

PEERING THROUGH PREFERENCES: UNRAVELING FEEDBACK ACQUISITION FOR ALIGNING LARGE LANGUAGE MODELS

Hritik Bansal, John Dang, Aditya Grover

Department of Computer Science, University of California Los Angeles
 {hbansal, john.dang, adityag}@cs.ucla.edu

ABSTRACT

Aligning large language models (LLMs) with human values and intents critically involves the use of human or AI feedback. While dense feedback annotations are expensive to acquire and integrate, sparse feedback presents a structural design choice between ratings (e.g., score Response A on a scale of 1-7) and rankings (e.g., is Response A better than Response B?). In this work, we analyze the effect of this design choice for the alignment and evaluation of LLMs. We uncover an *inconsistency problem* wherein the preferences inferred from ratings and rankings significantly disagree 60% for both human and AI annotators. Our subsequent analysis identifies various facets of annotator biases that explain this phenomena, such as human annotators would rate denser responses higher while preferring accuracy during pairwise judgments. To our surprise, we also observe that the choice of feedback protocol also has a significant effect on the evaluation of aligned LLMs. In particular, we find that LLMs that leverage rankings data for alignment (say model X) are preferred over those that leverage ratings data (say model Y), with a rank-based evaluation protocol (is X/Y’s response better than reference response?) but not with a rating-based evaluation protocol (score Rank X/Y’s response on a scale of 1-7). Our findings thus shed light on critical gaps in methods for evaluating the real-world utility of language models and their strong dependence on the feedback protocol used for alignment. Our code and data are available at https://github.com/Hritikbansal/sparse_feedback.

1 INTRODUCTION

Recently, alignment has emerged as a critical step for next-generation text-based assistants [37]. Specifically, its goal is to align large language models (LLMs) with human values and intents, making their generated content accurate, coherent, and harmless when responding to input queries [29, 2, 4, 40]. The process of model alignment involves three main components: feedback acquisition, where humans (or AI) assess the quality of the base model’s responses; alignment algorithms, which adjust the skills of the base model based on the feedback data; and model evaluation, which assesses the performance of the aligned model on a wide range of novel user instructions [21]. Prior work has primarily focused on designing alignment algorithms, such as PPO, DPO, and PRO [34, 26, 32, 36, 50, 22], under specific feedback protocols and evaluation setups. Additionally, previous research on feedback acquisition [8, 48, 33] has focused on developing fine-grained and dense feedback protocols for aligning LLMs; however, these protocols are challenging and expensive to acquire. Within the sparse feedback protocols that are easy to acquire, there is a structural design choice between *ratings* and *rankings*, and its impact on the alignment pipeline is still under-explored.

To this end, we analyze the effect of the two feedback protocols: ratings and rankings on the LLM alignment and further evaluation. Specifically, the rating protocol is an absolute form of feedback in which the annotator assigns a rating to a response from the base LLM using a predefined scale (e.g., a 1-7 Likert scale). In contrast, the ranking protocol is a relative form of feedback in which the annotator selects their preferred response from a pair of candidate responses generated by the base LLM. The ratings on the model responses quantifies their goodness which enables model builders to gauge their strengths and weaknesses, but the ratings are hard to determine for complex instructions

