RLAIF: Scaling Reinforcement Learning from Human Feedback with AI Feedback

Harrison Lee, Samrat Phatale, Hassan Mansoor, Kellie Lu, Thomas Mesnard, Colton Bishop, Victor Carbune, Abhinav Rastogi
Google Research
{harrisonlee,samratph,hassan}@google.com

Abstract

Reinforcement learning from human feedback (RLHF) is effective at aligning large language models (LLMs) to human preferences, but gathering high-quality human preference labels is a key bottleneck. We conduct a head-to-head comparison of RLHF vs. RL from AI Feedback (RLAIF) - a technique where preferences are labeled by an off-the-shelf LLM in lieu of humans, and we find that they result in similar improvements. On the task of summarization, human evaluators prefer generations from both RLAIF and RLHF over a baseline supervised fine-tuned model in ∼70% of cases. Furthermore, when asked to rate RLAIF vs. RLHF summaries, humans prefer both at equal rates. These results suggest that RLAIF can yield human-level performance, offering a potential solution to the scalability limitations of RLHF.

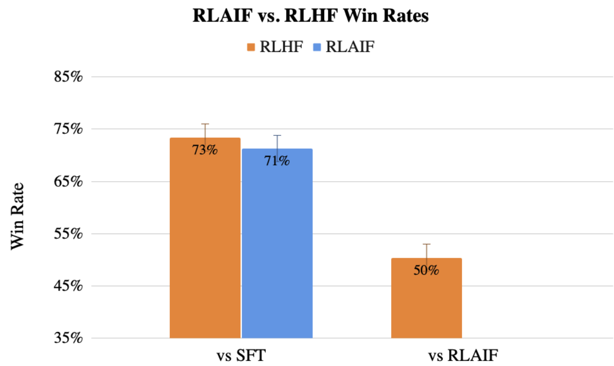

Figure 1: Human evaluators strongly prefer RLHF and RLAIF summaries over the supervised fine-tuned (SFT) baseline. The differences in win rates between *RLAIF vs. SFT* and *RLHF vs. SFT* are not statistically significant. Additionally, when compared head-to-head, RLAIF is equally preferred to RLHF by human evaluators. Error bars denote 95% confidence intervals.

1 Introduction

Reinforcement Learning from Human Feedback (RLHF) is an effective technique for aligning language models to human preferences (Stiennon et al., 2020; Ouyang et al., 2022) and is cited as one of the key drivers of success in modern conversational language models like ChatGPT and Bard (Liu et al., 2023; Manyika, 2023). By training with reinforcement learning (RL), language models can be optimized on complex, sequence-level objectives that are not easily differentiable with traditional supervised fine-tuning.

The need for high-quality human labels is an obstacle for scaling up RLHF, and one natural question is whether artificially generated labels can achieve comparable results. Several works have shown that large language models (LLMs) exhibit a high degree of alignment with human judgment - even outperforming humans on some tasks (Gilardi et al., 2023; Ding et al., 2023). Bai et al. (2022b) was the first to explore using AI preferences to train a reward model used for RL fine-tuning - a technique called "Reinforcement Learning from AI Feedback" (RLAIF)[1]. While they showed that utilizing a hybrid of human and AI preferences in conjunction with the "Constitutional AI" self-revision technique outperforms a supervised fine-tuned baseline, their work did not directly compare the efficacy of human vs. AI feedback, leaving the question unanswered whether RLAIF can be a suitable alternative to RLHF.

In this work, we directly compare RLAIF against RLHF on the task of summarization. Given a text and two candidate responses, we assign a preference label using an off-the-shelf LLM. We then train a reward model (RM) on the LLM preferences with a contrastive loss. Finally, we fine-tune a policy model with reinforcement learning, using

[1]We use "RLAIF" to denote training a reward model on AI-labeled preferences followed by conducting RL fine-tuning. This is distinct from "Constitutional AI", which improves upon a supervised learning model through iteratively asking an LLM to generate better responses according to a constitution. Both were introduced in Bai et al. (2022b) and are sometimes confused for one another.

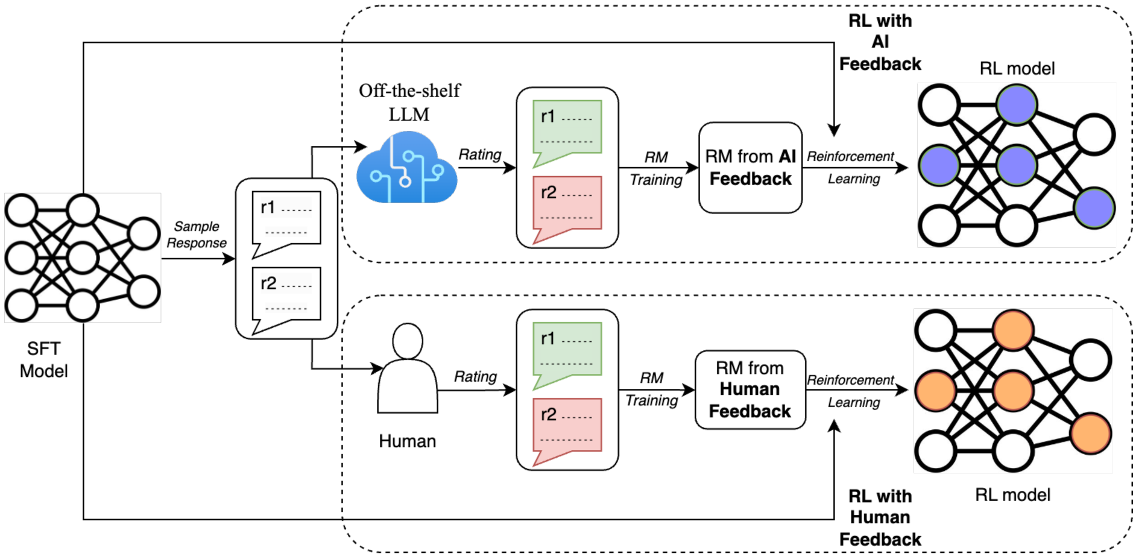

Figure 2: A diagram depicting RLAIF (top) vs. RLHF (bottom)

the RM to provide rewards.

Our results show that RLAIF achieves comparable performance to RLHF, measured in two ways. First, we observe that both RLAIF and RLHF policies are preferred by humans over a supervised fine-tuned (SFT) baseline 71% and 73% of the time, respectively, and the two win rates are not statistically significantly different. Second, when asked to directly compare generations from RLAIF vs. RLHF, humans prefer both at equal rates (i.e. 50% win rate). These results suggest that RLAIF is a viable alternative to RLHF that does not depend on human annotation and offers appealing scaling properties.

Additionally, we study techniques to maximize the alignment of AI-generated preferences with human preferences. We find that prompting our LLM with detailed instructions and soliciting chain-of-thought reasoning improve alignment. Surprisingly, we observe that both few-shot in-context learning and self-consistency - a process in which we sample multiple chain-of-thought rationales and average the final preferences - do not improve accuracy or even degrade it. Finally, we conduct scaling experiments to quantify the trade-offs between the size of the LLM labeler and the number of preference examples used in training vs. alignment with human preferences.

Our main contributions are the following:

- We demonstrate that RLAIF achieves comparable performance to RLHF on the task of summarization
- We compare various techniques for generating AI labels and identify optimal settings for RLAIF practitioners

2 Preliminaries

We first review the RLHF pipeline introduced in Stiennon et al. (2020); Ouyang et al. (2022), which consists of 3 phases: supervised fine-tuning, reward model training, and reinforcement learning-based fine-tuning.

2.1 Supervised Fine-tuning

A pre-trained LLM is fine-tuned on a high quality labeled dataset for a downstream task using token-level supervision to produce a supervised fine-tuned (SFT) model π^{SFT}.

