, in practice, the language model policy π_θ is also initialized to π_{SFT}.

Due to the discrete nature of language generations, this objective is non-differentiable and is typically optimized using reinforcement learning methods. Commonly utilized approaches are PPO (Schulman et al., 2017), REINFORCE (Williams, 1992), as well as other variants such as Natural Language Policy Optimization (NLPO) (Ramamurthy et al., 2023), amongst others. (See Figure 4)

Importantly, previous works (Ziegler et al., 2019; Stiennon et al., 2020; Bai et al., 2022) include in the reward function a term that penalizes the Kullback-Leibler (KL) divergence between the learned RL policy π_θ and the reference model π_{ref}. The complete reward function r can then be expressed as:

$$r(x, y) = r_\theta(x, y) - \beta \log\left[\frac{\pi_\theta(y \mid x)}{\pi_{\text{ref}}(y \mid x)}\right], \tag{9}$$

where β serves to weigh the KL divergence term. This KL term serves a dual purpose: acting as an entropy bonus and preventing the policy from producing outputs too dissimilar from those encountered by the reward model during training.

The standard approach has been to construct the reward function presented in Equation 9 and maximize it using Proximal Policy Optimization (PPO) (Schulman et al., 2017).

In conclusion, RLHF approaches offer two notable advantages compared to traditional Supervised Fine-tuning (SFT). Firstly, they leverage both positively and negatively labeled responses, leading to a more nuanced and informative learning process. Secondly, they can engage in self-bootstrapping to iteratively refine the model's responses, contributing to continuous improvement and adaptability.

2.5.1 Limitations and Challenges of RLHF

Despite the numerous advantages, critics raised multiple concerns regarding RLHF, highlighting what they perceive as its drawbacks. One notable concern is the heightened complexity in comparison to supervised learning

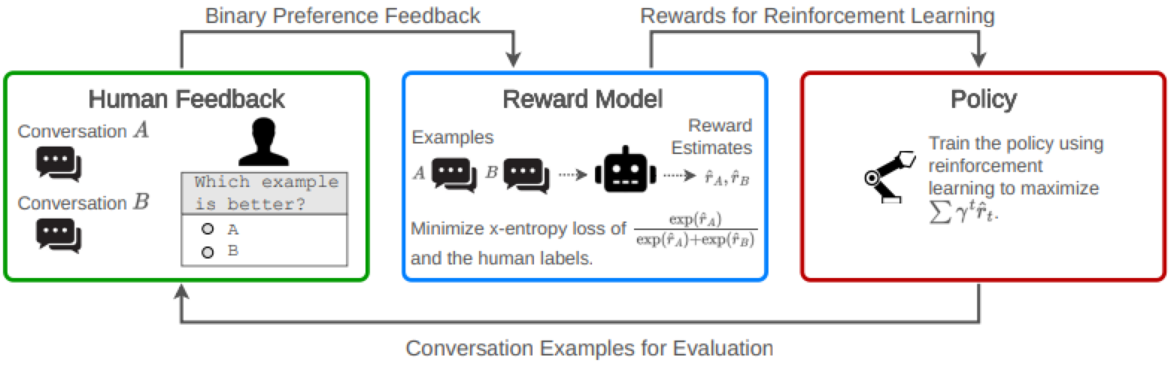

Binary Preference Feedback
Rewards for Reinforcement Learning
Human Feedback
Conversation A
Conversation B
Which example is better?
○ A
○ B
Reward Model
Examples
A
B
Reward Estimates
$\hat{r}_A, \hat{r}_B$
Minimize x-entropy loss of $\frac{\exp(\hat{r}_A)}{\exp(\hat{r}_A)+\exp(\hat{r}_B)}$ and the human labels.
Policy
Train the policy using reinforcement learning to maximize $\sum \gamma^t \hat{r}_t$.
Conversation Examples for Evaluation

Figure 5: An example of RLHF for finetuning chatbots with binary preference feedback, taken from Casper et al. (2023). Humans indicate which example between a pair they prefer. A reward model is trained using each example pair to provide rewards that reflect the human's preferences. Finally, the LLM policy is trained using the reward model and RL algorithms.

approaches. This augmented complexity is particularly notable, posing challenges that extend beyond conventional training paradigms. For instance, PPO learns in a trial-and-error fashion by interacting with the environment. In contrast to supervised learning, PPO training is difficult to implement, generally less stable, and computationally more demanding, resulting in a significantly slower training process. Furthermore, PPO exhibits heightened sensitivity to hyperparameters, demanding meticulous and precise tuning to achieve optimal outcomes.

To remove the PPO training from the RLHF process, **Reward rAnked FineTuning (RAFT)** (Dong et al., 2023) was proposed. RAFT is a novel alignment framework that iteratively alternates among three steps, 1) sample a batch of samples from the generative models; 2) use the reward function to score the samples from step 1 and filter them to get a filtered subset of high rewards; and 3) improve the generative models by fine-tuning on the filtered subset from step 2.

The proposed framework, RAFT, exhibits several advantages over the predominant PPO algorithm. It is based on SFT-like training and has improved stability and robustness compared to traditional RL-based PPO. Moreover, its limited set of hyperparameters simplifies the tuning and adjustment process. The decoupling of data generation and model fine-tuning reduces memory burden and provides flexibility in resource utilization. Additionally, RAFT can train various generative models if a reward model is available as the quality measure, including LLMs and diffusion models. This proposed framework prioritizes preferences over values and is resistant to reward scaling, leading the authors to hypothesize that RAFT is more robust against reward noise (variance and bias), which are known to be critical for the performance of PPO (Engstrom et al., 2020). Lastly, its preference-based objective is clear and interpretable given the filtered dataset, which helps in addressing the challenge of reward hacking[1] by monitoring the selected samples.