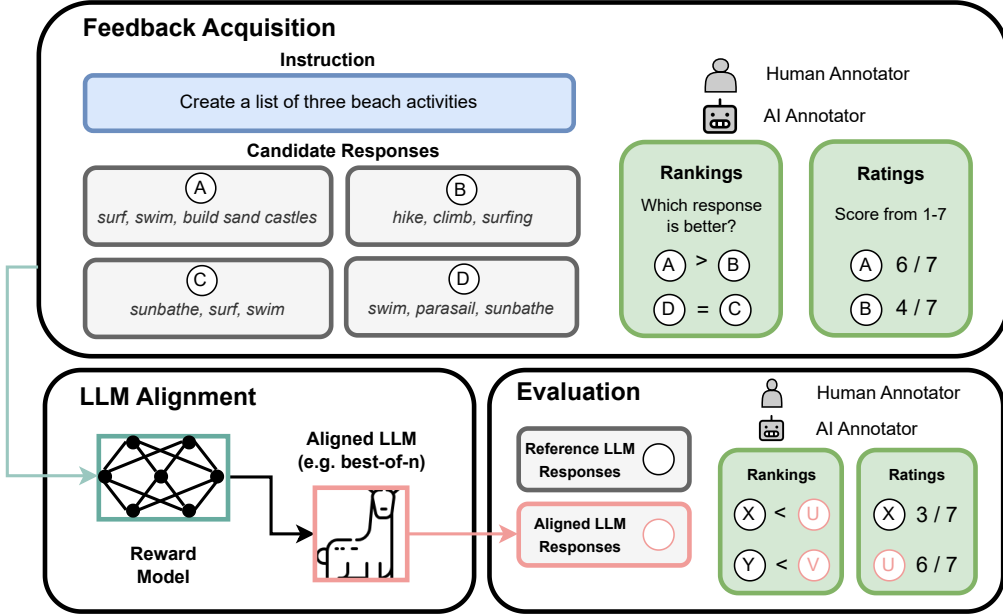


Figure 1: Overview of our pipeline to study the effect of the choice between sparse feedback protocols (ratings and rankings) for the alignment and evaluation of LLMs. First, we sample multiple responses from the LLM for the queries in the instructions dataset. Then, we acquire rankings and rating feedback data from the human and AI annotators, independently. Subsequently, the feedback data is used to train the reward models for the Best-of-n policy. Finally, we compute the win-rate against the reference model under ratings and rankings feedback protocol from humans and AI.

(poem ‘Roses and Lake in Shakespearean style’) [22, 35]. On the other hand, the rankings protocol is easy to acquire for complex instructions but does not quantify the gap between the pair of responses [37, 29]. Due to these unique behaviors, both the ratings and rankings feedback protocols are compelling subjects for our study.

In our work, we first explore the interplay between these feedback protocols. To do so, we collect ratings for individual responses from annotators, both human and AI, and transform them into their ranking form by comparing the assigned ratings (e.g., if Response A is rated 5 and Response B is rated 4, this implies that Response A is ranked higher than Response B). Additionally, we gather pairwise ranking preferences for the same responses (e.g., comparing Response A vs. Response B) from annotators, independently of their ratings. Surprisingly, we uncover a *feedback inconsistency problem* and find that a significant portion of all preference comparisons between ratings and rankings disagree with each other for humans 58% and AI annotator (GPT-3.5-Turbo) 59.4% (§4). We observe that responses receiving inconsistent feedback are actually perceived as *closer* to each other than those receiving consistent feedback. Qualitatively, we trace this problem back to the differences in how annotators assess various facets of response quality (§4.3). Our analysis provides insights for making decisions regarding feedback acquisition protocols for aligning LLMs.

After receiving feedback, our objective is to investigate the impact of the protocols on training reward models for model alignment and subsequent model evaluation. To achieve this, we train reward models (§2.3 on the ratings and rankings data. We employ the Best-of-n policy where the trained reward models are then employed to select the best response from a set of n candidate responses generated by the Alpaca-7B model. Subsequently, we assess the quality of the Best-of-n policies (both ratings and rankings) by evaluating their responses against DaVinci-003 [29]. Additionally, we vary the choice of feedback acquisition for model evaluation, considering both ratings for model responses and pairwise rankings between Best-of-n responses and DaVinci-003. Figure 1 provides an illustration of our pipeline.

To our surprise, we observe an *evaluation inconsistency*, a phenomenon where the choice of the feedback protocol (rankings) for model evaluation favors responses from the Best-of-n policy that utilizes the same feedback protocol (rankings), and vice versa. Additionally, we have found that feedback inconsistency occurs for both human evaluators and GPT-3.5-Turbo (ChatGPT) as response evaluators. Specifically, we have noted that the win-rate gap between the Best-of-n (rankings) policy and the base LLM is larger (by 11.2%) compared to the gap between the Best-of-n (ratings) policy and the base LLM (5.3%) when using human rankings for evaluation (see §5.3). However, the win-rate gap between the Best-of-n (ratings) policy and the base LLM (5%) is only slightly larger than the gap between the Best-of-n (rankings) policy and the base LLM (4%) using human ratings for evaluation.

