A Simple Reward Composition Method for Effectively Finetuning the Large Language Model with Diverse Feedbacks

Anonymous ACL submission

Abstract

Reinforcement learning from human feedback has emerged as a promising paradigm that significantly enhances the performance of large language models. Typically, reward models are trained to align with human preferences and are then utilized to optimize the pretrained language models. However, given the multifaceted nature of human preferences, it is challenging to appropriately combine rewards from different aspects. Recent studies have developed algorithms to address this issue by employing techniques such as weighting, and ranking. Nonetheless, these methods can perform poorly in certain scenarios despite their elegant design. In this paper, we explore the reward composition problem from a novel perspective. We posit that different reward models focus on distinct optimization directions, which the language model cannot discern, perceiving only the reward value. To formulate an appropriate reward signal, we introduce a global reward model that composes rewards from various aspects in a self-supervised manner, a simple yet effective approach. This global reward model can be trained without the need for additional supervised data and is compatible with any type of reward model. Experimental results demonstrate the superiority of our method across a range of scenarios with different types of rewards.

1 Introduction

In recent years, Large Language Models (LLMs) have made significant strides in the field of natural language processing, leading to their widespread use in downstream applications such as conversational agents (Brown et al., 2020; OpenAI, 2023; Touvron et al., 2023), code generation (Ahmad et al., 2021; Wang et al., 2021; Roziere et al., 2023), and machine translation (Wang et al., 2023; Moslem et al., 2023). Reinforcement learning from human feedback (RLHF) (Christiano et al., 2017; Ziegler et al., 2019; Ouyang et al., 2022; Rafailov et al., 2023) plays a critical role in this evolution, enhancing the models' ability to generate outputs that better align with human preferences and greatly increasing their versatility.

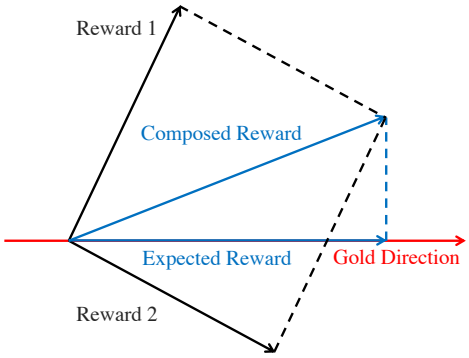


Figure 1: An example of the expected reward with two kinds of rewards with different preference direction.

Typically, RLHF involves three main steps. The first is supervised fine-tuning, where a foundational language model is refined using a targeted dataset. The second step involves training reward models that act as proxies for human preferences. Finally, the language model is optimized through a reinforcement learning algorithm (Schulman et al., 2017), wherein the language model functions as a policy model. It is evident that the reward model is crucial in ensuring the language model's outputs continually improves and adapts to human evaluative standards, which in turn, directly impacts the efficacy of the reinforcement learning phase.

One of the primary challenges identified in recent studies (Gao et al., 2023; Zhai et al., 2023; Yuan et al., 2023; Moskovitz et al., 2023) is overoptimization. This refers to the phenomenon where maximizing returns on the reward model beyond a certain threshold actually diminishes the performance of the policy model. Two main factors contribute to this problem. Firstly, the reward model is merely an approximation of human preferences, based on a limited dataset. Consequently, it is

prone to overconfidence when encountering out-of-distribution (OOD) data. Secondly, the reward model is tailored to optimize a specific aspect of success, which may not be in line with the ideal optimization path. Overly aggressive optimization steps with high rewards can cause the policy to deviate from this desired direction. Recent literature (Ouyang et al., 2022; Touvron et al., 2023; Zhai et al., 2023) often mitigating this problem via adding Kullback–Leibler (KL) penalties to prevent the model from diverging significantly from the reference model.

Another challenge arises in more complex scenarios involving multiple reward models (Ramamurthy et al., 2022; Glaese et al., 2022; Yuan et al., 2023; Bakker et al., 2022; Moskovitz et al., 2023). Here, employing a variety of reward models is advantageous as it allows the policy model to be assessed from diverse perspectives, enriching the evaluation process. Nonetheless, a consequent issue is that the policy model may become overly reliant on one or a select few reward models, achieving high returns during training while overlooking other relevant reward models. Therefore, devising an effective algorithm that integrates diverse reward models is essential. Recent studies have explored various methods for combining rewards, such as ranking (Yuan et al., 2023), weighting (Wu et al., 2023), employing welfare functions (Bakker et al., 2022), and implementing safe reinforcement learning (Moskovitz et al., 2023). Despite the intricate design of these approaches and the need for fine-tuning hyperparameters, they often lead to only incremental enhancements in performance.

To tackle the challenges outlined above, our research seeks a straightforward yet effective strategy to incorporate diverse rewards and combat overoptimization. Our research builds upon two key observations: firstly, different reward models provide insights into text quality from numerous angles, yet they are not entirely independent of each other. Secondly, language models struggle to discern the optimization direction of each individual reward model as they only receive a singular, aggregated reward score. We hypothesize the existence of an "gold direction" for optimization. The core concept of our approach is to calculate an expected reward that encapsulates overall progress towards this "gold direction." Figure 1 depicts an illustration with two reward models: we posit that the "gold direction" for optimizing the policy model is represented by the red line. When presented with two types of rewards, the expected reward should correspond to the projection of the composite rewards onto this "gold direction." Nevertheless, the "gold direction" and the optimization path for each reward model are not explicitly known; we only have access to their values. This constraint renders the direct calculation of the expected reward impracticable. To overcome this obstacle, we propose a straightforward and effective self-supervised method to train a model for composing rewards. Additionally, by penalizing the combined reward, we can approximate the expected reward while simultaneously mitigating the risk of overoptimization.

Our contributions are fourfold: (1) We propose an self-supervised method for training a model which can compose the rewards from different aspects. (2) By penalizing the composite reward with KL divergence, our method can not only approximate the expected reward but also simultaneously mitigating the issue of overoptimization (3) Our method is both simple and potent, easily adaptable to various reward models without incurring significant computational costs. (4) We validate our approach through experimental results across several scenarios involving diverse rewards, demonstrating its effectiveness.