The RAFT learning process comprises three steps, iteratively repeating them until the reward converges. For each stage t + 1,

Step 1: Data Collection. First sample a batch of prompts $D_t = \{x_1^t, \dots, x_b^t\}$ from X and generate $y_1, \dots, y_K \sim p_{G_t}^{1/\lambda}(\cdot \mid w_t, x_i^t)$ for each $x_i^t \in D_t$, where the parameter λ is used to control the output diversity.

Step 2: Data Ranking. In this step, firstly use the reward model to compute $\{r(x, y_1), \dots, r(x, y_K)\}$ for each $x \in D_t$. Then, simply take $y := \arg\max_{y_j \in \{y_1, \dots, y_K\}} r(x, y_j)$ and go through all the b prompts and collect a subset $\mathcal{B}$ of size b.

[1] The reward model used in RLHF is far from perfect, and algorithms can exploit these imperfections to seek high rewards, leading to reward hacking.

Step 3: Model Fine-Tuning. Then, we simply fine-tune the current model on $\mathcal{B}$, and the next stage begins.

RAFT involves iterative learning from the induced best-of-K policy, which samples K responses and selects the highest-rewarded one as the final output. The best-of-K policy guides inference using the reward model but incurs high inference costs. On the other hand, RAFT iteratively learns from this policy, enhancing the model.

Furthermore, RAFT can also be used in offline human preference learning where the global instruction set is continually updated with the top-ranked instructions in each batch. This contiguously updates the global instruction set to improve training data quality at each step.

Another noteworthy drawback of RLHF centers on the need for additional training dedicated to refining reward models and value networks, adding an extra layer of complexity to the RLHF process. The predominant framework (Ouyang et al., 2022) requires loading multiple LLMs for the PPO training, including the model being trained, the reference model, the reward model, and the critic model. This imposes a heavy burden on the memory resource, potentially limiting the scalability and practical applicability of RLHF in real-world scenarios.

2.6 Alternative Approaches to RLHF

In addressing these limitations, recent studies suggest novel and varied offline methodologies. Recently, offline methods such as **DPO** (Direct Preference Optimization) (Rafailov et al., 2023), **PRO** (Preference-based Reinforcement Optimization) (Song et al., 2023), **RRHF** (Rank-based Reward Modeling with Human Feedback) (Yuan et al., 2023) and **SLiC** (Sequence Likelihood Calibration) (Liu et al., 2023) have emerged as attractive alternatives to RLHF, offering improvements in stability and scalability while maintaining competitive performance.

Direct Preference Optimization (DPO) (Rafailov et al., 2023) Motivated by challenges in applying reinforcement learning algorithms to large-scale tasks like language model fine-tuning, DPO is proposed as a straightforward method for policy optimization using preferences *directly*. This method introduces a new parameterization of the reward model in RLHF, allowing for the extraction of the optimal policy in closed form without an RL training loop. The approach leverages an analytical mapping from reward functions to optimal policies, transforming a loss function over rewards into a loss function over policies. By doing so, it avoids fitting an explicit reward model while still optimizing under existing models of human preferences, such as the Bradley-Terry model. Essentially, the policy network represents both the language model and the implicit reward.

The optimal solution to the KL-constrained reward maximization objective in Equation 8 is formulated differently, making it possible to express the reward function r in terms of its corresponding optimal policy π_r, the reference policy π_{ref}, and an unknown partition function $Z(x) = \sum_y \pi_{\text{ref}}(y \mid x) \exp\left(\frac{1}{\beta} r(x,y)\right)$:

$$r(x,y) = \beta \log \frac{\pi_r(y \mid x)}{\pi_{\text{ref}}(y \mid x)} + \beta \log Z(x). \tag{10}$$

We can apply this reparameterization to the ground-truth reward r^* and corresponding optimal model π^*. Fortunately, this approach recognizes that the Bradley-Terry model represented in Equation 6 relies solely on the difference of rewards between two responses, i.e. $p^*(y_1 \succ y_2 \mid x) = \sigma(r^*(x,y_1) - r^*(x,y_2))$. Therefore, by substituting the reparameterization in Equation 10 for $r^*(x,y)$ into the preference model Equation 6, the partition function cancels and the human preference probability can be expressed solely in terms of the optimal policy π_* and the reference policy π_{ref}. Hence, the optimal RLHF policy π^* under the Bradley-Terry model satisfies the preference model:

$$p^*(y_1 \succ y_2 \mid x) = \frac{1}{1 + \exp\left(\beta \log \frac{\pi^*(y_2|x)}{\pi_{\text{ref}}(y_2|x)} - \beta \log \frac{\pi^*(y_1|x)}{\pi_{\text{ref}}(y_1|x)}\right)} \tag{11}$$

Subsequently, with the probability of human preference data expressed in terms of the optimal policy instead of the reward model, it is possible to formulate a maximum likelihood objective for a parametrized policy π_θ. Analogous to the reward modeling approach in Equation 7, the policy objective becomes:

$$\mathcal{L}_{DPO}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x,y_w,y_l)\sim D}\left[\log \sigma \left(\beta \log \frac{\pi_\theta(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)} - \beta \log \frac{\pi_\theta(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)}\right)\right]. \tag{12}$$

By using this alternative parametrization, an implicit reward can be fitted in such a way that the optimal policy simply becomes π_θ.

Rank Responses to Human Feedback (RRHF) (Yuan et al., 2023) Prior to the training process, RRHF collects responses from different origins, which can include responses generated by the model itself, ChatGPT, GPT-4, as well as pre-existing human-written responses of varying quality. Therefore, during training, there are k different responses y_i of x sampled using policy ρ_i, $1 \leq i \leq k$. Importantly, sampling with policy ρ_i is not restricted and can be any of the sources mentioned earlier. This sampling method leverages any existing good or bad responses for model alignment with humans, while PPO learns solely from its own model-generated samples.