Our contributions are as follows:

1. We study the effect of the choice between sparse feedback protocols, ratings (absolute scores), and rankings (pairwise preferences) for the alignment and evaluation of LLMs.
2. We uncover a feedback inconsistency problem where the ratings on the pair of responses disagree with the ranking between them for 60% of comparisons for both humans and AI.
3. We further analyze various facets of perceived response quality by the annotators while providing different forms of feedback.
4. Finally, we find that the choice of the feedback protocol has a sharp influence on the evaluation of the aligned LLMs in the form of evaluation inconsistency.

2 BACKGROUND

Language modeling is a task of learning a distribution p_θ over a sequence of words or tokens given the context of previous words i.e., $\mathbf{w} = \{w_1, w_2, \dots, w_n\}$ such that the $p_\theta(\mathbf{w}) = \prod_{i=1}^n p_\theta(w_i | w_{<i})$ [18, 5]. The large language models (LLMs) are trained on a large corpus of text data from the web [31, 7, 54, 16, 39]. Post-training, they can generate sentences (sequence of words) conditioned on a context, which can be used to solve complex natural language processing (NLP) tasks without specialized finetuning [47, 42]. Subsequently, [46, 43] perform supervised finetuning to teach the pretrained language model about a wide range of novel tasks, which further enable them to understand and perform well on unseen tasks [42]. However, [37] argue that maximizing for an automatic evaluation (e.g., ROUGE [23]) through supervised finetuning would not generate a high-quality summary of passages, but require optimizing for human preferences directly. Eventually, the LLM would generate content that aligns well with facets of human intents and values (e.g., helpfulness, harmlessness). Since then, there have been various works on aligning large language models to understand tasks in the wild [29, 4, 40] or challenging tasks e.g., math problems [22].

In this work, we focus on aligning an LLM to generate high-quality outputs, that are considered accurate, coherent and harmless by humans, for unseen instructions. The initial step in the alignment process is to equip the base LLM with the ability to understand human or machine-generated input queries or instructions. This is accomplished through supervised fine-tuning (SFT), where the base LLM is fine-tuned with pairs of human-written or machine-generated instructions and corresponding responses. Notably, there have been substantial advancements in constructing instruction fine-tuning datasets and models recently [38, 55, 45, 43, 30, 49, 14, 51, 12, 44].

Following SFT, the model generates candidate responses for new instructions (§ 2.1), and feedback data is acquired under a protocol (ratings or rankings) from the annotators (§ 2.2). Further, we employ an alignment algorithm (Best-of-n) that trains reward models on the ratings or rankings feedback data independently (§ 2.3).

2.1 INSTRUCTION-RESPONSE DATA COLLECTION

Consider a language model p_θ that can understand instructions and respond to them. We consider a set of instructions $\mathcal{X} = \{x_1, \dots, x_n\}$ where n is the number of instructions. Subsequently, we generate a set of k candidate responses $\{y_i^{(1)}, \dots, y_i^{(k)}\}$ for every instruction x_i by sampling from the model’s distribution $y_i^{(j)} \sim p_\theta(\cdot | x_i)$. Next, we collect the feedback data on the pre-defined instructions and the generated candidate responses $\mathcal{D} = \{(x_i, \{y_i^{(1)}, \dots, y_i^{(k)}\})\}$.

2.2 FEEDBACK DATA

Ratings Feedback Protocol With the given instruction and candidate response data \mathcal{D} , the objective of ratings feedback acquisition is to assign an ordinal value to each individual response independently. In our study, we request the annotators to rate the responses on a Likert scale (1 - 7). We instruct the annotators to evaluate the responses based on the response quality considering factors such as helpfulness, harmlessness, and coherence of the response in addition to their subjective judgment.

Formally, the annotators assign an absolute score $a(x_i, y_i^{(j)}) \in \{1, 2, \dots, 7\}$ where x_i and $y_i^{(j)}$ are instruction and a generated candidate response, respectively. The annotation process would thus create a feedback data $\mathcal{D}_A = \{(x_i, y_i^{(j)}, a(x_i, y_i^{(j)}))\}$ where A represents the ratings protocol.

Rankings Feedback Protocol Given the instruction and candidate response data \mathcal{D} , the aim of rankings protocol is to assign a *preferred* response (or choose ‘equal’) for every pair of candidate responses to a given instruction. To that end, we first transform the instruction-response data into instruction-*pairwise* response data by creating pairwise combinations of the responses i.e., $\mathcal{D}' = \{(x_i, y_i^{(j)}, y_i^{(\ell)})\}$ where $i \neq \ell$. Identical to the absolute feedback, the annotators are instructed to use the various quality axes and their subjective judgment for decision-making.