2 Preliminaries

2.1 Environments: generation as MDP

For each NLP task, we are given a dataset $D = \{(x^i, y^i)\}_{i=1}^N$ where N denotes the number of examples, $x \in \mathcal{X}$ denotes the prompt inputs and $y \in \mathcal{Y}$ denotes the target outputs. Generation task can be viewed as a Markov Decision Process (MDP) (Puterman, 2014) which can be depicted as a tuple $\mathcal{M} \triangleq (\mathcal{S}, \mathcal{A}, \mathcal{R}, P, \gamma, T)$ with a finite vocabulary $\mathcal{V}$. At the beginning of each episode, a datapoint (x, y) is sampled from the data buffer, and the episode ends when the time step exceeds the maximum length horizon length T or an end of sentence (EOS) token is generated. The prompt input $x = (x_0, x_1, ..., x_m)$ is used as the initial state $s_0 = (x_0, x_1, ..., x_m)$, where $s_0 \in \mathcal{S}$, $x_m \in \mathcal{V}$ and $\mathcal{S}$ represents the state space. At time step $t < T$, the policy model $\pi(a_t|s_t)$ selects an action $a_t \in \mathcal{A}$ conditioned on its current state s_t, and then a new sate is reached via the transition function $P : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$. Each episode can be summarized as a trajectory $\tau = (s_i, a_0, ..., s_T, a_T)$, and the goal of the policy model is to maximize the

expected return $R(\tau) = \sum_{t=0}^{T} \gamma^t \mathcal{R}(s_t, a_t)$, where the $\mathcal{R} \in \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ denotes the reward function and $\gamma \in [0, 1)$ denotes the discount factor.

2.2 Reward model for optimizing the policy

The reward models can be broadly classified into two categories. The first category (Bakker et al., 2022; Yuan et al., 2023; Wu et al., 2023; Rafailov et al., 2023) includes pretrained models that serve as proxies for human preferences within specific contexts. This is because having humans evaluate utterances and interact with the environment during the optimization process can be costly and inconvenient. The second category (Ramamurthy et al., 2022; Moskovitz et al., 2023) encompasses commonly used metrics in natural language processing (NLP) such as BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), and METEOR (Banerjee and Lavie, 2005). These metrics allow for automatic measurement and quick implementation. In addition to these categories, rewards can be either coarse-grained or fine-grained. Coarse-grained rewards offer a single, sparse reward at the termination of each episode, while fine-grained rewards are accessible at any given time step. In scenarios where different types of rewards coexist, we define the composite reward function as follows:

$$r_{com} = f(r_1, ..., r_n), \tag{1}$$

where r_i denotes the output reward from various reward models, n is the number of reward models involved, and $f(\cdot)$ represents any composite function. For simplicity, the time-step subscript t is omitted here and will continue to be excluded in the remainder of the text.

3 Method

This section is dedicated to addressing two pivotal questions: (1) What constitutes the expected reward? (2) How can we effectively approximate this reward using a neural network?

3.1 Expected Singular for Multi-preference

This paper relies on two main assumptions:

- Each reward model predominantly assesses text quality from a singular perspective, and these perspectives are interrelated rather than isolated.
- There exists an "gold direction" for optimizing the language model.

Referencing the example depicted in Figure 1, it is evident that rewards derived from disparate reward models can be conceptualized as vectors within a Euclidean plane. It logically follows that the aggregate reward vector, which encapsulates improvements in both aspects, should correspond to the diagonal within the parallelogram defined by these two vectors. If we assign the x-axis as the "gold direction" for policy optimization, our task then is to determine the magnitude of the diagonal's projection onto this "gold direction". Nonetheless, the inherent challenge lies in the fact that the exact orientations of the reward models, as well as the "gold direction", remain unknown, rendering the calculation of the expected reward intrinsically unfeasible.

3.2 Estimating the expected reward

Theory of the composite reward. Let $\mathcal{U}$ be a vector space of dimension n over the field $\mathbb{R}$. Under our first assumption, the rewards $r_1, \ldots, r_n$ can be projected into $\mathcal{U}$ such that $u_1, \ldots, u_n$ constitute a basis of $\mathcal{U}$, with the composite reward vector u_{com} also lying within this space. This implies that each vector u_i is not contained within the span of the remaining basis vectors, i.e., $u_i \notin \text{span}\{u_1, \ldots, u_{i-1}, u_{i+1}, \ldots, u_n\}$ for all i. The assumption is considered to be mild because each reward metric evaluates lexical quality in a distinct and significant manner. For instance, it is highly unlikely for BLEU (Papineni et al., 2002) to be represented as a linear combination of ROUGE (Lin, 2004) and METEOR (Banerjee and Lavie, 2005).

Our aim is to determine the composite reward r_{com}, which is subsequently used to calculate the expected reward r_{exp}. We posit that a set of scalars $k_0, \ldots, k_n$ exists, corresponding to each individual reward, which collectively fulfill the following relationship:

$$u_{com} = \sum_{i=1}^{n} k_i u_i. \tag{2}$$

Considering the inner product $\langle u_{com}, u_i \rangle$, which reflects the relationship between r_{com} and each r_i, and applying the Gram–Schmidt process (Leon et al., 2013), we can derive an orthogonal basis r' with $r_i \in r'$. This orthogonal basis allows us to state that:

$$\langle u_{com}, u_i \rangle = k_i ||u_i||^2 = ||u_{com}|| \cdot ||u_i|| \cos \alpha, \tag{3}$$

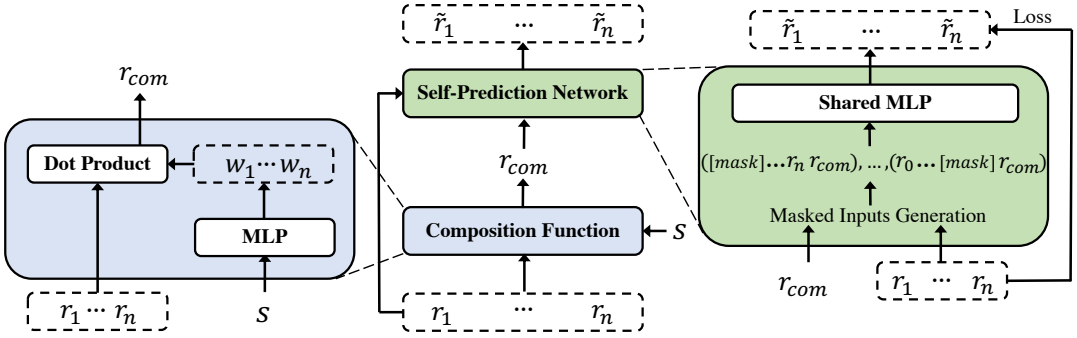


Figure 2: The composition model for integrating different kinds of rewards. The blue box represents the composition network which takes the state and the rewards as inputs and output the composite reward. The green box represent the module for self supervised learning and will not be used when optimizing the policy model.

where α is the angle between r_i and r_{com}.