The reward function assigns a score to each y_i, with $R(x, y_i) = r_i$. In order to match the scores $\{r_i\}_k$, the learned model π is used to give scores p_i for each y_i. The scores are assigned based on the quality of the response, with better responses being assigned larger scores and worse responses being assigned smaller scores. The total loss is then defined as the unweighted sum of a ranking loss and a cross-entropy loss similar to SFT.

$$\begin{aligned}\mathcal{L} = \mathcal{L}_{\text{rank}} + \mathcal{L}_{\text{ft}} &= \sum_{r_i < r_j} \max(0, p_i - p_j) - ||y_{i'}||\; p_{i'} \\ &= \sum_{r_i < r_j} \max\left(0, \pi_\theta(y_i \mid x) - \pi_\theta(y_j \mid x)\right) - ||y_{i'}||\; \pi_\theta(y_{i'} \mid x)\end{aligned} \tag{13}$$

where $i' = \arg\max_i r_i$, requiring the model to learn the response with the highest reward r_i.

In summary, RRHF requires only 1 to 2 models during tuning and can effectively align language models with human preferences, without the need for complex hyperparameter tuning. Furthermore, RRHF can be viewed as an extension of SFT and reward model training. While being simpler than PPO with respect to coding, model counts, and hyperparameters, it demonstrates comparable alignment performance.

Sequence Likelihood Calibration (SLiC) (Zhao et al., 2023) Similarly, SLiC effectively leverages feedback data from another model (off-policy), making it unnecessary to collect costly new feedback data. For sampled and labeled preference pairs, it combines a rank calibration loss and a cross-entropy regularization loss, taking advantage of their simplicity and natural fit to pairwise human feedback data. The loss function is then defined as:

$$\mathcal{L}(\theta) = \max\left(0, \delta - \log \pi_\theta(y_w \mid x) + \log \pi_\theta(y_l \mid x)\right) - \lambda \log \pi_\theta(y_{\text{ref}} \mid x) \tag{14}$$

where δ is a positive margin of the ranking loss, and π_θ is the learnable conditional probability function by the language model. The second term is the cross-entropy loss, with y_{ref} representing some target sequence and λ the regularization weight. SLiC intuitively applies a penalty to the model when the ratio $\frac{\pi_\theta(y_w \mid x)}{\pi_\theta(y_l \mid x)} < \exp(\delta)$, encouraging a substantial likelihood ratio between positive and negative outputs. Instead of directly fitting on human preference data, SLiC proposed refining its loss function using sequence pairs sampled from the SFT policy and labeling them using a pairwise reward-ranking model.

Although RRHF and SLiC approaches appear as scalable alternatives to PPO, they lack theoretical understanding. On the other hand, DPO utilizes human preference data from other policies and lacks explicit study on the effects of sampling. Sampling human preference pairs directly from π^* is highly challenging in reality, given the mismatch between the sampling distribution and π^*. Furthermore, both these approaches might fail to capture global differences corresponding to human preference in the long rankings.

Statistical Rejection Sampling Optimization (RSO) (Liu et al., 2023) RSO is introduced to address DPO's containment in sampling preference pairs from the optimal policy, given the lack of a reward model, and SLiC restriction to sampling preference pairs only from the SFT policy. This novel approach aims to source preference data from the target optimal policy using statistical rejection sampling, enabling a more accurate estimation of the optimal policy.

Statistical rejection sampling (Neal, 2003) is an efficient statistical technique to generate observations from a distribution. If we want to generate a distribution of density π_{r_θ}, we can use π_{sft} as the proposal distribution. Then we can follow the following sequence of steps:

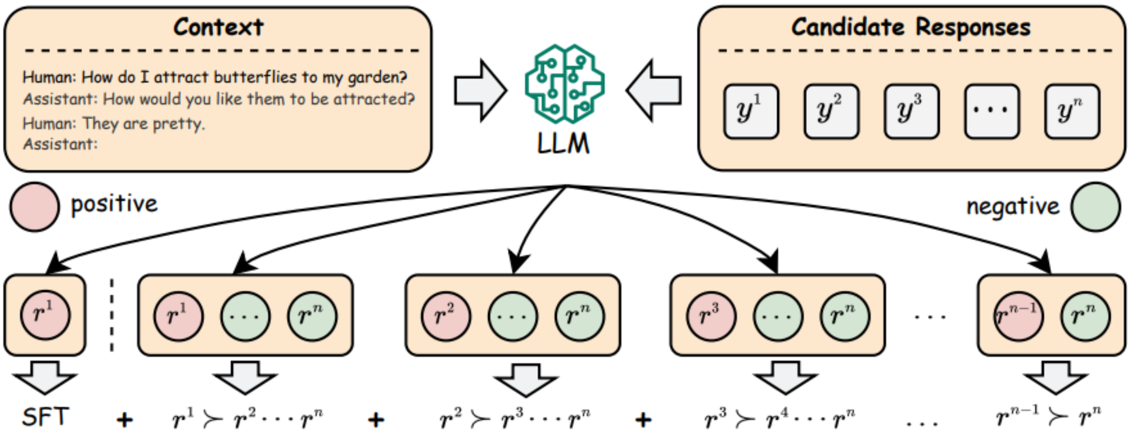


Figure 6: The pipeline of PRO for Human Feedback Alignment learning as depicted in (Song et al., 2023) which are optimized by Equation 17.

Step 1: Generate $y \sim \pi_{\text{SFT}}(y \mid x)$ and $u \sim U[0, 1]$.

Step 2: Let $M = \min\{m \mid m\pi_{\text{SFT}}(y \mid x) \geq \pi_{r_\theta}(y \mid x) \text{ for all } y\}$. If $u < \frac{\pi_{r_\theta}(y \mid x)}{M\pi_{\text{SFT}}(y \mid x)}$, then accept y. Otherwise, reject y and redo the sampling.

where $\pi_{r_\theta}(y \mid x)$ is closer to $\pi^*(y \mid x)$ than π_{unk}, which denotes some mixed unknown policies, and π_{SFT}. This mechanism introduces a bias toward accepting sequences that have a higher probability of being generated by the π_{r_θ} compared to the initial SFT model. This bias is more pronounced in regions where the $\pi_{r_\theta}(y \mid x)$ assigns higher rewards. Therefore, the sampling process becomes more focused on responses at the higher-reward regions, drawing samples from the optimal policy.