Formally, the annotators assign a relative feedback $r(x_i, y_i^{(j)}, y_i^{(\ell)}) \in \{\text{‘response 1’}, \text{‘response 2’}, \text{‘equal’}\}$ where ‘response 1’ indicates that $y_{i,j}$ is preferred over $y_i^{(\ell)}$. The annotation process would thus create a feedback data $\mathcal{D}_R = \{(x_i, y_i^{(j)}, y_i^{(\ell)}, r(x_i, y_i^{(j)}, y_i^{(\ell)}))\}$ where the subscript R represents the rankings feedback protocol.

2.3 REWARD MODELING

Here, we describe the training objectives for the reward models trained on ratings and rankings feedback.

Regression Reward Model. Here, we first normalize the ratings (between 1-7) $a(x_i, y_i^{(j)})$ for a given instruction in the dataset \mathcal{D}_A into score $a'(x_i, y_i^{(j)})$ between 0-1. Subsequently, we train a regression model $f_\theta(x_i, y_i^{(j)})$ where $(x_i, y_i^{(j)}) \in \mathcal{D}_A$ which outputs a scalar. The regression reward model is trained with the following objective:

$$\mathcal{L}_A(\mathcal{D}_A) = \mathbb{E}_{(x_i, y_i^{(j)}) \sim \mathcal{D}_A} [(\sigma(f_\theta(x_i, y_i^{(j)})) - a'(x_i, y_i^{(j)}))^2] \quad (1)$$

where $\sigma(\cdot)$ is the sigmoid function which projects any scalar value to a range between 0-1.

Negative Log-Likelihood (NLL) Reward Model. To train a reward model from the pairwise feedback data \mathcal{D}_R , prior works [29] train a reward model $g_\theta(x, y)$ which outputs a scalar value for a given response y for the instruction x . Firstly, we filter the instances that receive $r(x_i, y_i^{(j)}, y_i^{(\ell)}) = \text{‘equal’}$ from \mathcal{D}_R to get a new subset of data \mathcal{S}_R since one cannot learn from such examples in the relative reward modeling objective. For the remaining instances, the negative log-likelihood objective might be applied as follows:

$$\mathcal{L}_R(\mathcal{S}_R) = -\mathbb{E}_{(x, y_a, y_b) \sim \mathcal{S}_R} [\log \sigma(g_\theta(x_i, y_a) - g_\theta(x_i, y_b))] \quad (2)$$

where y_a and y_b are the preferred and unpreferred responses respectively from the pair of candidate responses $(y_i^{(j)}, y_i^{(\ell)})$ as decided by the value of $r(x_i, y_i^{(j)}, y_i^{(\ell)})$ in the dataset \mathcal{S}_R .

We provide more details on the choice of f_θ , g_θ , and the training setup in §5.1.

Best-of-n Policy. In our work, we use the Best-of-n policy (rejection sampling) \mathcal{P}_n that leverages the trained reward model to boost the performance of the SFT model towards human preferences. In this method, we simply sample n times from the SFT model from a given instruction, and utilize the

reward models (regression or NLL) to score the set of n responses. The final output of the policy is the response that achieves the highest score under the trained reward model. Formally the Best-of- n policy that leverages the ratings reward model f_θ ,

$$\mathcal{P}_n(f_\theta, p_\theta, x_i, \{y_i^{(1)}, \dots, y_i^{(n)}\}) = y_i^{(m)} \quad (3)$$

where $m = \operatorname{argmax}_j f_\theta(x_i, y_i^{(j)})$, x_i is the input instructions, and $y_i^{(j)}$ is j^{th} response from the n candidate responses sampled from the SFT model p_θ . We emphasize that our work is focused on understanding the impact of various forms of feedback, and their interplay, on the downstream performance reward modeling. Here, our approach can use any other RL algorithm such as PPO used for model alignment [29] or a mix of rejection sampling and PPO [40]. We focus on rejection sampling due to its simplicity, stability, and robust performance in comparison to other RL algorithms as noted by prior works [26, 53].