2.2 Reward Modeling

Given an input x, we sample a pair of responses from one or more models $(y_1, y_2) \sim \pi$, where oftentimes the SFT model is used. The input and responses are sent to human annotators to rate which response is better according to some criteria. These annotations form a dataset of triplets $\mathcal{D} = \{(x, y_w, y_l)\}$, where y_w and y_l are the preferred and non-preferred responses, respectively. A reward model r_ϕ is trained by minimizing the following loss:

$$\mathcal{L}_r(\phi) = \underset{(x,y_w,y_l)\sim\mathcal{D}}{-\mathbb{E}} [\log \sigma(r_\phi(x, y_w) - r_\phi(x, y_l))]$$

where σ is the sigmoid function.

Preamble	`A good summary is a shorter piece of text that has the essence of the original. ... Given a piece of text and two of its possible summaries, output 1 or 2 to indicate which summary best adheres to coherence, accuracy, coverage, and overall quality as defined above.`

1-Shot Exemplar	`»»»» Example »»»»`
	
	`Text - We were best friends over 4 years ...`
	`Summary 1 - Broke up with best friend, should I wish her a happy birthday... And what do you think of no contact?`
	`Summary 2 - should I wish my ex happy birthday, I broke no contact, I'm trying to be more patient, I'm too needy, and I don't want her to think I'll keep being that guy.`
	
	`Preferred Summary=1`
	
	`»»»» Follow the instructions and the example(s) above »»»»`

Sample to Annotate	`Text -` **`{text}`** `Summary 1 -` **`{summary1}`** `Summary 2 -` **`{summary2}`**
Ending	`Preferred Summary=`

Table 1: An example of a prompt fed to an off-the-shelf LLM to generate AI preference labels. "{text}", "{summary1}", and "{summary2}" are populated with unlabeled examples, and a preference distribution is obtained by computing the softmax of the log probabilities of generating the tokens "1" vs. "2".

2.3 Reinforcement Learning

A policy π_θ^{RL} is initialized from the SFT model weights and then optimized with reinforcement learning to maximize the reward given by the RM, which serves as a proxy for human preferences. Optionally, a Kullback-Leibler (KL) divergence loss $\mathbb{D}_{KL}$ is added to the objective to penalize π_θ^{RL} for deviating from the original SFT policy π^{SFT}, controlled by the hyperparameter β - a technique similar to natural policy gradients (Kakade, 2001). The KL loss helps prevent π_θ^{RL} from drifting into a region where it generates language that is highly rewarded by the RM yet consists of low-quality or unnatural language - a phenomenon known as "reward hacking" (Everitt and Hutter, 2016; Amodei et al., 2016). The full optimization objective is described by the equation below:

$$\max_\theta \mathbb{E}[r_\phi(y|x) - \beta \mathbb{D}_{KL}(\pi_\theta^{RL}(y|x)||\pi^{SFT}(y|x))]$$

3 RLAIF Methodology

In this section, we describe the techniques used to generate preference labels with an LLM, how we conduct RL, and evaluation metrics.

3.1 Preference Labeling with LLMs

We annotate preferences among pairs of candidates with an "off-the-shelf" LLM, which is a model pre-trained or instruction-tuned for general usage but not fine-tuned for a specific downstream task. Given a piece of text and two candidate summaries, the LLM is asked to rate which summary is better. The input to the LLM is structured as follows (example in Table 1):

1. *Preamble* - Introduction and instructions describing the task at hand

2. *Few-shot exemplars (optional)* - An example of a text, a pair of summaries, a chain-of-thought rationale (if applicable), and a preference judgment

3. *Sample to annotate* - A text and a pair of summaries to be labeled

4. *Ending* - An ending string to prompt the LLM (e.g. "*Preferred Summary*=")

After the LLM is given the input, we obtain the log probabilities of generating the tokens "1" and "2" and compute the softmax to derive a preference distribution.

There are numerous alternatives to obtain preference labels from LLMs, such as decoding a free-form response from the model and extracting the preference heuristically (e.g. *output = "The first summary is better"*), or representing the preference distribution as a one-hot representation. However, we did not experiment with these alternatives because our approach already yielded high accuracy.

We experiment with two types of preambles: *"Base"*, which briefly asks "which summary is better?", and *"OpenAI"*, which closely mimics the rating instructions given to the human preference annotators that generated the OpenAI TL;DR preference dataset and contains detailed information about what constitutes a strong summary (Stiennon et al., 2020). See Appendix Table 6 for full preambles. We also experiment with in-context learning by adding few-shot exemplars in the prompt, where exemplars were manually selected to cover different topics.

3.1.1 Addressing Position Bias

The order in which candidates are shown to the LLM can bias which candidate it prefers (Pezeshkpour and Hruschka, 2023). We find evidence that such a position bias exists, especially with smaller sizes of LLM labelers (see Appendix A).

To mitigate position bias in preference labeling, we make two inferences for every pair of candidates, where the order in which candidates are presented to the LLM is reversed for the second inference. The results from both inferences are then averaged to obtain the final preference distribution.

3.1.2 Chain-of-thought Reasoning

We experiment with eliciting chain-of-thought (COT) reasoning from our AI labelers to improve alignment with human preferences (Wei et al., 2022). We replace the *Ending* of the standard prompt (i.e. "*Preferred Summary=*") with "*Consider the coherence, accuracy, coverage, and overall quality of each summary and explain which one is better. Rationale:*" and then decode a response from the LLM. Finally, we concatenate the original prompt, the response, and the original *Ending* string "*Preferred Summary=*" together, and follow the scoring procedure in Section 3.1 to obtain a preference distribution. See Figure 3 for an illustration.

In zero-shot prompts, the LLM is not given an example of what reasoning should look like, while in few-shot prompts, we provide examples of COT reasoning for the model to follow. See Tables 7 and 8 for examples.

3.1.3 Self-Consistency

For chain-of-thought prompts, we also experiment with self-consistency - a technique to improve upon chain-of-thought reasoning by sampling multiple reasoning paths and aggregating the final answer produced at the end of each path (Wang et al., 2022). Multiple chain-of-thought rationales are sampled with a non-zero decoding temperature, and then LLM preference distributions are obtained for each one - following the approach in Section 3.1.2. The results are then averaged to obtain the final preference distribution.

3.2 Reinforcement Learning from AI Feedback

After preferences are labeled by the LLM, a reward model (RM) is trained to predict preferences. Since our approach produces soft labels (e.g. $preferences_i = [0.6, 0.4]$), we apply a cross-entropy loss to the softmax of the reward scores generated by the RM instead of the loss mentioned in Section 2.2. The softmax converts the unbounded scores from the RM into a probability distribution.

Training a RM on a dataset of AI labels can be viewed as a form of model distillation, especially since our AI labeler is often larger and more powerful than our RM. An alternative approach is to bypass the RM and use AI feedback directly as a reward signal in RL, though this approach is more computationally expensive since the AI labeler is larger than the RM.

With the trained RM, we conduct reinforcement learning with a modified version of the Advantage Actor Critic (A2C) algorithm adapted to the language modeling domain (Mnih et al., 2016) (details in Appendix B). While many recent works use Proximal Policy Optimization (PPO) (Schulman et al., 2017) - a similar method that adds a few techniques to make training more conservative and stable (e.g. clipping the objective function), we utilize A2C given that it is simpler yet still effective for our problem.

3.3 Evaluation

We evaluate our results with three metrics - *AI Labeler Alignment*, *Pairwise Accuracy*, and *Win Rate*.