Assuming there are only two possible responses, denoted as y_1 and y_2, for a given prompt x in the LLM response space, the reward model should prioritize y_1 over y_2. However, if we consider a larger response space with n possible responses, denoted as $\{y_i\}$, and a human-annotated order of $y_1, \ldots, y_n$ where $y_1 \succ y_2 \succ \ldots \succ y_n$, we can define a partial order between y_1 and all the candidates behind it as $y_{1,2:n} = y_1 \succ y_2, \ldots, y_n$.

Preference Ranking Optimization (PRO) (Song et al., 2023) PRO was proposed as an alternative to PPO, extending Bradley-Terry's pairwise comparison to include comparisons within preference rankings of arbitrary lengths. Equation 6 presents the Bradley-Terry model, designed to convey the understanding that $y_1 \succ y_2$ through score comparison. When given a prompt x and only two responses, y_1 and y_2, in the LLM response space, the reward model should prefer y_1 over y_2. If the the LLM response space is extended to include n possible responses $\{y_i\}$, and the human-annotated order $y_1, \ldots, y_n$ is $y_1 \succ y_2 \succ \ldots \succ y_n$, the partial order between y_1 and all candidates behind it is defined as $y_{1,2:n} = y_1 \succ \{y_2, \ldots, y_n\}$. The objective of Bradley-Terry can then be expressed as:

$$P(y_{1,2:n} \mid x) = \frac{\exp(r(x, y_1))}{\sum_{i=1}^{n} \exp(r(x, y_i))} \tag{15}$$

To fully leverage the preference rankings $y_1, \ldots, y_n$ and not disregard the $n - 2$ valuable rankings such as $y_2 \succ y_3, \ldots, y_n$ and $y_{n-1} \succ y_n$, they propose an extension to Equation 15 as follows:

$$P(y_{1,\ldots,n} \mid x) = \prod_{k=1}^{n-1} P(y_{k,k+1:n} \mid x) = \prod_{k=1}^{n-1} \frac{\exp(r(x, y_k))}{\sum_{i=k}^{n} \exp(r(x, y_i))} \tag{16}$$

This extension enforces the desired ranking indicated by $y_1 \succ y_2 \succ \ldots \succ y_n$, employing Equation 15 iteratively. It starts with the first response, treating the remaining responses as negatives. Then, it removes the current response and moves to the next, repeating this process until no responses are left. Remarkably, this objective closely aligns with the overarching goal of Human alignment, which is the task of selecting desired responses from the extensive response space of LLMs. Essentially, as $n \longrightarrow \infty$, this extension can exhaustively explore all possible responses, annotating y^1 as the most desired response, thus achieving perfect alignment with humans. Figure 6 demonstrates the pipeline of the PRO algorithm.

The reward function is then defined using the desired π_{PRO}. The LLM π_{PRO} calculates the score for response y_k by multiplying the probabilities of tokens generated by π_{PRO} itself. Optimizing Equation 16 enables π_{PRO} to consistently generate the most preferred response with a higher output probability from a candidate set, aligning with human preferences. The overall optimization objective can be defined as follows:

$$\mathcal{L}(y_{1,\ldots,n} \mid x) = \mathcal{L}_{\text{PRO}} + \beta \mathcal{L}_{\text{SFT}} \tag{17}$$

where $\mathcal{L}_{\text{SFT}}$ is the negative log-likelihood loss of the top 1 candidate, and β is the hyper-parameter that balances text quality and human preference. L_{PRO} is defined as:

$$\mathcal{L}_{\text{PRO}} = -\sum_{k=1}^{n-1} \log \frac{\exp(r_{\pi_{\text{PRO}}}(x, y_k))}{\sum_{i=k}^{n} \exp(r_{\pi_{\text{PRO}}}(x, y_i))} \tag{18}$$

PRO and RLHF share the goal of human alignment, but PRO achieves this through differentiable ranking scores, avoiding RL-associated drawbacks. Additionally, PRO aims for high-quality model outputs with a differentiable alignment objective, allowing efficient single-stage training and multi-task learning.

2.7 Summary of the Related Work

In recent advancements in the field of training large language models (LLMs), such as GPT-3 (Brown et al., 2020), GPT-4 (OpenAI, 2023), Google's PaLM (Chowdhery et al., 2023; Anil et al., 2023), and Meta's LLaMA (Touvron et al., 2023a), there have been significant improvements in the models' ability to perform tasks without prior training or with minimal examples (zero-shot or few-shot scenarios) (Radford et al., 2019). This progress aligns with the broader evolution from discriminative models (e.g. BERT (Devlin et al., 2018)) to generative models (e.g., GPT-3 (Brown et al., 2020)) as part of the generative foundation model approach (Bommasani et al., 2021). Generative foundation models, pre-trained on extensive data and adaptable to diverse downstream tasks, have reshaped the landscape of natural language processing (NLP) and exhibited emergent capabilities in complex reasoning tasks. Despite their success, generative foundation models grapple with implicit biases, often resulting in inaccurate or toxic outcomes.

RLHF (Ziegler et al., 2019) focuses on maximizing rewards through interaction with a reward model using reinforcement learning algorithms like PPO (Schulman et al., 2017). While RLHF has demonstrated success, the process of fine-tuning large language models using reinforcement learning remains a challenge due to its instability, reward hacking, and scalability (Zhao et al., 2023; Rafailov et al., 2023).

Addressing these challenges, recent works explore alternatives to PPO (Yuan et al., 2023; Donato et al., 2022) and the introducement of robust RL-free approaches to optimize for human preference feedback (Zhao et al., 2023; Rafailov et al., 2023; Liu et al., 2023; Song et al., 2023; Yuan et al., 2023; Dong et al., 2023). These latter alternative approaches aim to sidestep the practical obstacles associated with RL-based fine-tuning, providing promising avenues for enhancing language model performance without the complexities and potential pitfalls of Reinforcement Learning.