3 FEEDBACK DATA ACQUISITION

Here, we describe the feedback data acquisition process. We start by collecting responses to a large set of instructions from Alpaca-7B (§3.1). Subsequently, we describe feedback acquisition from AI (§3.2) which is large-scale and thus used for training the reward models. Further, we describe feedback acquisition from humans in §3.3 followed by meticulous feedback data analysis §3.4.

3.1 INSTRUCTION-RESPONSE DATA

The instructions are designed to present a collection of innovative queries that could potentially be posed to a text-based AI assistant in real-world scenarios.

In total, we collect 5.2K instructions from varied sources:

1. **Dolly** [12]: It contains 15K high-quality human-generated prompts and response pairs. We select a subset of 4K prompts randomly.
2. **Self-Instruct (User Oriented)** [45]: It is a set of 252 expert-written instructions.
3. **Super-NI** [43]: Originally, it contains 1600+ NLP tasks with their expert-written task-descriptions. In our work, we select a subset of 100 dense NLP tasks such as ‘question generation based on a given scenario’ instead of the ones that require just ‘yes/no’ answer (Appendix §B). Subsequently, we randomly select 10 instances for every task.

In our work, we use *Alpaca-7B* [38], which is constructed by instruction-tuning the base LLaMA-7B [39] LLM on 52K instruction-response data, as our base LLM. Thus, we prompt Alpaca to generate **five** candidate responses for a range of diverse instructions. We set the max length for each generated response to 128. The overall composition of the instructions is presented in Table 1.

Instructions Composition	
# of instructions from Dolly	4K
# of instructions from Self-Instruct (User Oriented)	252
# of instructions from Super-NI	1K
# of generations per instruction	5
Feedback Composition (GPT-3.5-Turbo)	
# of instances w/ Ratings	24.6K
# of instances w/ Pairwise Rankings	46.5K
Feedback Composition (Humans)	
# of instances w/ Ratings	4K
# of instances w/ Pairwise Rankings	2K

Table 1: Feedback data statistics. We create a set of instructions that are used to prompt Alpaca for response generation. Subsequently, we acquire feedback on those responses from humans and AI.

3.2 FEEDBACK FROM AI

Prior works [13] demonstrate that the existing AI systems (LLMs) such as GPT-4 [28] can be leveraged for providing pairwise feedback data and improve over the base LLM. In addition, [55] makes a case for LLM as a substitute for human preferences in a scalable and reliable way. In our work, we collect large-scale ratings and rankings feedback data consisting of 71K instances from an **GPT-3.5-Turbo** (ChatGPT). We chose ChatGPT since it is $20\times$ cheaper than GPT-4 and was more easily accessible at the time of the project. [15] showed that GPT-3.5-Turbo outperforms crowd-workers for text-annotations tasks.

To collect ratings feedback, we prompt GPT-3.5-Turbo to assign a score between 1-7 to the individual candidate responses for the instructions, collected in §3.1, independent of each other. The model is specifically prompted to assign the ratings score after evaluating the response for its accuracy, coherence, and harmlessness, in addition to its subjective judgment. We further clarify that a rating of 1 implies a low quality response whereas a rating of 7 indicates a high quality response. Similar instructions were provided to the human annotators in §3.3. In total, we collect 24.6K instances of ratings feedback and spend \$150 in the acquisition.

To collect the rankings feedback, we prompt the GPT-3.5-Turbo LLM with the instruction and a pair of candidate responses ('response 1', 'response 2'), and command it to provide its preference between the two responses. To mitigate any potential bias in 'response 1' and 'response 2' ordering [55], we run two queries for all instances covering both the possible orderings. If the model preferences flip by flipping the order of the responses, then we consider it as tie situation and assign an 'equal' feedback to the pair of responses, as done in [6]. In total, we collect 46.5K unique instances of rankings feedback, excluding the two possible orderings of the pair of candidate responses, and spend \$600 in the acquisition. We will provide more analysis on the feedback data in §3.4.

3.3 FEEDBACK FROM HUMANS

We also collect feedback data under the ratings and rankings protocols from the humans 6K instances of annotations. Such data serves a multitude of purposes, notably: (a) providing insights into the behavior of different forms of feedback collected from humans, (b) facilitating a comparative analysis of the similarities and dissimilarities between feedback patterns derived from AI systems and those obtained from human participants, and (c) enabling a comprehensive assessment of the agreement between the feedback data generated by AI systems and the feedback provided by human participants. In our work, we emphasize that the human feedback data is not used for training the reward models, and hence model alignment, due to its small scale which is costly to expand.