AI Labeler Alignment measures the accuracy of AI-labeled preferences with respect to human preferences. For a single example, it is calculated by

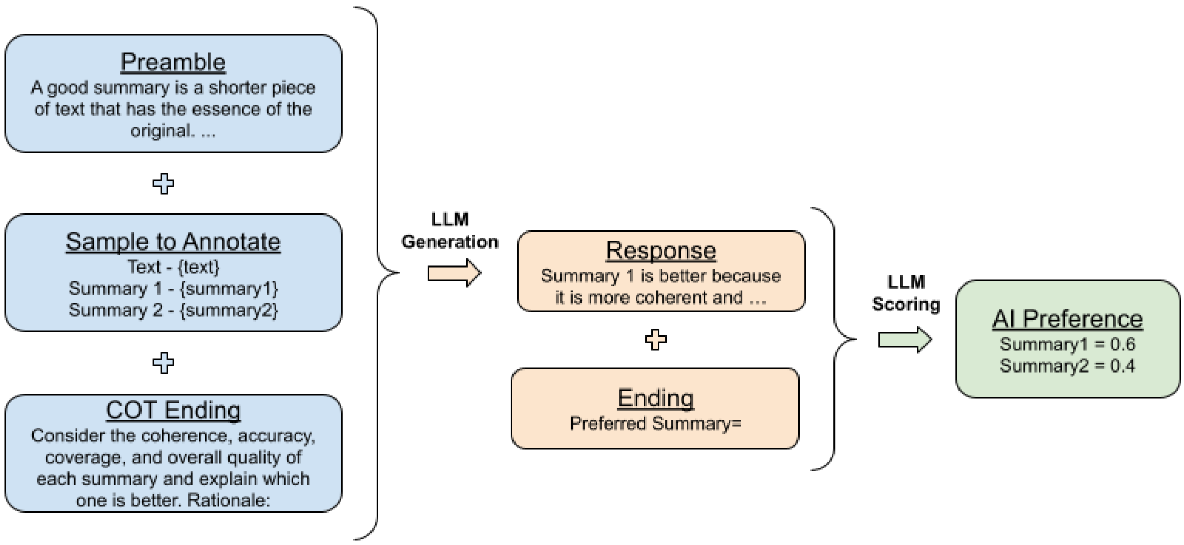

Figure 3: To derive an AI preference label, the LLM is first prompted to verbally explain its thoughts on the quality of the two candidates (blue). The LLM response is then appended to the original prompt (orange) and fed to the LLM a second time to generate a preference distribution over "1" vs. "2" based on their log probabilities (green).

converting a soft AI-labeled preference to a binary representation (e.g. $preferences_i = [0.6, 0.4] \rightarrow [1, 0]$), and then assigning a 1 if the label agrees with the target human preference and 0 otherwise. It can be expressed as follows:

$$Align = \frac{\sum_{i=1}^{D} \mathbb{1}[\arg\max_x pr_{ai_i} = \arg\max_x pr_{h_i}]}{|D|}$$

where pr_{ai} and p_h are binary representations of AI and human preferences, respectively, x is an index, and D is a dataset.

Pairwise Accuracy measures how accurate a trained reward model is with respect to a held-out set of human preferences. Given a shared context and pair of candidate responses, the *Pairwise Accuracy* is 1 if the RM scores the preferred candidate higher than the non-preferred candidate, according to the human label. Otherwise the value is 0. This quantity is averaged over multiple examples to measure the total accuracy of the RM.

Win Rate evaluates the end-to-end quality of two policies by measuring how often one policy is preferred by humans over another one. Given an input and two generations, human annotators select which generation is preferred. The percentage of instances where policy A is preferred over policy B is referred to as the *"Win Rate of A vs. B"*.

4 Experimental Details

4.1 Datasets

Following the work of Stiennon et al. (2020), we use the filtered Reddit TL;DR dataset curated by OpenAI. TL;DR contains ~3 million posts from Reddit[2] across a variety of topics (also known as "subreddits") alongside summaries of the posts written by the original authors. The data is additionally filtered by OpenAI to ensure high quality, which includes using a whitelist of subreddits that are understandable to the general population. Additionally, only posts where the summaries contain between 24 and 48 tokens are included. The filtered dataset contains 123,169 posts, where ~5% is held out as a validation set. More details on the dataset can be found in the original paper.

Additionally, OpenAI curated a human preference dataset from the filtered TL;DR dataset. For a given post, two candidate summaries were generated from different policies, and labelers were asked to rate which summary they preferred. The total dataset comprises ~92k pairwise comparisons.

4.2 LLM Labeling

For evaluating the efficacy of AI labeling techniques (e.g. prompting, self-consistency), we select examples from the TL;DR preference dataset where human annotators preferred one summary

[2] www.reddit.com

over the other with higher confidence[3]. We evaluate AI Labeler Alignment on a random 15% subset of the training split of the dataset to enable faster experiment iteration, yielding 2851 examples for evaluation. For reward model training, the full training split of the TL;DR preference dataset is labeled by the LLM and used for training - regardless of confidence scores.

We use PaLM 2 as our LLM for labeling preferences (Google et al., 2023). Unless otherwise specified, we use the Large model size with a maximum context length of 4096 tokens. For chain-of-thought generation, we set a maximum decoding length of 512 tokens and sample with temperature $T = 0$ (i.e. greedy decoding). For self-consistency experiments, we use temperature $T = 1$ with top-K sampling (Fan et al., 2018), where $K = 40$.

4.3 Model Training

We train a SFT model on OpenAI's filtered TL;DR dataset, using PaLM 2 Extra-Small (XS) as our initial checkpoint.

We then initialize our RMs from the SFT model and train them on OpenAI's TL;DR human preference dataset. For the results in Section Table 1 and 5.1, we generate AI-labeled preferences using PaLM 2 L, using the "OpenAI + COT 0-shot" prompt (see Section 5.2) without self consistency, and then train the RM on the full preference dataset.

For reinforcement learning, we train the policy with Advantage Actor Critic (A2C) as described in Appendix B. Both policy and value models are initialized from the SFT model. We rollout our policies using the filtered Reddit TL;DR dataset as our initial states.

For more training details, see Appendix C.

4.4 Human Evaluation

We collected 1200 ratings from humans to evaluate RLHF and RLAIF policies. For each rating task, the evaluator was presented with a post and 4 summaries generated from different policies (one from each of RLAIF, RLHF, SFT, and Human Reference) and asked to rank them in order of quality without ties. Posts were drawn from the held-out set of the TL;DR supervised fine-tuning dataset, which was not used in any other evaluation. Once these rankings were collected, it was possible to calculate win rates with respect to any two policies.

[3]This follows the evaluation procedure in Stiennon et al. (2020)

5 Results

5.1 RLAIF vs. RLHF

Our results show that RLAIF achieves similar performance to RLHF (see Table 1). RLAIF is preferred by human evaluators over the baseline SFT policy 71% of the time. In comparison, RLHF is preferred over SFT 73% of the time. While RLHF slightly outperforms RLAIF, the difference is not statistically significant[4]. We also directly compare the win rate of RLAIF vs. RLHF and find that they are equally preferred - i.e. the win rate is 50%. To better understand how RLAIF compares to RLHF, we qualitatively compare summaries generated by both policies in Section 6.

We also compare RLAIF and RLHF summaries vs. human-written reference summaries. RLAIF summaries are preferred over the reference summaries 79% of the time, and RLHF are preferred over the reference summaries 80% of the time. The difference in win rates between RLAIF and RLHF vs. the reference summaries is also not statistically significant.

One confounding factor in our results is that our RLAIF and RLHF policies tend to generate longer summaries than the SFT policy, which can account for some of the quality improvements. Similar to Stiennon et al. (2020), we conduct post-hoc analysis that suggests that while both RLAIF and RLHF policies benefit from producing longer summaries, both still outperform the SFT policy by a similar margin after controlling for length. Full details in Appendix D.