Particularly, some methods have been proposed for optimizing relative preferences without relying on RL (Zhao et al., 2023; Yuan et al., 2023; Rafailov et al., 2023). These methods optimize the model's compatibility with preference datasets under models like the BT model, focusing on human or model-ranked data pairs. SLiC (Zhao et al., 2023) advocates for training a pairwise reward-ranking model, labeling pairs generated from supervised fine-tuning (SFT) using the reward model, and subsequently training the model using a contrastive calibration loss (Zhao et al., 2022) and a regularization fine-tuning loss. On the other hand, RRHF (Yuan et al., 2023) assumes access to a list of responses and their reward values to the same input, utilizing a zero-margin likelihood contrastive loss. Although RRHF and SLiC are shown to be a scalable alternative to PPO, they lack theoretical understanding. DPO (Rafailov et al., 2023) introduces a new parameterization of the reward model in RLHF that enables the extraction of the corresponding optimal policy in closed form, solving the standard RLHF problem with only a simple classification loss. Based on the Bradley-Terry model (Bradley and Terry, 1952), DPO proposes a maximum likelihood estimator (MLE) to fit on human preference data *directly*. It establishes a theoretical foundation that connects a language model with a preference model as a density estimation problem from the labeled response pairs.

To further enhance the performance of these models, additional works (Liu et al., 2023) propose an approach that seeks to integrate the methodologies of SLiC and DPO, considering the choice of loss perspective and incorporating theoretical foundations. The proposed method also introduces enhancements in the collection of preference pairs through statistical rejection sampling (Neal, 2003), a technique for generating samples from a target distribution using a proposal distribution. RLHF works (Bai et al., 2022; Stiennon et al., 2020; Touvron et al., 2023b) usually refer to "rejection sampling" as best-of-N or top-k-over-N algorithm, where they sample a batch of N completions from a language model policy and then evaluate them across a reward model, returning the best one or the best k ones. RSO shows that the already existing approach is a special case of their algorithm. RSO demonstrates that its algorithm constitutes a specialized instance of this existing approach.

With the same objectives in mind, the PRO (Song et al., 2023) approach proposes an extended version of the reward model training objective proposed in (Ziegler et al., 2019), to further finetune LLMs to align with human preference. PRO extends the pairwise Bradley-Terry (Bradley and Terry, 1952) model to accommodate arbitrary-length preference rankings. Given instruction x and a set of responses with human preference, it fully leverages the output's ranking order $y_1 \succ y_2 \succ \ldots \succ y_n$.

3 Proposed Solution

Based on all the information gathered from the literature analysis and taking into account the work objectives and expected contributions, in this section we present our proposed approach.

3.1 Approach

As described in Section 2.6, prominent RL-free models have a competitive performance to RLHF and overcome a lot of its drawbacks, namely stability, complexity, computational weight, and significant hyperparameter tuning.

With this insight, our technical approach involves a meticulous comparison of existing models, particularly in the domains of dialogue (single-turn and multi-turn), machine translation, and summarization tasks. Starting with backbone models pre-trained to autoregressively predict the next token in a large text corpus, we fine-tune them via supervised fine-tuning (SFT) using supervised learning (maximum likelihood). We plan to use these supervised models to sample initial outputs for collecting comparisons, to initialize our policy, and as baselines for evaluation.

Following that, we intend to implement Direct Preference Optimization (DPO), Statistical Rejection Sampling Optimization (RSO), and Preference Ranking Optimization (PRO), and fine-tune hyperparameters (e.g. KL) to enhance results across the considered tasks. Furthermore, thoroughly investigate the impact of affordable preference ranking using efficient non-human annotators, penalty of positive examples towards negative examples, and self-bootstrapping (Song et al., 2023).

Additionally, we propose and implement a novel model utilizing similar backbone models, with the objective of surpassing existing models by leveraging the strengths of each and attaining superior performance to robust baseline RLHF models. For instance, by incorporating Bradley-Terry model's extension from PRO in Equation 15 with the reward function from DPO in Equation 10 we can deduce that the optimal RLHF policy π^* satisfies the preference model:

$$P^*(y_1 \succ \{y_2, \cdots, y_n\} \mid x) = \frac{1}{1 + \sum_{i=2}^{n} \exp\left(\beta \log \frac{\pi^*(y_i \mid x)}{\pi_{\text{ref}}(y_i \mid x)} - \beta \log \frac{\pi^*(y_1 \mid x)}{\pi_{\text{ref}}(y_1 \mid x)}\right)} \tag{19}$$

To fully leverage the ranking and not disregard the $n-2$ valuable rankings, as it is done in Equation 16, we get:

$$\begin{aligned} P^*(y_1 \succ \cdots \succ y_n \mid x) &= \prod_{k=1}^{n-1} P^*(y_k \succ \{y_{k+1}, \ldots, y_n\}) \\ &= \prod_{k=1}^{n-1} \frac{1}{1 + \sum_{i=k+1}^{n} \exp\left(\beta \log \frac{\pi^*(y_i \mid x)}{\pi_{\mathrm{ref}}(y_i \mid x)} - \beta \log \frac{\pi^*(y_1 \mid x)}{\pi_{\mathrm{ref}}(y_1 \mid x)}\right)} \end{aligned} \tag{20}$$

We can then formulate a maximum likelihood objective for a parametrized policy π_θ. This new approach combines both DPO's new parameterization of the reward model that enables the extraction of the corresponding optimal policy in closed form and PRO's extension of the pairwise Bradley-Terry model (Bradley and Terry, 1952) to accommodate arbitrary-length preference rankings, which closely aligns with the overarching goal of human alignment.