Firstly, we randomly select a subset of 500 from 46.5K from the GPT-3.5-Turbo rankings feedback data. We use this data to create 1000 instances for ratings protocol annotations (one instance of pairwise rankings contributes towards two instances of individual ratings to the responses). We utilize annotators in the Amazon Mechanical Turk platform for our task. Before the qualifications, the annotators are shown a few solved examples for both protocols. Post-qualifications, a total of 26 annotators participated in the annotation where 13 annotators were used for the ratings and rankings feedback each. The annotators were paid an hourly wage of \$18/hour. The overall cost for human annotations was \$1000. We assign each instance of the feedback data to **four** workers. In summary, we have 2000 rankings feedback ($= 500 \text{ instances} \times 4 \text{ workers per instance}$) and 4000 ratings ($= 1000 \text{ instances} \times 4 \text{ workers per instance}$) from human annotators.

3.4 DATA ANALYSIS

Ratings Distribution. We present the score distribution for the 4K ratings data from the human annotations and their corresponding GPT-3.5-Turbo annotations in Figure 2. We find that the majority of the responses ($\sim 80\%$) achieve better than average ratings i.e., a score > 4 from the humans and the AI. This indicates that the quality of candidate responses from Alpaca-7B is good which can be attributed to its instruction-following capabilities. In addition, we find that the human annotators tend to give a perfect score of 7 to a large proportion of the responses, whereas GPT-3.5-Turbo assigns the perfect score to less than $< 5\%$ cases. We also observe that the majority of responses achieve an ratings score of 5 or 6 from GPT-3.5, and the distribution flattens to either side of these

scores. Our observation of the skewness in the ratings score distribution from AI systems is corroborated by Figure 2 in [30].

Feedback data is unbiased towards longer and unique responses. Prior works [55, 44] have shown that LLMs may favor longer and verbose responses, to varying degrees, despite being inaccurate and of lower quality. To this end, we assess whether the length or number of unique words in the candidate responses bias the ratings and rankings feedback judgments from humans and AI in our data. We present the results for such assessment in the ratings and rankings feedback in Appendix Table 6 and 7 respectively. In Table 6, we find that the ratings scores assigned to the individual responses do not increase with the average length and the average number of unique words in the response for both humans and AI. Similarly in Table 7, we find that there is no discernible difference between the average length and average number of unique tokens of the preferred and unpreferred response in the rankings feedback collected from the humans and AI. This highlights that our feedback data is unbiased towards longer and unique responses, and the differences observed from the prior works might be attributed to differences in the experimental setups.

Agreement Analysis. Here, we conduct an agreement analysis comparing the assessments provided by human-human and human-AI annotators for both ratings and rankings feedback. Specifically, we have collected 1000 instances with ratings feedback and 500 instances with rankings feedback from four human annotators and GPT-3.5-Turbo. To establish the ground truth for each instance, we consider the first three human annotations as the gold label, and we use the fourth annotation as the human prediction label. This arrangement allows us to calculate the human-human agreement. For instances with ratings feedback, we compute the average of the scores given by the three human annotators and round it off to the nearest integer and consider it as the gold label. For instance, with rankings feedback, we determine the final label based on the majority vote among the three human annotations. The possible choices for human feedback responses are {‘response 1’, ‘response 2’, ‘equal’}. If an instance receives two votes for ‘equal’ and one vote for ‘response 1’, we assign it a gold label of ‘equal’. In the event of a tie among the three choices, where each option receives one vote, we randomly sample the gold label from the possible choices.

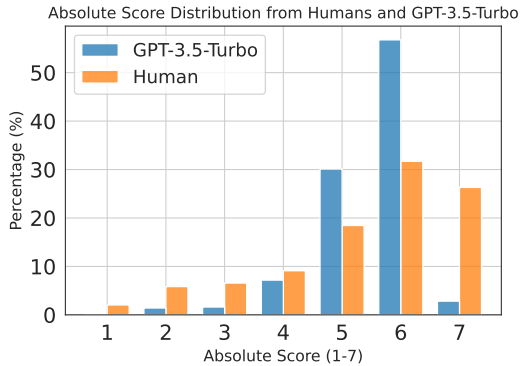


Figure 2: We present the ratings (ratings) score distribution acquired from humans and AI.