Although the exact "angle" information in Equations (2) and (3) remains unknown, these equations guide us to construct meaningful relationships between the composite and individual rewards. Equation (2) indicates that the composite reward can be expressed as **a weighted sum of individual rewards**, while Equation (3) inspires us to **define specific relationships** between the composite reward and individual rewards through a neural network. Hence, we have designed neural network models, as depicted in Figure 2, to derive the composite reward.

Composing the rewards. First, we construct a neural network to calculate the weights for each of the rewards, which can be expressed as follows:

$$w = \sigma(f_w(\texttt{repr}_s)), \tag{4}$$

where $f_w(\cdot)$ represents a trainable neural network, and $w = w_1, \ldots, w_n$ denotes the set of weights for the n reward models, $\texttt{repr}_s$ denotes the representation of the s. Subsequently, we compute the composite reward r_{com} using the formula:

$$r_{com} = f(r_1, ..., r_n) = \sum_{i=1}^{n} w_i \cdot r_i. \tag{5}$$

Establishing specific relationships via self-supervised learning. Next, we design another neural network that predicts each individual reward based on the other rewards and the composite reward. This method is analogous to the "mask and predict" technique used in the pretraining of language models. Specifically, we first generate n tuples of inputs $\texttt{input}_r = ([mask], \ldots, r_n, r_{com}), \ldots, (r_1, \ldots, [mask], r_{com})$, and then predict the masked reward using the equation:

$$\tilde{r} = f_r(\texttt{input}_r), \tag{6}$$

where $f_r(\cdot)$ denotes another trainable neural network, $\tilde{r} = \tilde{r}_1, \ldots, \tilde{r}_n$ represents the set of predicted rewards, and $\tilde{r}_i$ is the predicted value for the i-th reward.

The composition network is trained using the following loss function:

$$\mathcal{L}_r = \frac{1}{n} \sum_{i=1}^{n} (\tilde{r}_i - r_i)^2. \tag{7}$$

It is important to note that the reward composition network is trained in advance of the policy model's fine-tuning process. Once the fine-tuning phase for the policy model begins, the parameters of the reward composition network remain fixed. Additionally, during this phase, the self-prediction network is not employed.

Approximating expected reward and mitigating overoptimization. We postulate that the expected reward r_{exp}, in alignment with the "goal direction", can be estimated by incorporating the composite reward with a KL penalty, as expressed in the following equation:

$$r_{exp} \approx r_{com} - \beta \cdot \mathrm{KL}(\pi_\theta(a|s) \parallel \pi_{ref}(a|s)), \tag{8}$$

where π_θ represents the policy model, π_{ref} signifies the reference model, and β is the coefficient determining the magnitude of the KL penalty. Meanwhile, this penalization also help to curb the overoptimization tendencies of the policy model (Ramamurthy et al., 2022; Moskovitz et al., 2023; Wu

et al., 2023). The expected reward can be employed to fine-tune the language model using any reinforcement learning algorithm. In this paper, we opt for Proximal Policy Optimization (PPO) (Schulman et al., 2017), the specifics of which will be elaborated upon in section 4.

Analysis. Equation (5) can be interpreted as an alternative form of Equation (2), wherin the computation reflects the composition relationship in the mapped vector space. Empirically, it is evident that each weight w_i lies within the open interval (0,1), and so we add the sigmoid function $\sigma(\cdot)$ to constrain the weights to lie within this range. Furthermore, by minimizing the loss described in Equation (7), we explicitly define a certain relationship among the rewards $(r_1, \ldots, r_n, r_{com})$. This is achieved by employing a shared neural network that effectively captures the interrelationships among the composite and individual rewards, functioning as a practical realization of the conceptual relationship depicted in Equation (3).

4 Experiment

We evaluate the effectiveness of our method on different scenarios with different language models. In alignment with Moskovitz et al. (2023) and Wu et al. (2023), we carry out experiments on dialogue generation and question answering tasks. Details concerning hyperparameters and training details are provided in the Appendix.

4.1 Dialogue Generation

4.1.1 Experimental Settings

Dataset. We conducted an experiment using a widely recognized dataset called DailyDialog (Li et al., 2017), which consists of transcripts of conversations between humans.

Reward models. For the rewards, we selected METEOR (Banerjee and Lavie, 2005), Intent score (Ramamurthy et al., 2022), BLEU (Papineni et al., 2002), and Bert score (Zhang et al., 2019), as they capture the desired behavior of the text from different perspectives. Among these, Intent score and Bert score are estimated via from a pretrained human preference model, RoBERTa (Liu et al., 2019) and BERT (Devlin et al., 2018), the other two are n-gram metrics. The reward is coarse-grained, with each response receiving a single reward that reflects the quality of the entire utterance.

Baselines. GPT2 (Radford et al., 2019) is used as the initial policy model. We selected Proximal Policy Optimization (PPO) (Schulman et al., 2017) as our baseline algorithm, wherein the rewards are computed as a linear combination of individual metrics, each with a predetermined fixed weight. Additionally, we employed Constrained Reinforcement Learning from Human Feedback (Constrained RLHF) (Moskovitz et al., 2023) as a comparative baseline.