3.2 Architecture

As backbone models, we propose **LLaMA-7B** (Touvron et al., 2023a), which has become a widespread test field for LLM research. We intend to fine-tune it with multiple alternatives to RL algorithms built on Huggingface.Library. This backbone model was used in novel approaches such as DPO, PRO, RAFT, and RRHF. We also intend to investigate alternative models, particularly: **Mistral 7B** (Jiang et al., 2023) given its competitive performance against Llama 2 13B across all assessed benchmarks and its superiority to Llama 1 34B in reasoning, mathematics, and code generation tasks; and **Zephyr-7B** (Tunstall et al., 2023), an aligned version of Mistral-7B, which surpasses Llama2-Chat-70B, the best open-access RLHF-based model, on many NLP benchmarks.

Similarly to related work (Tunstall et al., 2023; Rafailov et al., 2023), we intend to use the Transformer Reinforcement Learning (TRL) library for fine-tuning (Leandro von Werra et al., 2020), in conjunction with DeepSpeed ZeRO3 (Rajbhandari et al., 2020) and FlashAttention-2 (Dao, 2023) to optimize memory and improve training speed.

3.3 Datasets

We intend to focus on tree datasets that have previously been shown to produce strong models:

- **UltraChat** (Ding et al., 2023) is a self-refinement dataset consisting of 1.47M multi-turn dialogues generated by GPT-3.5-TURBO over 30 topics and 20 different types of text material. To address problems like incorrect capitalization and undesired prefaces in responses (e.g., *"I don't have personal experiences"*), while emphasizing helpfulness and removing undesirable responses (Tunstall et al., 2023), we plan to utilize the pre-processed dataset available on the Hugging Face Hub[2].

- **UltraFeedback** (Cui et al., 2023) consists of 64k prompts, each of which has four LLM responses that are rated by GPT-4 according to criteria like instruction-following, honesty, and helpfulness. We also plan to use the avalaible[2] pre-processed dataset that contains binary preferences (Tunstall et al., 2023).

- **Reddit TL;DR**, a summarization dataset (Völske et al., 2017), along with human preferences gathered by (Stiennon et al., 2020). The original dataset contains ~3 million posts from `reddit.com` across a variety of topics (subreddits), as well as summaries of the posts written by the original poster (TL;DRs).

4 Evaluation

In order to analyze the effectiveness of each algorithm, we intend to first compare the results between multiple approaches and reference RLHF and RL-free baselines. Subsequently, we assess our newly proposed model in comparison to the best previous approaches.

[2] https://huggingface.co/collections/HuggingFaceH4/zephyr-7b-6538c6d6d5ddd1cbb1744a66

Powerful LLMs such as GPT-4 and Claude have been used as evaluators to judge model responses by scoring model outputs or ranking responses in a pairwise setting. These automatic approaches emerged as a scalable approach for rapidly assessing human preferences (Song et al., 2023). Therefore, we intend to evaluate algorithms with their win rate against a baseline policy, using GPT-4 (OpenAI, 2023) as a proxy for human evaluation of response helpfulness and summary quality, and response helpfulness in the dialogue and summarization settings, respectively. For dialogue, we use the preferred response in the test dataset as the baseline; as for summarization, we use the preferred summaries in the test set as the baseline.

For the evaluation of single-turn and multi-turn chat, we plan to assess a model's ability to follow instructions and respond to challenging prompts across diverse domains using the following benchmarks:

- **MT-Bench** (Zheng et al., 2023) is a multi-turn benchmark comprising 160 questions spanning eight knowledge domains. Models are required to answer an initial question and provide a second response to a predefined follow-up question. GPT-4 rates each model's response on a scale from 1-10, with the final score determined by the mean over the two turns.

- **AlpacaEval** (Dubois et al., 2023) is a single-turn benchmark involving the generation of responses to 805 questions on various topics, predominantly focusing on helpfulness. GPT-4 scores the models, and the ultimate metric is the pairwise win rate against a baseline model (*text-davinci-003*).

We can also evaluate our models on the **Open LLM Leaderboard**[3] (Beeching et al., 2023), which measures the performance of LMs across different multiclass classification tasks, as it provides a useful signal to validate whether fine-tuning has introduced regressions on the base model's reasoning and truthfulness capabilities.

To assess summary quality, we discard some existing automatic metrics for evaluation such as ROUGE, as they have faced criticism for demonstrating a limited correlation with human judgments. Optimizing against ROUGE using a simple optimization scheme doesn't consistently increase quality and peaks both sooner and at a substantially lower quality rate than optimization against their reward models (Stiennon et al., 2020). However, we can utilize BLEU (BiLingual Evaluation Understudy) to assess the text quality, to compare inference results with preferred responses in test sets.

Similarly to recent works (Rafailov et al., 2023), we plan to evaluate other existing approaches to training language models to adhere to human preferences, namely zero-shot prompting with **GPT-J** in the summarization task and 2-shot prompting with **Pythia-2.8B** in the dialogue task. In addition, we plan to evaluate the SFT model as well as **Preferred-FT**, which is a model fine-tuned with supervised learning on the chosen completion y_w from either the SFT model (in summarization) or a generic LM (in dialogue). Finally, we also intend to consider **PPO** using a reward function learned from the preference data.

5 Work Schedule

To elaborate on the work proposed in this paper and ensure its successful execution, we plan to follow the following schedule (Figure 7):