Feedback	Human-Human	Human-GPT-3.5	Number of examples
Ratings	1.08	0.9	1000
Rankings	62.7%	60.5%	500

Table 2: Agreement analysis between the gold human annotations and the held-out human annotations (Column 1) for ratings and rankings feedback. Similarly, we perform the agreement analysis for the gold human annotations and annotations from GPT-3.5-Turbo.

We calculate the average ratings difference between the gold label feedback and human (and AI) feedback to quantify the ratings agreement. Whereas, we calculate the percentage of instances (out of 500) where the gold label and human (and AI) feedback agrees with each other to quantify the rankings agreement. Specifically, we assign a score of 1 if the gold label matches the prediction, and a score of 0.5 if only one of the gold label or prediction assigns ‘equal’ to the pair of responses. We present our results in Table 2. In the ratings feedback (Row 1), we find that the average ratings difference between human-human ratings (1.08) is close to the human-GPT-3.5-Turbo ratings (0.9). Additionally, we observe that the human-human agreement and human-GPT-3.5-Turbo agreement is 62.7% and 60.5% respectively in the rankings feedback. We observe that our agreement rates are close to the 60% human-human agreement rates reported in the prior works [21]. In summary, our

agreement results indicate that GPT-3.5-Turbo can provide ratings and rankings feedback close to the human’s gold label for the responses to the instructions in our dataset.

4 FEEDBACK INCONSISTENCY PROBLEM

4.1 CALCULATING CONSISTENCY

Here, we make use of the observation that the ratings on a pair of responses for a given instruction can be compared to convert the ratings feedback data into its rankings form. Formally, we can convert the absolute feedback data \mathcal{D}_A to $\mathcal{D}_A^R = \{x_i, y_i^{(j)}, y_i^{(\ell)}, h(a(x_i, y_i^{(j)}), a(x_i, y_i^{(\ell)}))\}$ where $h(a(x_i, y_i^{(j)}), a(x_i, y_i^{(\ell)})) \in \{\text{‘response 1’}, \text{‘response 2’}, \text{‘equal’}\}$ where ‘response 1’ indicates that $y_i^{(j)}$ is preferred over $y_i^{(\ell)}$. For example, an instance in the absolute feedback data with the instruction ‘Create a list of three beach activities and with two of the candidate responses ‘*surf, swim, build castles*’ and ‘*hike, climb, birding*’ achieving the absolute scores 5 and 2, respectively, can be converted to the relative feedback data where the response ‘*surf, swim, build castles*’ is *preferred* over the response ‘*hike, climb, birding*’.

This conversion of the ratings data \mathcal{D}_A to the rankings data \mathcal{D}_A^R allows us a unique opportunity to study the interplay between the absolute feedback \mathcal{D}_A and relative feedback \mathcal{D}_R collected from the annotators, independently. Here, we define the term **consistency** as the agreement between the ratings (converted to its rankings form) and the rankings received by a pair of responses to a given instruction independent of the ratings data. Formally,

$$C(x_i, y_i^{(j)}, y_i^{(\ell)}) = \begin{cases} 1 & r(x_i, y_i^{(j)}, y_i^{(\ell)}) = h(x_i, y_i^{(j)}, y_i^{(\ell)}) \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

where $(x_i, y_i^{(j)}, y_i^{(\ell)}) \in \mathcal{S}$ and $\mathcal{S} = \mathcal{D}_A^R \cap \mathcal{D}_R$. Using equation 4, we can calculate the consistency for all the instances in the data \mathcal{S} . We will provide the results for the consistency analysis in §4.

We emphasize that the concept of ‘consistency’ can be subject to various interpretations within the existing literature. For instance, it has been explored in terms of intransitivity in human preferences [41] and (dis-)agreements arising among different annotators when providing feedback on a single entity. Nevertheless, in the context of our study, we operationalize consistency specifically concerning inter-feedback agreements. Such analysis grows in significance when making crucial design decisions related to the selection of feedback protocols within alignment pipelines.

4.2 RESULTS

We present the results for the consistency across 42K comparisons in the feedback data from GPT-3.5-Turbo (Table 3) and 500 comparisons in the feedback data from the humans (Table 4). In both tables, a particular cell (say row 1 and column 2) will indicate the percentage of total instances for which the ratings feedback considers the pair of candidate responses ‘equal’ while the rankings feedback prefers ‘response 1’ for the same pair of candidate responses.