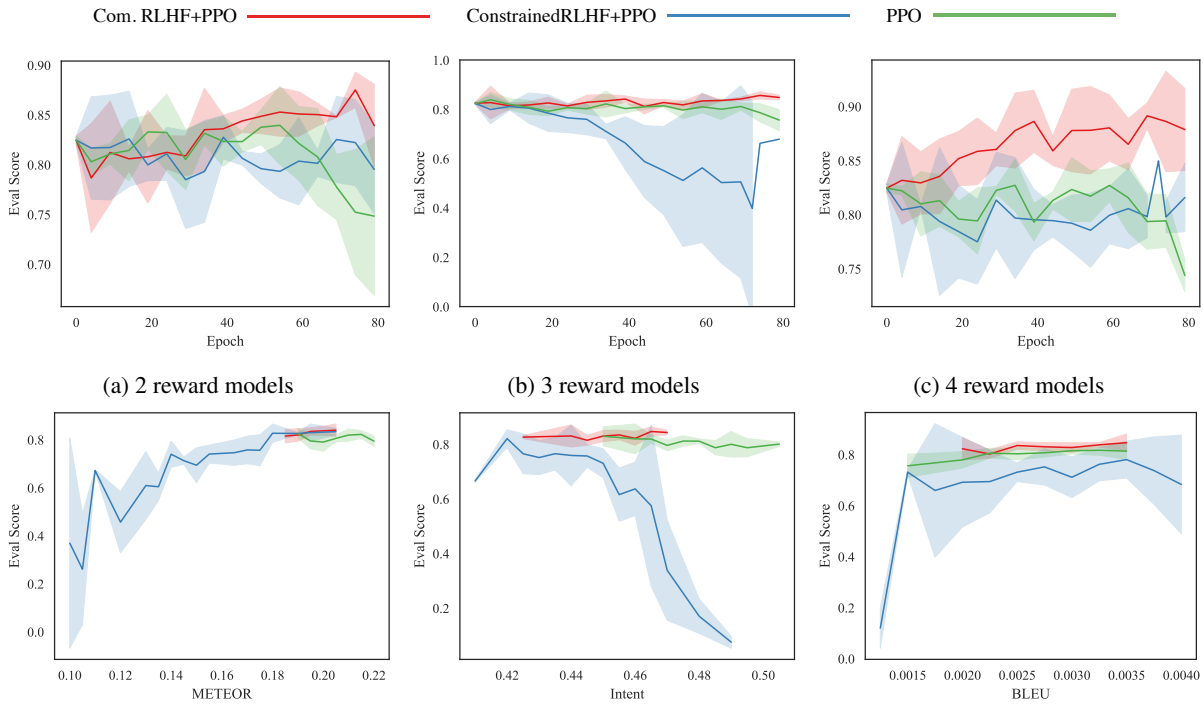
Evaluation metrics. To automatically evaluate each method, we adopted a similar way with Moskovitz et al. (2023), which involves calculating an evaluation score based on six distinct metrics. The chosen metrics, published by Moskovitz et al. (2023), are external to the reward model. Specifically, we utilized SacreBLEU (m_a) (Post, 2018), ROUGE-2 (m_b) (Lin, 2004; Ganesan, 2018), and ROUGE-L (m_c) as metrics related to lexicon, and Conditional Entropy-3 (m_u), vocab-size-3-nopunct (m_v), and mean-prediction-length-nopunct (m_w) as metrics related to diversity. In a manner akin to that of Moskovitz et al. (2023), we normalized the score of each metric to fall within a range of 0 to 1, using the minimum and maximum values observed in Constrained RLHF experiments across three distinct reward model settings. The evaluation score (m_{eval}) is subsequently computed as outlined in Equation (9)

$$m_{eval} = \frac{m_a + m_b + m_c + m_u + m_v + m_w}{6} \tag{9}$$

To complement our automated assessment, we also carried out a human evaluation to gain further insights into the performance of the models. This additional step allowed us to capture subjective quality aspects and nuances that automated metrics might not fully reflect, providing a more comprehensive understanding of the models' capabilities in generating realistic and coherent dialogue.

4.1.2 Experimental Results

Stablility across varying numbers of reward. We carried out experiments utilizing configurations with 2, 3, and 4 reward models. Figure 3(a-c) depicts the improvement of model performance over training epochs, where the results are the mean of three random seeds, and the shaded area indicates the standard deviation. We observed that all methods remained stable when only two reward models, METEOR and Intent Score, were



(a) 2 reward models (b) 3 reward models (c) 4 reward models

(d) 3 reward models (e) 3 reward models (f) 3 reward models

Figure 3: The evaluation score of different methods on three scenarios with different number of rewards.

utilized. However, for the ConstrainedRLHF, performance significantly deteriorated and training became unstable upon the incorporation of a third metric, BLEU. In contrast, the baseline PPO and our method demonstrated stability. As illustrated in the figure, our method outperforms the others in these settings, thereby demonstrating the effectiveness of the reward composition model. The strong baseline, ConstrainedRLHF, delivered unsatisfactory performance except in the two-reward setting, which can be primarily attributed to its explicit requirement for rewards from each aspect to surpass certain thresholds, without considering potential conflicts among them. Consequently, the type and number of reward models employed will significantly influence its performance.

Overoptimization and reward conflict phenomena. Nonetheless, two phenomena require attention. Firstly, the performance of the language model tends to decline after approximately 75 epochs, which may be due to the fact that KL regularization, despite mitigating optimization, cannot completely eliminate it. Consequently, there is a tendency for the policy to overfit on the reward models. Secondly, the peak performance obtained with three reward models is lower than that achieved with two, a result that may stem from the potential conflict between differing objectives, impairing further improvement.

Table 1: Human evaluation results on DailyDialog.

Method	Selection Rate
PPO	40%
Com. RLHF+PPO(Ours)	**64%**
ConstrainedRLHF+PPO	12%
No preference	13%

Improving the policy model comprehensively. Figure 3(d-f) illustrates how the evaluation score varies with different metrics in the 3-reward-model setting. It shows that each metric score of our method experiences less fluctuation over the course of training, ultimately achieving the highest evaluation score. This highlights that the composite reward is both useful and appropriate for facilitating the optimization of the policy.

Human evaluation results. We randomly sampled 50 dialogue contexts from the dataset along with their generated responses for human evaluation. To assess these, we recruited 20 human evaluators and tasked them with choosing the most appropriate response for each context through anonymous questionnaires. We allowed for multiple se-

lections, providing evaluators the option to indicate "no preference" when they encountered difficulty in discerning a clear favorite or if none of the responses seemed fitting. As depicted in Table 1, the outcomes of our human evaluation are in agreement with those from the automated assessment, confirming that our method significantly outperforms the competing approaches in terms of performance.

4.2 Question answering

4.2.1 Experimental Settings

Dataset. We conduct experiment on QA-Feedback dataset provided by Wu et al. (2023), consisting of 3,853 training examples, 500 development examples, and 948 test examples.

Reward models. In this dataset the rewards are fine-grained which means they are granted after several timesteps, pertinent to each subsentence. Three reward models were trained on human-labeled data, focusing on three distinct categories: relevance, correctness, and completeness.

Baseline. Following Wu et al. (2023), we selected the T5-large (Raffel et al., 2020) model that had been fine-tuned on 1,000 training examples (referred to as SFT) as baseline, which also served as the initial policy model. Concurrently, we compared our method to Fine-grained RLHF (F.G. RLHF) (Wu et al., 2023), an approach that amalgamates different rewards using fixed weights predefined by experts.