These results suggest that RLAIF is a viable alternative to RLHF that does not depend on human annotation. To understand how well these findings generalize to other NLP tasks, experiments on a broader range of tasks are required, which we leave to future work.

5.2 Prompting Techniques

We experiment with three types of prompting techniques - preamble specificity, chain-of-thought reasoning, and few-shot in-context learning - and report the results in Table 2.

Using the detailed OpenAI preamble improves alignment +1.3% (77.4% "OpenAI 0-shot" vs. 76.1% "Base 0-shot"), and chain-of-thought reasoning improves alignment +1.4% (77.5% "Base +

[4]When conducting a two-sample t-test, p-value = 0.25, which is greater than 0.05. Therefore, we cannot reject the null hypothesis.

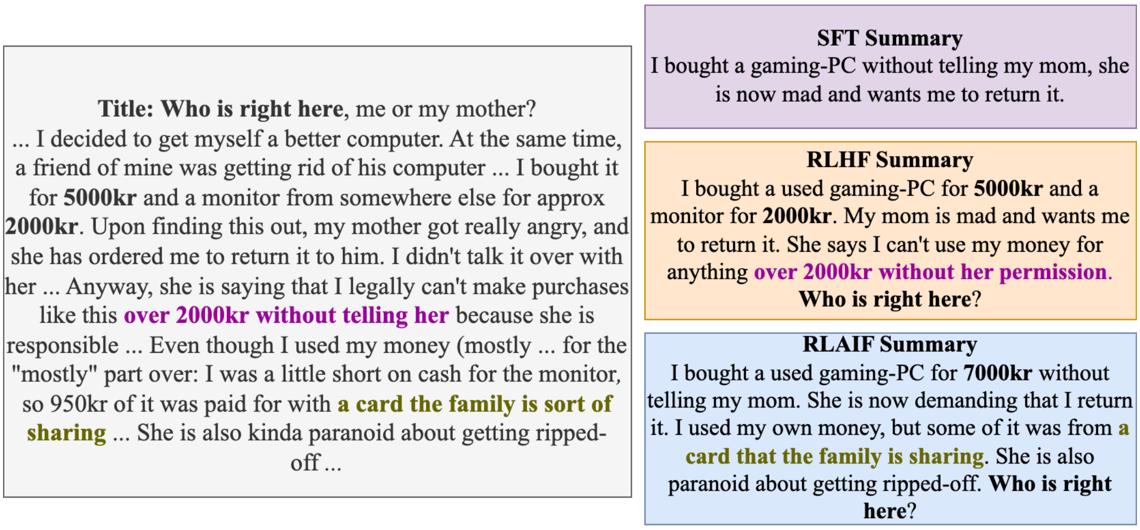

Figure 4: Example summaries generated by SFT, RLHF and RLAIF policies for a Reddit post. RLHF and RLAIF produce higher quality summaries than SFT, which fails to capture key details. Salient details are bolded.

Prompt	AI Labeler Alignment
Base 0-shot	76.1%
Base 1-shot	76.0%
Base 2-shot	75.7%
Base + COT 0-shot	77.5%
OpenAI 0-shot	77.4%
OpenAI 1-shot	76.2%
OpenAI 2-shot	76.3%
OpenAI 8-shot	69.8%
OpenAI + COT 0-shot	**78.0%**
OpenAI + COT 1-shot	77.4%
OpenAI + COT 2-shot	76.8%

Table 2: We observe that prompting with the detailed OpenAI preamble and eliciting chain-of-thought reasoning gives the highest AI Labeler Alignment. In-context learning does not improve accuracy, and possibly even makes it worse.

COT 0-shot" vs. 76.1% "Base 0-shot"). Though the improvement from combining the two techniques does not match the sum of their individual gains, the techniques are still complementary, together yielding +1.9% improvement.

We observe that few-shot in-context learning does not improve alignment, even potentially degrading it. For "OpenAI + COT k-shot" prompts, we see accuracy monotonically decrease as k increases from 0 to 2. One hypothesis is that the LLM is able to generate more useful chain-of-thought rationales on its own than when it follows the rationales given in our 1-shot and 2-shot exemplars (see Table 9 for examples).

To understand if adding more exemplars might yield improvements, we experimented with an 8-shot prompt and found that accuracy decreased by -7.6% (69.8% "OpenAI 8-shot" vs. 77.4% "OpenAI 0-shot"). We verified that all examples used in this experiment fit within our AI labeler's context length.

Overall, we observe that the optimal configuration employs a detailed preamble, chain-of-thought reasoning, and no in-context learning ("OpenAI + COT 0-shot"). This combination achieves an AI Labeler Alignment of 78.0%, which is +1.9% higher than using our most basic prompt ("Base 0-shot"). As a point of comparison, Stiennon et al. (2020) estimated that human inter-annotator agreement was 73-77% on the human preference dataset, suggesting that our LLM performs rather well. We use the "OpenAI + COT 0-shot" prompt for all other experiments.

5.3 Self-Consistency

Self-Consistency	AI Labeler Alignment
1 sample, T=0	**78.0%**
4 samples, T=1	72.6%
16 samples, T=1	72.8%

Table 3: Sampling several chain-of-thought rationales with $T = 1$ results in lower alignment with human preferences. Note: 1, 4, and 16 samples represent 2, 8, and 32 inferences given our position de-biasing technique (see Section 3.1.1).

We experiment with self-consistency using 4 and 16 samples with decoding temperature of 1 as described in Section 3.1.3, and both settings show drops in alignment of greater than -5% versus not using self-consistency. Manually inspecting chain-of-thought rationales did not reveal any common patterns for why self-consistency might result in lower accuracy (see examples in Table 10).

One hypothesis for the degradation in accuracy is that using a temperature of 1 leads the model to generate lower quality chain-of-thought rationales compared to greedy decoding, ultimately leading to worse accuracy overall. Using a temperature between 0 and 1 may yield better results.

5.4 Size of LLM Labeler

Model Size	AI Labeler Alignment
PaLM 2 XS	62.7%
PaLM 2 S	73.8%
PaLM 2 L	**78.0%**

Table 4: AI Labeler Alignment increases as the size of the LLM labeler increases.

Large model sizes are not widely accessible and can be slow and expensive to run. We experiment with labeling preferences with different model sizes and observe a strong relationship between alignment and size. Alignment drops -4.2% when moving from PaLM 2 Large (L) down to PaLM 2 Small (S), and it drops another -11.1% when moving down to PaLM 2 XS. This trend is consistent with scaling laws observed in other work (Kaplan et al., 2020). One contributing factor to the decline in performance could be the increase in position bias in smaller LLMs (see Appendix A).

On the end of this trend, these results also suggest that scaling up AI labeler size may produce even higher quality preference labels. Since the AI labeler is only used to generate preference examples once and is not queried during RL training, using an even larger AI labeler is not necessarily prohibitively expensive. Furthermore, Section 5.5 suggests that a small number of examples may be sufficient to train a powerful RM (e.g. on the order of O(1k)), further reducing the costs of using a larger labeler model.

5.5 Number of Preference Examples

To understand how RM accuracy changes with the number of training examples, we train a RM on varying amounts of AI-labeled preference examples and evaluate Pairwise Accuracy on a held-out set of human preferences. We obtain different amounts of training examples by randomly subsampling the full preference datasets. Results are displayed in Figure 5.