[3] https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard

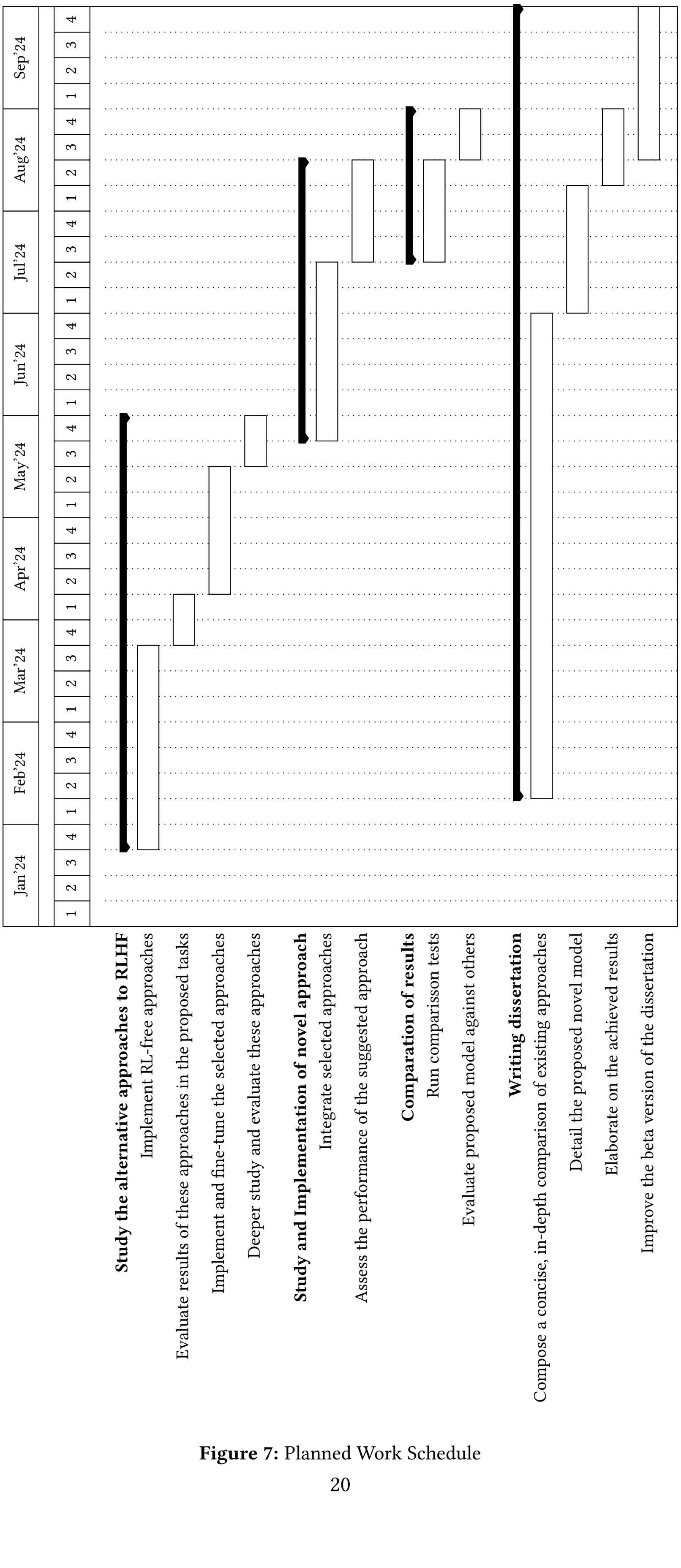


Figure 7: Planned Work Schedule

Bibliography

R. Anil, A. M. Dai, O. Firat, M. Johnson, D. Lepikhin, A. Passos, S. Shakeri, E. Taropa, P. Bailey, Z. Chen, et al. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*, 2023.

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.

E. Beeching, C. Fourrier, N. Habib, S. Han, N. Lambert, N. Rajani, O. Sanseviero, L. Tunstall, and T. Wolf. Open llm leaderboard. `https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard`, 2023.

R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

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. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

S. Casper, X. Davies, C. Shi, T. K. Gilbert, J. Scheurer, J. Rando, R. Freedman, T. Korbak, D. Lindner, P. Freire, et al. Open problems and fundamental limitations of reinforcement learning from human feedback. *arXiv preprint arXiv:2307.15217*, 2023.

A. T. Chaganty, S. Mussman, and P. Liang. The price of debiasing automatic metrics in natural language evaluation. *arXiv preprint arXiv:1807.02202*, 2018.

A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann, et al. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113, 2023.

H. W. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, Y. Li, X. Wang, M. Dehghani, S. Brahma, et al. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*, 2022.

G. Cui, L. Yuan, N. Ding, G. Yao, W. Zhu, Y. Ni, G. Xie, Z. Liu, and M. Sun. Ultrafeedback: Boosting language models with high-quality feedback. *arXiv preprint arXiv:2310.01377*, 2023.

T. Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. *arXiv preprint arXiv:2307.08691*, 2023.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

N. Ding, Y. Chen, B. Xu, Y. Qin, Z. Zheng, S. Hu, Z. Liu, M. Sun, and B. Zhou. Enhancing chat language models by scaling high-quality instructional conversations. *arXiv preprint arXiv:2305.14233*, 2023.

D. Donato, L. Yu, W. Ling, and C. Dyer. Mad for robust reinforcement learning in machine translation. *arXiv preprint arXiv:2207.08583*, 2022.

H. Dong, W. Xiong, D. Goyal, R. Pan, S. Diao, J. Zhang, K. Shum, and T. Zhang. Raft: Reward ranked finetuning for generative foundation model alignment. *arXiv preprint arXiv:2304.06767*, 2023.

Y. Dubois, X. Li, R. Taori, T. Zhang, I. Gulrajani, J. Ba, C. Guestrin, P. Liang, and T. B. Hashimoto. Alpacafarm: A simulation framework for methods that learn from human feedback. *arXiv preprint arXiv:2305.14387*, 2023.

L. Engstrom, A. Ilyas, S. Santurkar, D. Tsipras, F. Janoos, L. Rudolph, and A. Madry. Implementation matters in deep policy gradients: A case study on ppo and trpo. *arXiv preprint arXiv:2005.12729*, 2020.

P. Fernandes, A. Madaan, E. Liu, A. Farinhas, P. H. Martins, A. Bertsch, J. G. de Souza, S. Zhou, T. Wu, G. Neubig, et al. Bridging the gap: A survey on integrating (human) feedback for natural language generation. *arXiv preprint arXiv:2305.00955*, 2023.

G. R. Ghosal, M. Zurek, D. S. Brown, and A. D. Dragan. The effect of modeling human rationality level on learning rewards from multiple feedback types. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 5983–5992, 2023.