4.2.2 Experimental Results

Resolving conflicts among competing rewards The evaluation results for the test dataset are presented in Table 2, where R_1, R_2, and R_3 denote relevance reward, factuality reward, and completeness reward, respectively. Compared with baseline methods, our approach achieves the maximum reward in almost all aspects, with the exception of factuality. This can be attributed to the inherent conflicts among these reward models, which makes it challenging to optimize them simultaneously (please refer to Appendix for more details). Specifically, the model RLHF achieves the highest R_1 score, owing to the higher predefined weight assigned to it.

Human evaluation results. Similar with Wu et al. (2023), we randomly selected 50 test examples and enlisted 20 individuals to conduct a human evaluation to compare our method with F.G RLHF.

Table 2: Results on QA-Feedback test set.

Method	Rouge	R_1	R_2	R_3
SFT	49.16	0.469	0.793	0.225
F.G. RLHF	50.16	0.518	0.823	0.226
Com. RLHF	**50.18**	**0.526**	0.798	**0.245**

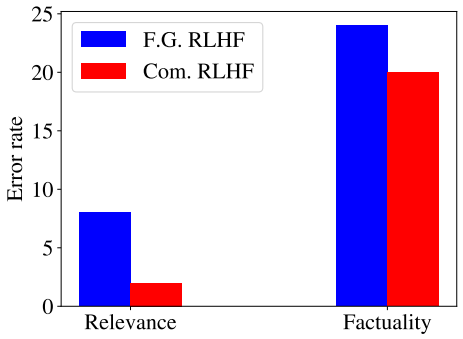


Figure 4: Human evaluation results on relevance and factuality.

Each evaluator was charged with assessing the responses from each model for (1) irrelevance and (2) incorrectness, Furthermore, workers were asked to compare the (3) completeness of the information provided in the responses from the different models. Evaluators also had the option to indicate their preferred response, with "hard to decide" as a permissible selection.

The evaluation results for irrelevance and incorrectness are illustrated in Figure 4. Responses from our method were found to be consistently more relevant to the questions asked. Moreover, our approach resulted in a relative lower rate of factual errors. The assessment of completeness and preference is presented in Table 3. Our method surpassed F.G RLHF in terms of completeness and matched it in terms of preference.

Table 3: Human pairwise comparison win rate on information completeness and their preference response on QA-Feedback test set.

Ours *vs.* F.G RLHF	Win	Tie	Lose
Completeness	44%	16%	40%
Preference	44%	12%	44%

5 Related Work

5.1 Reinforcement learning from human feedback

Reinforcement Learning from Human Feedback (RLHF) (Christiano et al., 2017; Ziegler et al.,

2019; Ouyang et al., 2022; Rafailov et al., 2023) has emerged as a pivotal technology in fine-tuning language models to better align with human intentions. It demonstrates its effectiveness in downstream tasks such as summarization (Stiennon et al., 2020), story-telling (Ziegler et al., 2019) instruction following, and harmlessness reducing (Bai et al., 2022; Lu et al., 2022; Ganguli et al.). Initially, reward models are trained to act as proxies for human preferences. Subsequently, the policy model is refined by maximizing cumulative returns through reinforcement learning algorithms like REINFORCE and Proximal Policy Optimization (PPO). Nonetheless, a primary challenge of RLHF is overoptimization (Gao et al., 2023), where inaccurate reward models may be overconfident in sample evaluations, leading to misguided policy updates and degraded performance. To counteract this issue, some researchers (Ouyang et al., 2022; Touvron et al., 2023) have introduced penalty terms to constrain the policy model from deviating excessively from a reference model, enhancing stability and reducing uncertainty. Other studies (Yuan et al., 2023; Rafailov et al., 2023; Song et al., 2023) have sought to circumvent the reward modeling process altogether, optimizing the policy directly. Despite these advancements, Li et al. (2023) show that reward-model-based approaches offer advantages when dealing with out-of-preference samples.

5.2 Combining different rewards

To enhance the language model's alignment with diverse preferences, various forms of feedback are typically utilized to reflect the policy's behavior across multiple dimensions (Bakker et al., 2022; Glaese et al., 2022; Yuan et al., 2023; Wu et al., 2023; Moskovitz et al., 2023). Nonetheless, integrating these rewards poses a challenge, as the policy might disproportionately emphasize one or a few specific reward models. Some studies (Wu et al., 2023; Ramamurthy et al., 2022) address this by summing the different rewards, assigning predefined weights based on prior knowledge. Alternatively, another body (Yuan et al., 2023; Glaese et al., 2022) of research suggests optimizing the agent's policy by ranking multiple sampled responses. More specifically, Yuan et al. (2023) introduced a ranking loss designed to elevate the probabilities of superior responses, while Glaese et al. (2022) suggested a reranking score to serve as the overall reward, providing a bonus to comparatively high-quality responses within the samples. Bakker et al. (2022) proposed a welfare function that quantifies and ranks consensus statements according to their attractiveness to the aggregate reward models. Moskovitz et al. (2023) adopted constraint reinforcement learning to deter the agent from excessively optimizing each reward model beyond set proxy points. However, the policy struggles to discern the intentions underlying the design of rewards and receives only a scalar value. Therefore, we investigate methods to yield an anticipated reward that holistically improves the language model. We propose a straightforward yet potent approach for training a reward composition model in a self-supervised way.

6 Conclusion

In this study, we concentrate on scenarios involving multi-faceted reward models to fine-tune large language models. We posit that various reward models assess text from distinct perspectives, converging towards an optimal "gold direction" for policy optimization. To amalgamate the rewards from diverse models, we introduce a straightforward yet potent reward composition model, which can be trained through a self-supervised manner. By imposing penalties on the composite reward, our approach not only aligns closely with the expected reward in the "gold direction" but also mitigates the issue of overoptimization. We validate our approach through a series of experiments utilizing assorted reward types, and the empirical evidence attests to the effectiveness of our method.

Limitations and Future Work. Our study, while demonstrating promising results, is subject to several limitations. Firstly, the effectiveness of our method hinges on an assumption that lacks formal mathematical guarantees. Secondly, the process of training a reward composition model incurs additional computational overhead. Lastly, given that the outputs from different reward models might conflict, our approach does not currently possess a mechanism to discriminate between more and less useful models, instead aggregating them to compute a composite reward. For future works, we aim to refine our methodology by establishing theoretical foundations to bolster its reliability. Additionally, exploring ways to obtain feedback from more powerful large language model such as GPT-4 could offer interesting avenues to enhance model performance.