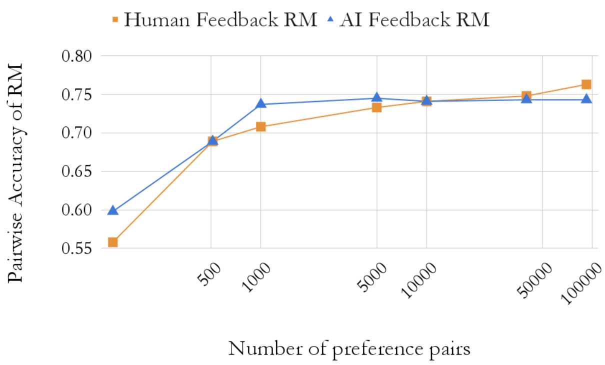

Figure 5: RM accuracy on a held-out set of human preferences increases rapidly as more preference pairs are used in training. After training on a few thousand examples, performance is close to training on the full dataset. The x-axis is in log-scale.

We observe that the performance of the AI preference RM quickly plateaus after training on a few thousand examples. The RM achieves $\sim$60% accuracy when training on only 128 examples and then reaches an accuracy close to that of training on the full dataset when training with only 5,000 examples (roughly $\frac{1}{20}$ of the full dataset).

We also conduct a parallel set of experiments on a RM trained on human preferences. We find that the human and AI RMs follow similar scaling curves. One difference is that the human preference RM appears to continually improve as the number of training examples increases, though more training examples only bring small improvements to accuracy. This trend suggests that RMs trained on AI preferences may not benefit as much from scaling up the number of training examples as RMs trained on human preferences.

Given the limited improvement from scaling up the number of AI preference examples, more resources may be better spent on labeling with larger model sizes (see Section 5.4) rather than labeling more preference examples.

6 Qualitative Analysis

To better understand how RLAIF compares to RLHF, we manually inspected summaries generated by both policies. In many cases, the two policies produced similar summaries, which is reflected in their similar win rates (see Section 5.1). However, we identified two patterns where they frequently diverged.

One pattern we observed is that RLAIF appears less likely to hallucinate than RLHF. The hallucinations in RLHF are often plausible but are inconsistent with the original text. For instance, in Example #1 of Table 11, the RLHF summary states that the author is 20 years old, but this is not mentioned or implied by the original text.

Another pattern we observed is that RLAIF sometimes produces less coherent or grammatical summaries than RLHF. For instance, in Example #1 of Table 12, the RLAIF summary produces run-on sentences.

Overall, while we observe certain tendencies for each policy, both produce high-quality summaries that are relatively similar.

7 Related Work

LLMs (Brown et al., 2020; Thoppilan et al., 2022; Chowdhery et al., 2022; Google et al., 2023; OpenAI, 2023) have shown impressive performance over a wide range of NLP tasks. For several of these tasks, RL has emerged as an effective optimization technique. While initial applications of RL on tasks such as translation (Wu et al., 2016, 2018) and summarization (Gao et al., 2019; Wu and Hu, 2018) used automatic evaluation metrics as rewards, such simplified formulations of rewards did not fully align with human notions of quality.

Reinforcement learning from human feedback Christiano et al. (2017) has been used as a technique to directly align LLMs with human preferences (Ziegler et al., 2019) by training a reward model on pairwise comparisons of natural language responses, and has been successfully applied for summarization (Stiennon et al., 2020), generalized instruction following (Ouyang et al., 2022; Lai et al., 2023), dialogue (Gilardi et al., 2023; Manyika, 2023; Glaese et al., 2022; Bai et al., 2022a) and question answering (Nakano et al., 2021).

LLMs have also been extensively used for data generation (Wang et al., 2021; Meng et al., 2023), augmentation (Feng et al., 2021) and in self-training setups (Wang et al., 2022; Madaan et al., 2023). Bai et al. (2022b) introduced the idea of RL from AI feedback (RLAIF), which used LLM labeled preferences in conjunction with human labeled preferences to jointly optimize for the two conflicting objectives of helpfulness and harmlessness. Recent works have also explored related techniques for generating rewards from LLMs (Roit et al., 2023; Kwon et al., 2022; Yang et al., 2023). These works demonstrate that LLMs can generate useful signals for RL fine-tuning, which inspired this work's investigation into whether LLMs can serve as a viable alternative to humans in collecting preference labels for reinforcement learning.

8 Conclusion

In this work, we show that RLAIF can produce comparable improvements to RLHF without depending on human annotators. Our experiments show that RLAIF greatly improves upon a SFT baseline, and the margin of improvement is on par with that of RLHF. In head-to-head comparisons, RLAIF and RLHF are preferred at similar rates by humans. We also study various AI labeling techniques and conduct scaling studies to understand the optimal settings for generating aligned preferences.

While this work highlights the potential of RLAIF, we note some limitations of these findings. First, this work only explores the task of summarization, leaving an open question about generalizability to other tasks. Second, we did not estimate whether LLM inference is advantageous versus human labeling in terms of monetary costs. Additionally, there remain many interesting open questions, such as whether RLHF combined with RLAIF can outperform a single approach alone, how well using a LLM to directly assign rewards performs, whether improving AI Labeler Alignment translates to improved final policies, and whether using a LLM labeler the same size as the policy model can further improve the policy (i.e. whether a model can "self-improve"). We leave these questions for future work.

We hope that this paper motivates further research in the area of RLAIF.

Acknowledgements

We would like to thank many people who have helped make this work complete. We thank Chen Zhu for optimizing our LLM inference setup, Le Hou for suggesting prompt improvements and experimenting with self-consistency, Johan Ferret for tips on how to tune better RL policies, and Léonard Hussenot for bringing the problem of position bias in LLMs to our attention.

We thank everyone who thoroughly reviewed our work and provided valuable feedback: Hakim

Sidahmed, Michal Valko, Nevan Wichers, Sian Gooding, Sushant Prakash, and Yuan Cao.

Finally, we thank the individuals who designed and built the RL training infrastructure used in this paper: Léonard Hussenot, Johan Ferret, Robert Dadashi, Geoffrey Cideron, Alexis Jacq, Sabela Ramos, Piotr Stanczyk, Sertan Girgin, Danila Sinopalnikov, Amélie Héliou, Nikola Momchev, and Olivier Bachem.

References

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. 2016. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. 2022a. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, Jamie Kerr, Jared Mueller, Jeffrey Ladish, Joshua Landau, Kamal Ndousse, Kamile Lukosuite, Liane Lovitt, Michael Sellitto, Nelson Elhage, Nicholas Schiefer, Noemi Mercado, Nova DasSarma, Robert Lasenby, Robin Larson, Sam Ringer, Scott Johnston, Shauna Kravec, Sheer El Showk, Stanislav Fort, Tamera Lanham, Timothy Telleen-Lawton, Tom Conerly, Tom Henighan, Tristan Hume, Samuel R. Bowman, Zac Hatfield-Dodds, Ben Mann, Dario Amodei, Nicholas Joseph, Sam McCandlish, Tom Brown, and Jared Kaplan. 2022b. Constitutional ai: Harmlessness from ai feedback.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30.

Bosheng Ding, Chengwei Qin, Linlin Liu, Yew Ken Chia, Boyang Li, Shafiq Joty, and Lidong Bing. 2023. Is GPT-3 a good data annotator? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11173–11195, Toronto, Canada. Association for Computational Linguistics.

Tom Everitt and Marcus Hutter. 2016. Avoiding wireheading with value reinforcement learning. In *Artificial General Intelligence: 9th International Conference, AGI 2016, New York, NY, USA, July 16-19, 2016, Proceedings 9*, pages 12–22. Springer.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898, Melbourne, Australia. Association for Computational Linguistics.

Steven Y. Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. 2021. A survey of data augmentation approaches for NLP. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 968–988, Online. Association for Computational Linguistics.