A. Glaese, N. McAleese, M. Trębacz, J. Aslanides, V. Firoiu, T. Ewalds, M. Rauh, L. Weidinger, M. Chadwick, P. Thacker, et al. Improving alignment of dialogue agents via targeted human judgements. *arXiv preprint arXiv:2209.14375*, 2022.

B. Hancock, A. Bordes, P.-E. Mazare, and J. Weston. Learning from dialogue after deployment: Feed yourself, chatbot! In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3667–3684, 2019.

A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. d. l. Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

J. Kreutzer, S. Khadivi, E. Matusov, and S. Riezler. Can neural machine translation be improved with user feedback? *arXiv preprint arXiv:1804.05958*, 2018.

Y. B. Leandro von Werra, L. Tunstall, E. Beeching, T. Thrush, N. Lambert, and S. Huang. Trl: Transformer reinforcement learning. 2020.

J. Li, A. H. Miller, S. Chopra, M. Ranzato, and J. Weston. Dialogue learning with human-in-the-loop. *arXiv preprint arXiv:1611.09823*, 2016.

Z. Li, P. Sharma, X. H. Lu, J. C. Cheung, and S. Reddy. Using interactive feedback to improve the accuracy and explainability of question answering systems post-deployment. *arXiv preprint arXiv:2204.03025*, 2022.

B. Liu, G. Tur, D. Hakkani-Tur, P. Shah, and L. Heck. Dialogue learning with human teaching and feedback in end-to-end trainable task-oriented dialogue systems. *arXiv preprint arXiv:1804.06512*, 2018.

T. Liu, Y. Zhao, R. Joshi, M. Khalman, M. Saleh, P. J. Liu, and J. Liu. Statistical rejection sampling improves preference optimization. *arXiv preprint arXiv:2309.06657*, 2023.

R. M. Neal. Slice sampling. *The annals of statistics*, 31(3):705–767, 2003.

OpenAI. Gpt-4 technical report, 2023.

L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

R. Rafailov, A. Sharma, E. Mitchell, S. Ermon, C. D. Manning, and C. Finn. Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint arXiv:2305.18290*, 2023.

S. Rajbhandari, J. Rasley, O. Ruwase, and Y. He. Zero: Memory optimizations toward training trillion parameter models. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–16. IEEE, 2020.

R. Ramamurthy, P. Ammanabrolu, K. Brantley, J. Hessel, R. Sifa, C. Bauckhage, H. Hajishirzi, and Y. Choi. Is reinforcement learning (not) for natural language processing: Benchmarks, baselines, and building blocks for natural language policy optimization. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=8aHzds2uUyB.

J. Scheurer, J. A. Campos, J. S. Chan, A. Chen, K. Cho, and E. Perez. Training language models with natural language feedback. *arXiv preprint arXiv:2204.14146*, 8, 2022.

J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

W. Shi, Y. Li, S. Sahay, and Z. Yu. Refine and imitate: Reducing repetition and inconsistency in persuasion dialogues via reinforcement learning and human demonstration. *arXiv preprint arXiv:2012.15375*, 2020.

F. Song, B. Yu, M. Li, H. Yu, F. Huang, Y. Li, and H. Wang. Preference ranking optimization for human alignment. *arXiv preprint arXiv:2306.17492*, 2023.

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. Thoppilan, D. De Freitas, J. Hall, N. Shazeer, A. Kulshreshtha, H.-T. Cheng, A. Jin, T. Bos, L. Baker, Y. Du, et al. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*, 2022.

H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023a.

H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023b.

L. Tunstall, E. Beeching, N. Lambert, N. Rajani, K. Rasul, Y. Belkada, S. Huang, L. von Werra, C. Fourrier, N. Habib, et al. Zephyr: Direct distillation of lm alignment. *arXiv preprint arXiv:2310.16944*, 2023.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

M. Völske, M. Potthast, S. Syed, and B. Stein. Tl;dr: Mining reddit to learn automatic summarization. In *Proceedings of the Workshop on New Frontiers in Summarization*, pages 59–63, 2017.

Y. Wang, Y. Kordi, S. Mishra, A. Liu, N. A. Smith, D. Khashabi, and H. Hajishirzi. Self-instruct: Aligning language model with self generated instructions. *arXiv preprint arXiv:2212.10560*, 2022.

J. Wei, M. Bosma, V. Y. Zhao, K. Guu, A. W. Yu, B. Lester, N. Du, A. M. Dai, and Q. V. Le. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*, 2021.

R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8:229–256, 1992.

J. Xu, M. Ung, M. Komeili, K. Arora, Y.-L. Boureau, and J. Weston. Learning new skills after deployment: Improving open-domain internet-driven dialogue with human feedback. *arXiv preprint arXiv:2208.03270*, 2022.

J. Ye, X. Chen, N. Xu, C. Zu, Z. Shao, S. Liu, Y. Cui, Z. Zhou, C. Gong, Y. Shen, et al. A comprehensive capability analysis of gpt-3 and gpt-3.5 series models. *arXiv preprint arXiv:2303.10420*, 2023.

Z. Yuan, H. Yuan, C. Tan, W. Wang, S. Huang, and F. Huang. Rrhf: Rank responses to align language models with human feedback without tears. *arXiv preprint arXiv:2304.05302*, 2023.

Y. Zhao, M. Khalman, R. Joshi, S. Narayan, M. Saleh, and P. J. Liu. Calibrating sequence likelihood improves conditional language generation. *arXiv preprint arXiv:2210.00045*, 2022.

Y. Zhao, R. Joshi, T. Liu, M. Khalman, M. Saleh, and P. J. Liu. Slic-hf: Sequence likelihood calibration with human feedback. *arXiv preprint arXiv:2305.10425*, 2023.

L. Zheng, W.-L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E. Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *arXiv preprint arXiv:2306.05685*, 2023.

D. M. Ziegler, N. Stiennon, J. Wu, T. B. Brown, A. Radford, D. Amodei, P. Christiano, and G. Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.