References

Wasi Uddin Ahmad, Saikat Chakraborty, Baishakhi Ray, and Kai-Wei Chang. 2021. Unified pre-training for program understanding and generation. *arXiv preprint arXiv:2103.06333*.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. 2022. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*.

Michiel Bakker, Martin Chadwick, Hannah Sheahan, Michael Tessler, Lucy Campbell-Gillingham, Jan Balaguer, Nat McAleese, Amelia Glaese, John Aslanides, Matt Botvinick, et al. 2022. Fine-tuning language models to find agreement among humans with diverse preferences. *Advances in Neural Information Processing Systems*, 35:38176–38189.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

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.

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.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Kavita Ganesan. 2018. Rouge 2.0: Updated and improved measures for evaluation of summarization tasks. *arXiv preprint arXiv:1803.01937*.

D Ganguli, A Askell, N Schiefer, T Liao, K Lukošiūtė, A Chen, et al. The capacity for moral self-correction in large language models. arxiv 2023. *arXiv preprint arXiv:2302.07459*.

Leo Gao, John Schulman, and Jacob Hilton. 2023. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, pages 10835–10866. PMLR.

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*.

Steven J Leon, Åke Björck, and Walter Gander. 2013. Gram-schmidt orthogonalization: 100 years and more. *Numerical Linear Algebra with Applications*, 20(3):492–532.

Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017. Dailydialog: A manually labelled multi-turn dialogue dataset. *arXiv preprint arXiv:1710.03957*.

Ziniu Li, Tian Xu, Yushun Zhang, Yang Yu, Ruoyu Sun, and Zhi-Quan Luo. 2023. Remax: A simple, effective, and efficient method for aligning large language models. *arXiv preprint arXiv:2310.10505*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ximing Lu, Sean Welleck, Jack Hessel, Liwei Jiang, Lianhui Qin, Peter West, Prithviraj Ammanabrolu, and Yejin Choi. 2022. Quark: Controllable text generation with reinforced unlearning. *Advances in neural information processing systems*, 35:27591–27609.

Ted Moskovitz, Aaditya K Singh, DJ Strouse, Tuomas Sandholm, Ruslan Salakhutdinov, Anca D Dragan, and Stephen McAleer. 2023. Confronting reward model overoptimization with constrained rlhf. *arXiv preprint arXiv:2310.04373*.

Yasmin Moslem, Rejwanul Haque, and Andy Way. 2023. Adaptive machine translation with large language models. *arXiv preprint arXiv:2301.13294*.

R OpenAI. 2023. Gpt-4 technical report. arxiv 2303.08774. *View in Article*, 2:13.

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.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Matt Post. 2018. A call for clarity in reporting bleu scores. *arXiv preprint arXiv:1804.08771*.

Martin L Puterman. 2014. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint arXiv:2305.18290*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Rajkumar Ramamurthy, Prithviraj Ammanabrolu, Kianté Brantley, Jack Hessel, Rafet Sifa, Christian Bauckhage, Hannaneh Hajishirzi, and Yejin Choi. 2022. Is reinforcement learning (not) for natural language processing?: Benchmarks, baselines, and building blocks for natural language policy optimization. *arXiv preprint arXiv:2210.01241*.

Baptiste Roziere, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Tal Remez, Jérémy Rapin, et al. 2023. Code llama: Open foundation models for code. *arXiv preprint arXiv:2308.12950*.

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

Feifan Song, Bowen Yu, Minghao Li, Haiyang Yu, Fei Huang, Yongbin Li, and Houfeng Wang. 2023. Preference ranking optimization for human alignment. *arXiv preprint arXiv:2306.17492*.

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.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Longyue Wang, Chenyang Lyu, Tianbo Ji, Zhirui Zhang, Dian Yu, Shuming Shi, and Zhaopeng Tu. 2023. Document-level machine translation with large language models. *arXiv preprint arXiv:2304.02210*.

Yue Wang, Weishi Wang, Shafiq Joty, and Steven CH Hoi. 2021. Codet5: Identifier-aware unified pre-trained encoder-decoder models for code understanding and generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8696–8708.

Zeqiu Wu, Yushi Hu, Weijia Shi, Nouha Dziri, Alane Suhr, Prithviraj Ammanabrolu, Noah A Smith, Mari Ostendorf, and Hannaneh Hajishirzi. 2023. Fine-grained human feedback gives better rewards for language model training. *arXiv preprint arXiv:2306.01693*.

Zheng Yuan, Hongyi Yuan, Chuanqi Tan, Wei Wang, Songfang Huang, and Fei Huang. 2023. Rrhf: Rank responses to align language models with human feedback without tears. *arXiv preprint arXiv:2304.05302*.

Yuanzhao Zhai, Han Zhang, Yu Lei, Yue Yu, Kele Xu, Dawei Feng, Bo Ding, and Huaimin Wang. 2023. Uncertainty-penalized reinforcement learning from human feedback with diverse reward lora ensembles. *arXiv preprint arXiv:2401.00243*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

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 Experimental Details

A.1 Experimental Settings

Dialogue generation. We adopted a similar experimental setup as Ramamurthy et al. (2022); Moskovitz et al. (2023) for our dialogue generation, utilizing a context window with a span of five utterances. This segmentation approach produced a dataset comprising 35,000 utterances for training, 3,000 for validation, and 3,000 for testing. Echoing Moskovitz et al. (2023), our decoding process employed a top-k sampling strategy with k set to 20. Inputs to the model were presented as concatenated segments of human dialogue, with speaker transitions denoted by a distinct end-of-utterance (<EOU>) token. Additionally, the intent classification reward mechanism are established based on a fine-tuned RoBERTa (Liu et al., 2019) model. This system assigned a score of 1 when the model's inferred intent for a generated utterance matched that of the corresponding reference or ground-truth utterance, otherwise attributing a score of 0. Prior to their introduction into the composition model, we normalized each reward model by subtracting the mean value computed from the training set. In addition, we harnessed the fine-tuned RoBERTa model to extract representations of the input text, subsequently utilizing these representations as the state variable s in Equation (4). Given the coarse-grained nature of the rewards, these representations were calculated as the temporal mean across the entire text. Consistent with (Moskovitz et al., 2023) study, we adopted the GPT2 (Radford et al., 2019) architecture for both the policy and value models. We select four distinct rewards to conduct experiments, the specifics of which are detailed in Table 4.