Yang Gao, Christian M Meyer, Mohsen Mesgar, and Iryna Gurevych. 2019. Reward learning for efficient reinforcement learning in extractive document summarisation. *arXiv preprint arXiv:1907.12894*.

Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. 2023. Chatgpt outperforms crowd-workers for text-annotation tasks. *arXiv preprint arXiv:2303.15056*.

Amelia Glaese, Nat McAleese, Maja Trębacz, John Aslanides, Vlad Firoiu, Timo Ewalds, Maribeth Rauh, Laura Weidinger, Martin Chadwick, Phoebe Thacker, et al. 2022. Improving alignment of dialogue agents via targeted human judgements. *arXiv preprint arXiv:2209.14375*.

Rohan Anil Google, Andrew M. Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, Eric Chu, Jonathan H. Clark, Laurent El Shafey, Yanping Huang, Kathy Meier-Hellstern, Gaurav Mishra, Erica Moreira, Mark Omernick, Kevin Robinson, Sebastian Ruder, Yi Tay, Kefan Xiao, Yuanzhong Xu, Yujing Zhang, Gustavo Hernandez Abrego, Junwhan Ahn, Jacob Austin, Paul Barham, Jan Botha, James Bradbury, Siddhartha Brahma, Kevin Brooks, Michele Catasta, Yong Cheng, Colin Cherry, Christopher A. Choquette-Choo, Aakanksha Chowdhery, Clément Crepy, Shachi Dave, Mostafa Dehghani, Sunipa Dev, Jacob Devlin, Mark Díaz, Nan Du, Ethan Dyer, Vlad Feinberg, Fangxiaoyu Feng, Vlad Fienber, Markus Freitag, Xavier Garcia, Sebastian Gehrmann, Lucas Gonzalez, Guy Gur-Ari, Steven Hand, Hadi Hashemi, Le Hou, Joshua Howland, Andrea Hu, Jeffrey Hui, Jeremy Hurwitz, Michael Isard, Abe Ittycheriah, Matthew Jagielski, Wenhao Jia, Kathleen

Kenealy, Maxim Krikun, Sneha Kudugunta, Chang Lan, Katherine Lee, Benjamin Lee, Eric Li, Music Li, Wei Li, YaGuang Li, Jian Li, Hyeontaek Lim, Hanzhao Lin, Zhongtao Liu, Frederick Liu, Marcello Maggioni, Aroma Mahendru, Joshua Maynez, Vedant Misra, Maysam Moussalem, Zachary Nado, John Nham, Eric Ni, Andrew Nystrom, Alicia Parrish, Marie Pellat, Martin Polacek, Alex Polozov, Reiner Pope, Siyuan Qiao, Emily Reif, Bryan Richter, Parker Riley, Alex Castro Ros, Aurko Roy, Brennan Saeta, Rajkumar Samuel, Renee Shelby, Ambrose Slone, Daniel Smilkov, David R. So, Daniel Sohn, Simon Tokumine, Dasha Valter, Vijay Vasudevan, Kiran Vodrahalli, Xuezhi Wang, Pidong Wang, Zirui Wang, Tao Wang, John Wieting, Yuhuai Wu, Kelvin Xu, Yunhan Xu, Linting Xue, Pengcheng Yin, Jiahui Yu, Qiao Zhang, Steven Zheng, Ce Zheng, Weikang Zhou, Denny Zhou, Slav Petrov, and Yonghui Wu. 2023. Palm 2 technical report.

Sham M Kakade. 2001. A natural policy gradient. *Advances in neural information processing systems*, 14.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Minae Kwon, Sang Michael Xie, Kalesha Bullard, and Dorsa Sadigh. 2022. Reward design with language models. In *The Eleventh International Conference on Learning Representations*.

Viet Dac Lai, Chien Van Nguyen, Nghia Trung Ngo, Thuat Nguyen, Franck Dernoncourt, Ryan A Rossi, and Thien Huu Nguyen. 2023. Okapi: Instruction-tuned large language models in multiple languages with reinforcement learning from human feedback. *arXiv preprint arXiv:2307.16039*.

Yiheng Liu, Tianle Han, Siyuan Ma, Jiayue Zhang, Yuanyuan Yang, Jiaming Tian, Hao He, Antong Li, Mengshen He, Zhengliang Liu, et al. 2023. Summary of chatgpt/gpt-4 research and perspective towards the future of large language models. *arXiv preprint arXiv:2304.01852*.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. 2023. Self-refine: Iterative refinement with self-feedback. *arXiv preprint arXiv:2303.17651*.

James Manyika. 2023. An overview of bard: an early experiment with generative ai. `https://ai.google/static/documents/google-about-bard.pdf`. Accessed: 2023-08-23.

Yu Meng, Martin Michalski, Jiaxin Huang, Yu Zhang, Tarek Abdelzaher, and Jiawei Han. 2023. Tuning language models as training data generators for augmentation-enhanced few-shot learning. In *International Conference on Machine Learning*, pages 24457–24477. PMLR.

Volodymyr Mnih, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Timothy P. Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. 2016. Asynchronous methods for deep reinforcement learning. *CoRR*, abs/1602.01783.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. 2021. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*.

OpenAI. 2023. Gpt-4 technical report.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Pouya Pezeshkpour and Estevam Hruschka. 2023. Large language models sensitivity to the order of options in multiple-choice questions. *arXiv preprint arXiv:2308.11483*.

Paul Roit, Johan Ferret, Lior Shani, Roee Aharoni, Geoffrey Cideron, Robert Dadashi, Matthieu Geist, Sertan Girgin, Léonard Hussenot, Orgad Keller, et al. 2023. Factually consistent summarization via reinforcement learning with textual entailment feedback. *arXiv preprint arXiv:2306.00186*.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost. *CoRR*, abs/1804.04235.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33:3008–3021.

Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. 1999. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. 2022. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. In

The Eleventh International Conference on Learning Representations.

Zirui Wang, Adams Wei Yu, Orhan Firat, and Yuan Cao. 2021. Towards zero-label language learning. *arXiv preprint arXiv:2109.09193*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256.

Lijun Wu, Fei Tian, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. 2018. A study of reinforcement learning for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3612–3621.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Yuxiang Wu and Baotian Hu. 2018. Learning to extract coherent summary via deep reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, page 5602.

Kevin Yang, Dan Klein, Asli Celikyilmaz, Nanyun Peng, and Yuandong Tian. 2023. Rlcd: Reinforcement learning from contrast distillation for language model alignment.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2019. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*.

A Position Bias in LLM Labelers

Model Size	% Same Position Preferred
PaLM 2 L	18%
PaLM 2 S	21%
PaLM 2 XS	56%

Table 5: Position bias is more prevalent in smaller model sizes, as indicated by *"% Same Position Preferred"*, which measures the percentage of examples where the LLM prefers the same position even after swapping the order of candidates. Analysis is conducted using the "*OpenAI + COT 0-shot*" prompt.

Our analysis suggests that the LLMs used for preference labeling are biased by the order in which candidates are shown. For each example in our AI labeling evaluation set, we query the LLM preferences for the pair of candidates, swap the order in which candidates are presented, and then query the LLM preferences again.

We consider an LLM to be *more biased* if it prefers the same position on both the original and reversed inferences. For example, let candidates A and B be in positions 1 and 2 for the first inference and then in positions 2 and 1 for the second, respectively. If the LLM prefers the same position on both inferences, we consider the LLM to be position-biased. We measure position bias by computing *"% Same Position Preferred"* - the percentage of inference pairs where this occurs, and a higher metric value indicates a more biased LLM.

We find that PaLM 2 L, S, and XS prefer the same position 18%, 21%, and 56% of the time, respectively (see Table 5), suggesting that position bias is inversely proportional to model size. One hypothesis is that larger models are more capable and therefore more faithfully judge preferences based on the content of the candidates rather than their positions, which are supposed to be immaterial.

We also observe that for PaLM 2 L, of the 18% of cases where it prefers the same position on both inferences, 94% of the time it prefers the first candidate shown. On the other hand, PaLM 2 S and XS show affinity for the second candidate shown, preferring it 91% and 99% of the time, respectively, when the same position is preferred on both inferences. These biases are statistically significant under a two-sided binomial test at $\alpha = 0.05$.

B A2C for Language Models

Consider a generic MDP $(\mathcal{X}, \mathcal{A}, R, P, \gamma)$. At each step t, given the current state $X_t \in \mathcal{X}$ and the next action $A_t \in \mathcal{A}$, the model receives a reward $R_t = R(X_t, A_t)$ and transitions to the next state $X_{t+1} = (X_t, A_t)$.

In the context of language models, X_t is the concatenation of the input text and all text the policy has generated up to time t. Action A_t is the token decoded at time t by the stochastic policy $\pi_\theta(\cdot|X_t)$ from the considered vocabulary, where θ represents the policy parameters. Finally, the reward R_t is given by the RM. The RM is only evaluated when the language model response has been fully generated, and therefore all rewards before the last token are 0 while the reward corresponding to the final token is $R_{T_{last}}$.

The cumulative sum of rewards received when following the policy π from a state-action pair $(X_t = x, A_t = a)$ is called the return. Generally, it is defined as $Z^{\pi}_{x,a} = \sum_{s=t}^{T_{last}} \gamma^{s-t} R_s$. However, since only the terminal reward is non-zero and we use $\gamma = 1$, the return can be simplified to $Z^{\pi}_{x,a} = R_{T_{last}}$.

Given a trajectory $(X_t, A_t, R_t)_{t=0}^{T_{last}}$ generated under π_θ, the Advantage Actor Critic estimator is defined as follows:

$$\mathcal{L}_{\text{A2C}} = \sum_{t \geq 0} \log \pi_\theta(A_t|X_t) \overline{\left(R_{T_{last}} - V^{\pi}_{\psi}(X_t)\right)}$$

where the bar notation denotes that no gradient is passed through the advantage term during the policy training phase.

The baseline value function $V^{\pi}_{\psi}(x)$ estimates the return-to-go $R_{T_{last}}$ when following the policy π_θ and is parametrized by ψ (Williams, 1992; Sutton et al., 1999). It is trained with the following loss:

$$\mathcal{L}_{\text{baseline}} = \sum_t (R_{T_{last}} - V^{\pi}_{\psi}(X_t))^2$$

C Model Training Details

We train a SFT model with a batch size of 128 for a single epoch. We use the Adafactor (Shazeer and Stern, 2018) optimizer with a learning rate of 10^{-5}, and we set a maximum input and output length of 1024 and 128 tokens, respectively.

Each RM is trained for 3 epochs, which is when loss and accuracy curves plateau. We use the Adafactor optimizer with a learning rate of 10^{-5} and a batch size of 128.

For reinforcement learning, we sample from our language model policies with a temperature of $T = 0.9$ to encourage exploration. We train with a batch size of 128 and learning rate of 10^{-5} for 8 epochs, resulting in $\sim$1 million episodes. We set $\beta = 0.05$ for the KL divergence loss.

D Controlling for Summary Length

Our RLAIF and RLHF policies tend to generate longer summaries than the baseline SFT policy. For the summaries sent to human evaluation, the mean character-length of summaries produced by RLAIF, RLHF, and SFT policies were 164, 161, and 132, respectively. We conduct post-hoc analysis to estimate the win rates of RLAIF and RLHF vs. SFT after controlling for length.

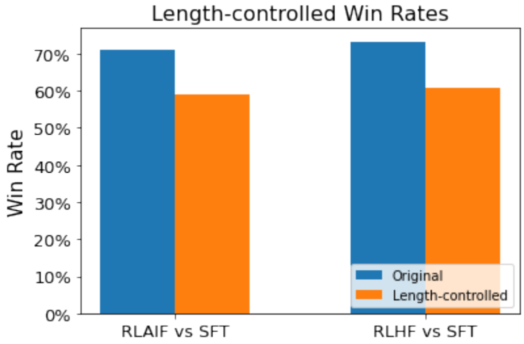

Figure 6: After controlling for summary length, RLAIF and RLHF policies both still outperform the baseline SFT policy and achieve similar win rate.

We take an approach similar to Stiennon et al. (2020). For each of our RL policies, we train a logistic regression model where the input is the ratio of the RL summary length to the SFT summary length (in characters) and the target is a binary label indicating whether RL was preferred to SFT. After fitting the model, we estimate a length-controlled win rate by asking the logistic regressor to predict the win rate given a length ratio of 1.0, which represents the scenario where both the RL and SFT summaries are of equal length.

After controlling for length, our estimated win rates for RLAIF and RLHF vs. SFT are 59% and 61%, respectively (see Figure 6). Both RL policies continue to outperform the SFT policy by a similar margin, supporting our initial conclusion that RLAIF is comparable to RLHF.

We note that this post-hoc method of controlling for length is imperfect, as it assumes the logistic regression model can accurately learn the relationship between summary length and human preference. A more principled approach would be to have all policies generate summaries of similar length (e.g. by encouraging policies to generate summaries of a fixed length during optimization).

"Base" preamble	`You are an expert summary rater. Given a piece of text and two of its possible summaries, output 1 or 2 to indicate which summary is better.`
"OpenAI" preamble	`A good summary is a shorter piece of text that has the essence of the original. It tries to accomplish the same purpose and conveys the key information from the original post. Below we define four evaluation axes for summary quality: coherence, accuracy, coverage, and overall quality.` `Coherence: This axis answers the question “how coherent is the summary on its own?” A summary is coherent if it’s easy to understand when read on its own and free of English errors. A summary is not coherent if it’s difficult to understand what the summary is trying to say. Generally, it’s more important that the summary is understandable than it being free of grammar errors.`  `Accuracy: This axis answers the question “does the factual information in the summary accurately match the post?” A summary is accurate if it doesn’t say things that aren’t in the article, it doesn’t mix up people, and generally is not misleading.`  `Coverage: This axis answers the question “how well does the summary cover the important information in the post?” A summary has good coverage if it mentions the main information from the post that’s important to understand the situation described in the post. A summary has poor coverage if someone reading only the summary would be missing several important pieces of information about the situation in the post. A summary with good coverage should also match the purpose of the original post (e.g. to ask for advice).`  `Overall quality: This axis answers the question “how good is the summary overall at representing the post?” This can encompass all of the above axes of quality, as well as others you feel are important. If it’s hard to find ways to make the summary better, the overall quality is good. If there are lots of different ways the summary can be made better, the overall quality is bad.`  `You are an expert summary rater. Given a piece of text and two of its possible summaries, output 1 or 2 to indicate which summary best adheres to coherence, accuracy, coverage, and overall quality as defined above.`

Table 6: The "Base" and "OpenAI" preambles given to the LLM labeler to obtain preference labels.

Preamble	`A good summary is a shorter piece of text that has the essence of the original. ... Given a piece of text and two of its possible summaries, explain which summary best adheres to coherence, accuracy, coverage, and overall quality as defined above.`

Sample to Annotate	`Text - {text}`
	`Summary 1 - {summary1}`
	`Summary 2 - {summary2}`

Ending	`Consider the coherence, accuracy, coverage, and overall quality of each summary and explain which one is better.`
	
	`Rationale:`

Table 7: The template used for the "OpenAI + COT 0-shot" prompt, with some text removed for brevity. For COT prompts, we first decode a response from the LLM and then concatenate it with the original prompt and the ending *"Preferred Summary="* before following the scoring procedure in Section 3.1 to obtain a preference distribution.