Table 4: Chosen rewards in dialogue generation task

Setting	Chosen metric or model
2 rewards	METEOR; INTENT
3 rewards	METEOR; INTENT; BLEU
4 rewards	METEOR; INTENT; BLEU; BERT

Question Answering. In the question answering scenario, we similarly employ a top-k sampling approach, setting k to 20. Diverging from our previous task, we have opted for T5-large (Raffel et al., 2020) as the policy model and T5-base for the value model. Owing to the fine-grained nature of the reward, we obtain text representations using the T5-large reference model at each time step, which then act as the state variable s in Equation (4). The training process for each reward model, aimed at aligning with human preferences, follows the methodology outlined by (Wu et al., 2023). For an in-depth understanding, readers are directed to their original publication.

Details for training reward composition model. Each individual reward is normalized before being fed into the reward composition network. For the preference-based reward model, we apply z-normalization. For the other reward models, we initially scale them to a range between 0 and 1, and then subtract their mean value. Regarding the self-prediction network, it comprises two dense layers, both shared across all masked tuples. The first layer features 32 units with ReLU activation, functioning as the common encoder, while the second layer has a single unit, establishing the specific relationship between the tuple and the masked value.

Training algorithm. The comprehensive training protocol we adopted is encapsulated in Algorithm 1. This framework adheres to the standard Proximal Policy Optimization (PPO) algorithm (Schulman et al., 2017), augmented with an additional step dedicated to the calculation of the expected reward.

Table 5: Hyperparameters for finetuning policy model

Settings	DailyDialog	QA-Feedback
Total epochs	80	10
Batch size	64	12
Learning rate	1e-6	1e-5
Clip ratio ϵ	0.2	0.2
Rollouts top-k	20	20
Temperature	0.7	0.7
Discount factor γ	0.99	0.99
GAE λ	0.95	0.95
KL coefficient β	0.2	0.3
Policy model	GPT2	T5-large
Value model	GPT2	T5-base
Model for $\texttt{repr}_s$	RoBERTa	T5-large

Hyperparameters. One of the strengths of our method is that it does not introduce additional hyperparameters beyond those required by the baseline PPO algorithm. Furthermore, all weights within the composite reward model are learned through a neural network architecture. For transparency and reproducibility, we have detailed all

Algorithm 1 Optimizing a Language Model with Multiple Reward Models

Initialize: reference language model π_{ref}; initial value model V_φ; n reward models $\mathcal{R}_1, ..., \mathcal{R}_n$; task dataset D; hyperparameters

Finetune the reference language model on dataset D and get the initial policy model π_θ
Training the reward models $\mathcal{R}_1, ..., \mathcal{R}_n$ on dataset D
Training the composition reward model f on dataset D
for epoch $ep = 1, ..., k$ **do**
    Sample a batch D_b from D
    Sample output sequence $y^i \sim \pi_\theta(\cdot|x^i)$ for each $x^i \in D_b$
    Compute rewards $r_1, ..., r_n$ via $\mathcal{R}_1, ..., \mathcal{R}_n$.
    Compute composite rewards r_{com} via Equation (4) and Equation (5)
    Compute expected rewards r_{exp} via Equation (8)
    Compute advantages $\{A\}_{t=1}^{|y^i|}$ and target values $\{V'\}_{t=1}^{|y^i|}$ for each y^i with V_φ
    Update the policy model by:
    $\theta \leftarrow \arg\max_\theta \frac{1}{|D_b|} \sum_{i=1}^{D_b} \frac{1}{|y^i|} \sum_{t=1}^{y_i} \min(\frac{\pi_\theta(a_t|s_t)}{\pi_{ref}(a_t|s_t)} A_t, \text{clip}(\frac{\pi_\theta(a_t|s_t)}{\pi_{ref}(a_t|s_t)}, 1-\epsilon, 1+\epsilon) A_t)$
    Update the policy model by:
    $\varphi \leftarrow \arg\min_\varphi \frac{1}{|D_b|} \sum_{i=1}^{D_b} \frac{1}{|y^i|} \sum_{t=1}^{y_i} (V_\varphi(a_t|s_t) - V'(a_t|s_t))^2$
end for

Output: π_θ

Table 6: Hyperparameters for training composition reward model

Settings	DailyDialog	QA-Feedback
Total epochs	2	2
Batch size	256	128
Initial learning rate	5e-5	1e-4
Optimizer	Adam	Adam

the hyperparameters associated with fine-tuning the policy and training the composite reward model in Table 5 and Table 6, respectively.

Computational resources. All of our experiments were conducted on a single NVIDIA A100 GPU. For the dialogue generation task, the optimization of the language model typically required between 8 and 10 hours. Additionally, the training of the composition network incurred an extra 2 to 4 hours of computational time. For the question answering task, the optimization of the language model required between 25 and 30 hours, and the training of the reward composition network incurred an extra 1 to 3 hours of computational time.

A.2 More Experimental Results

Visualizing the conflicting rewards. To more effectively analyze the correlations among the various rewards in the question answering task, we plotted the reward values at each time step and fitted a polynomial surface to the data, as shown in Figure 5. It is evident that the reward for relevance conflicts with the other two types of rewards, complicating the optimization of the policy.

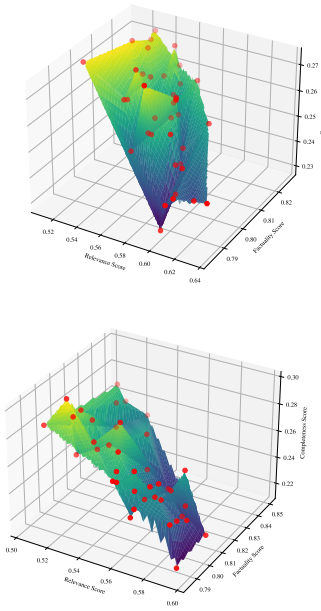


Figure 5: Correlations among rewards. Upper F.G. RLHF, lower Com. RLHF.

Showcases We present examples of the questionnaires and showcase the generated responses for two tasks in Table 7, Table 9, Table 8 and Table 10.

Table 7: Example of the questionnaire for DailyDialog

Context (<EOU> denotes the end of an utterance and signals a switch in speaker roles.)	Hello . This is Mrs . Wilson . I'd like to buy a new car . Could you offer me a new type of the car , please ? <EOU> Oh , Madam , buy what ? <EOU> A new car . And I've got a small family , two children , and I haven't got a lot of money and ... <EOU> Oh , oh , Madam , Madam , I'm afraid you have the wrong number . <EOU> Isn't this the car store ? <EOU>
Response A	I have no idea. You have just given me your number <EOU>
Response B	It's raining. It's raining a lot that is just that.<EOU>
Response C	No, it's the car wash. <EOU>
Please chose your preference (Multiple selections are allowed and D for "no suitable response or hard to decide")	

Table 8: Generation sample of DailyDialog

Sample 1	Prompt	A dessert sounds nice . <EOU> What kind are you thinking of getting ? <EOU> I want to get some pie . <EOU> What kind of pie do you want ? <EOU> I have no idea . <EOU>
	Reference	Do you want to know what kind of pie I like ? <EOU>
	PPO	Pie?? You mean like you had a whole bunch on your plate. <EOU>
	ConstrainedRLHF+PPO	I have no idea. <EOU>
	ComposedRLHF+PPO	Do you have a favorite? <EOU>
Sample 2	Prompt	You're made a good choice . This china tea set is unusual . <EOU> Where was it from ? <EOU> It was made in Jingdezhen . <EOU> Jingdezhen ? Isn't it called ' the capital of porcelain ' ? <EOU> You are right . <EOU>
	Reference	This is the very thing I've been dreaming of . <EOU>
	PPO	It was made in Shaoi, China. <EOU>
	ConstrainedRLHF+PPO	I'm not sure why you think that's so. <EOU>
	ComposedRLHF+PPO	So why is it called Jingdezhen? <EOU>

Table 9: Example of the questionnaire for QA-Feedback

Question	When does spring break start in the us?
Context	wikipage: Spring break text: In Jamaica, the spring break starts in the first week of Good Friday. The break may range from one week to two weeks, often times two. This break starts depending on which month the Easter holiday begins, March or April. In Mexico, spring break takes place during the Holy Week and the one after it. In the United States, spring break at the college and university level can occur from March to April, depending on term dates and when Easter holiday falls. Usually, spring break is about one week long, but many K–12 institutions in the United States schedule a two-week-long break known as "Easter Break," "Easter Holidays", or "Easter Vacation", as they generally take place in the weeks before or after Easter. However, in the states of Massachusetts and Maine, schools typically schedule spring break for the week of the third Monday in April to coincide with Patriots' Day. In Guatemala and Honduras, it takes place during Easter; schools give students a whole week to rest while the staff workforce rests approximately three days. In Colombia, spring break takes place the first week of April, during the Holy Week until the second week.
Answer A	In the United States, spring break at the college and university level can occur from March to April, depending on term dates and when Easter holiday falls. However, in the states of Massachusetts and Maine, schools typically schedule spring break for the week of the third Monday in April to coincide with Patriots' Day. In Jamaica, the spring break starts in the first week of Good Friday.
Answer B	In the United States, spring break at the college and university level can occur from March to April, depending on term dates and when Easter holiday falls. Usually, spring break is about one week long, but many K–12 institutions in the United States schedule a two-week-long break known as "Easter Break," "Easter Holidays", or "Easter Vacation", as they generally take place in the weeks before or after Easter. However, in the states of Massachusetts and Maine, schools typically schedule spring break for the week of the third Monday in April to coincide with Patriots' Day.
Please Select	Is answer A relevant to the question? Is answer B relevant to the question? Is there any factual error in answer A? Is there any factual error in answer B? Which information is more complete, answer A or B? (C for "hard to decide") Which answer do you prefer, answer A or B? (C for "hard to decide")

Table 10: Generation sample of QA-Feedback

Question	Who did meg's voice first on family guy?
Context	wikipage: Meg Griffin text: Meg was voiced by an uncredited Lacey Chabert for the first season, and by Mila Kunis in subsequent seasons after Chabert became busy with school and appearing on "Party of Five" at the time, although some of her work became second season episodes due to production order. Mila Kunis won the role after auditions and a slight rewrite of the character, in part due to her performance on "That '70s Show". MacFarlane called Kunis back after her first audition, instructing her to speak more slowly, and then told her to come back another time and enunciate more. Once she claimed that she had it under control, MacFarlane hired her. MacFarlane stated that Kunis "had a very natural quality to Meg" and she's "in a lot of ways [...] almost more right for the character". Kunis's voice is first heard as Meg in Episode 3 of season two "Da Boom", and the voices switch back and forth in the broadcast order until settling on Kunis. Tara Strong provides Meg's singing voice in "Don't Make Me Over". Meg is very unpopular in high school due to both her plain appearance and personality. She desperately tries to be part of the cool crowd, but is usually coldly rebuffed. | wikipage: Meg Griffin text: Meg Griffin Megan "Meg" Griffin is a fictional character in the animated television series "Family Guy". Meg is the eldest child of Peter and Lois Griffin and older sister of Stewie and Chris, but is also the family's scapegoat who receives the least of their attention and bears the brunt of their abuse. She is often bullied, ridiculed, and ignored. Meg first appeared on television, along with the rest of the Griffin family, in a 15-minute short on December 20, 1998. She was created and designed by "Family Guy" creator Seth MacFarlane, who was asked to pitch a pilot to the Fox Broadcasting Company, based on "The Life of Larry" and "Larry & Steve", two shorts made by MacFarlane featuring a middle-aged man named Larry and an intellectual dog, Steve. After the pilot was given the greenlight, the Griffin family appeared in the episode "Death Has a Shadow". Originally voiced by Lacey Chabert during the first season, Meg has been voiced by Mila Kunis since season 2. Meg is a self-conscious and insecure adolescent girl.
F.G. RLHF	Meg Griffin is a fictional character in the animated television series Family Guy. She is the eldest child of Peter and Lois Griffin and older sister of Stewie and Chris, but is also the family's scapegoat who receives the least of their attention and bears the brunt of their abuse. Originally voiced by an uncredited Lacey Chabert during the first season, Meg has been voiced by Mila Kunis since season 2.
Com. RLHF	Meg Griffin is a fictional character in the animated television series Family Guy. She was voiced by an uncredited Lacey Chabert for the first season, and by Mila Kunis in subsequent seasons after Chabert became busy with school and appearing on "Party of Five" at the time. Kunis's voice is first heard as Meg in Episode 3 of season two "Da Boom", and the voices switch back and forth in the broadcast order until settling on Kunis. Tara Strong provides Meg's singing voice in "Don't Make Me Over".