Preamble	A good summary is a shorter piece of text that has the essence of the original. ... Given a piece of text and two of its possible summaries, explain which summary best adheres to coherence, accuracy, coverage, and overall quality as defined above.
1-shot Exemplar	»»»» Example »»»» Text - We were best friends over 4 years ... Summary 1 - Broke up with best friend, should I wish her a happy birthday... And what do you think of no contact? Summary 2 - should I wish my ex happy birthday, I broke no contact, I'm trying to be more patient, I'm too needy, and I don't want her to think I'll keep being that guy. Thoughts on Summary 1 - Coherence - 7. Rationale: The summary is generally understandable, though it could be written with better grammar. Accuracy - 9. Rationale: The summary doesn't say things that aren't in the original text, and isn't misleading. Coverage - 6. Rationale: The summary covers most of the important information in the post and conveys the gist of the original text. However, it places more emphasis on "no contact" and could have mentioned the smothering/neediness to be more complete. Overall Quality - 7. Rationale: The summary represents the post fairly well with only minor areas where it could be improved. Thoughts on Summary 2 - Coherence - 3. Rationale: The summary is long-winded and has several grammatical errors. Accuracy - 4. Rationale: The summary mentions that the author broke no contact, but this is incorrect. Otherwise, it is accurate. Coverage - 8. Rationale: The summary covers the key points in the original text. Overall Quality - 4. Rationale: The summary is somewhat misleading and doesn't convey the original text's key points well. Preferred Summary=1 »»»» Follow the instructions and the example(s) above »»»»
Sample to Annotate	Text - {text} Summary 1 - {summary1} Summary 2 - {summary2}
Ending	Thoughts on Summary 1 -

Table 8: The template used for the "OpenAI + COT 1-shot" prompt, with some text removed for brevity.

Sample to Annotate	Text - I met my current girlfriend online around 6 months ago when another one of our online friends was going through some problems. ... However, I feel incredibly insecure as she has yet to begin college. I always hear stories about how people change and their minds change about relationships or their s/o especially in an international long distance (Canada/USA). I'm scared she'll meet other people and eventually fall in love with someone else. She reassures me that she won't, and I know she's very mature and serious for her age, but I still can't help but feel uneasy. Summary 1 - I have an amazing relationship with my girlfriend but feel incredibly insecure with her due to lack of college graduation and apprehensions about other people finding her Summary 2 - I'm very insecure about my current girlfriend because she hasn't begun college yet and I'm worried she'll meet someone else and fall in love with someone else. Am I being ridiculous?
COT 0-shot Rationale	Rationale: Summary 2 is better. Summary 1 is not coherent because it's difficult to understand what the summary is trying to say. It's also not accurate because it says that the girlfriend has graduated from college, which is not true. Summary 2 is coherent and accurate. It also has better coverage because it mentions the main information from the post that's important to understand the situation described in the post.
COT 1-shot Rationale	Thoughts on Summary 1 - Coherence - 7. Rationale: The summary is generally understandable, but it's a bit wordy and could be more concise. Accuracy - 8. Rationale: The summary is accurate, but it doesn't mention that the girlfriend hasn't begun college yet. Coverage - 7. Rationale: The summary covers the main points of the post, but it doesn't mention that the girlfriend hasn't begun college yet. Overall Quality - 7. Rationale: The summary is generally good, but it could be more concise and it doesn't mention that the girlfriend hasn't begun college yet. Thoughts on Summary 2 - Coherence - 9. Rationale: The summary is concise and easy to understand. Accuracy - 9. Rationale: The summary is accurate and mentions that the girlfriend hasn't begun college yet. Coverage - 9. Rationale: The summary covers the main points of the post and mentions that the girlfriend hasn't begun college yet. Overall Quality - 9. Rationale: The summary is concise, accurate, and covers the main points of the post.

Table 9: An example of the different chain-of-thought rationales produced by the 0-shot ("OpenAI + COT 0-shot") vs. 1-shot ("OpenAI + COT 1-shot") prompts.

Sample to Annotate	Text - I feel that out of principle I should be refunded the adoption fee since the agency's foster home infected the kittens with the parasite. Both cats were born in the foster home and there are 20 other cats. Do I have any legal right to ask for the fee back? Or help with the cost of treating? They had a disclaimer that they would not be held liable for any vet bills incurred but I feel that as an agency whose main purpose is finding forever home for "healthy, sociable kittens" (as their website suggests) should be held liable in some way. Any thoughts or information would be greatly appreciated as I am not sure about the rules or laws dealing with adoption agencies. Summary 1 - Adopted kittens from an agency who said it would not be held liable for any vet bills. Two days later they are diagnosed with Giardia. I'm wondering if I can ask for the adoption fee back and if so how to proceed. Summary 2 - Adopted kittens have parasite. Want to know if I can get reimbursed from the agency.
Rationale @ $T = 0$	Rationale: Summary 1 is better. It is coherent, accurate, and has good coverage. It also has good overall quality. Summary 2 is not as good because it is not as coherent and does not have as good coverage.
Rationale @ $T = 1$	Rationale: Summary 2 is more coherent and accurate. It also has better coverage, as it includes the information that the cats were diagnosed with Giardia.

Table 10: An example comparing chain-of-thought rationales produced at different temperatures for self-consistency experiments.

Example #1	RLAIF summary: I have anxiety and I think I have a heart problem. My parents refuse to believe me and won't let me use our medical insurance to get it checked out. What should I do? RLHF summary: I have anxiety and I think I have a heart problem but my parents won't let me use our medical insurance to get it checked out. What should I do? I'm 20 years old. ..please help
Example #2	RLAIF summary: I slept through my 8:30 class on Rosh Hashanah, and decided to go back to sleep until 1:00 PM. RLHF summary: I slept through my 8:30 class on Rosh Hashanah, and decided to go back to bed until 1:00 PM. I'm a lazy f***.
Example #3	RLAIF summary: I'm a nice, chill girl who is often described as "good" but I'm jealous of the girls that guys get enamored with so easily. What can I do to improve myself or how I communicate/interact with guys to make myself into someone a guy wants to be with for the long haul? RLHF summary: I'm a nice, chill girl who is often described as "boring". I'm jealous of the girls that guys get enamored with so easily. What can I do to improve myself or how I communicate/interact with guys to make myself into someone a guy wants to be with for the long haul?

Table 11: We observe that the RLHF policy tends to hallucinate more frequently than the RLAIF policy. Hallucinations are highlighted in red.

Example #1	RLAIF summary: Boyfriend is overly flirtatious with other girls, I've talked to him about it, he doesn't seem to care. It's causing trust issues. Am I overreacting? What else can I do? RLHF summary: Boyfriend is overly flirty with other girls and it makes me uncomfortable. I've talked to him about it multiple times, but he doesn't seem to care. What else can I do? Am I overreacting?
Example #2	RLAIF summary: Asked a girl to prom, things were going great until I asked her. Now our conversations are awkward and I'm not sure if I should ask her out. Should I just give up? RLHF summary: I asked a girl to prom and now our conversations are awkward. I like her and I'm wondering if I should ask her out or just give up. I'm not sure
Example #3	RLAIF summary: Broke up with girl 1, she started dating my best friend. Dated girl 2, she became unstable and we broke up. I still feel attached to both. Why? How do I get over this? Any advice? RLHF summary: I broke up with a girl I was with for years, and she started dating my best friend. I started dating a new girl, and she became depressed and distant. I still feel attached to both. Why?

Table 12: Another pattern identified through manually inspected summaries is that summaries from the RLAIF policy tend to be less coherent and grammatical than summaries from the RLHF policy. Less coherent phrases are highlighted in red.