We have observed a consistency issue in both human and AI feedback data. Interestingly, the consistency score falls within a similar range of 40% – 42% for both humans and AI, suggesting that a substantial portion of the feedback data can yield contradictory preferences depending on the feedback protocol employed. This consistency problem underscores several critical points: (a) it indicates variations in the perceived quality of responses based on the choice of the feedback acquisition protocols, (b) it underscores that the alignment pipeline can vary significantly depending on whether ratings or rankings are used as sparse forms of feedback, and (c) it emphasizes the necessity of meticulous data curation when working with multiple feedback protocols for aligning LLMs.

Hedging. We find that the GPT-3.5-Turbo tends to hedge its predictions way more than humans. Specifically, 47.1% rankings feedback and 57.4% ratings feedback are assigned an ‘equal’ preference by GPT-3.5-Turbo (sum of first row and column in Table 3). However, 30.9% rankings and 40.8% ratings feedback is assigned an ‘equal’ preference by the humans. We observe that the hedging percentages are higher for the ratings feedback as compared to the rankings feedback for both

		Rankings			Total
		Equal	Response 1	Response 2	
Ratings	Equal	27.7%	14.7%	15%	57.4%
	Response 1	9.7%	7.4%	4.5%	21.6%
	Response 2	9.7%	4.4%	6.9%	21.0%
	Total	47.1%	26.5%	26.4%	42%

Table 3: Results for the (dis-)agreements between feedback acquisition protocols (ratings and rankings) annotated by GPT-3.5-Turbo for 42K comparisons.

		rankings			Total
		Equal	Response 1	Response 2	
ratings	Equal	13.3%	12.4%	15.1%	40.8%
	Response 1	8.2%	13.1%	7.8%	29.1%
	Response 2	9.3%	6.4%	14.2%	30.0%
	Total	30.9%	32%	37.1%	40.6%

Table 4: Results for the (dis-)agreements between feedback acquisition protocols (ratings and rankings) annotated by humans for 500 comparisons. Here, each comparison is annotated by 4 workers.

humans and AI. This can be attributed to a large fraction of individual responses receiving a score of 5 or 6 from GPT-3.5-Turbo, as discussed in §3.4.

Decisive rankings feedback. We observe that whenever the rankings feedback, from humans and GPT-3.5-Turbo, is decisive (the feedback prefers one of the responses from the pair instead of selecting ‘equal’), the fraction of consistent assignments is higher than the inconsistent ones. Specifically, we focus on Column 2 and Column 3 of the tables. In Table 3, 7.4% (Column 2 and Row 2) is higher than 4.4% and 6.9% is higher than 4.5%. Similarly, in Table 4, 13.1% (Column 2 and Row 2) is higher than 6.4% and 14.2% is higher than 7.8%.

Decisive ratings feedback. We observe that whenever the ratings feedback converted to its rankings form is decisive, the fraction of consistent assignments is higher than the inconsistent ones. Specifically, we focus on Row 2 and Row 3 of the tables. In Table 3, 7.4% (Column 2 and Row 2) is higher than 4.5% and 6.9% is higher than 4.4%. Similarly, in Table 4, 13.1% (Column 2 and Row 2) is higher than 7.8% and 14.2% is higher than 6.4%.

Ideally, we would like to have no inconsistent instances between the forms of feedback i.e., the non-diagonal elements of the tables should be zero. Our results indicate that the current feedback data is far from the ideal scenario, however, it still exhibits some desirable behaviors including the fraction of consistent instances being more inconsistent ones when the feedback is decisive. Our observations lay the foundation for careful feedback data curation in model alignment.

4.3 FINE-GRAINED ANALYSIS OF CONSISTENT AND INCONSISTENT DATA

We perform fine-grained experiments to understand the difference between consistent and inconsistent subsets of the feedback data from humans and AI.

Difference in the Ratings. Our aim is to quantify the average ratings difference for the pair of candidate responses that receive a decisive preference (either of ‘response 1’ or ‘response 2’ but not ‘equal’) from the ratings feedback data. Specifically, we focus on Row 2 and Row 3 of Table 3 and 4. Within these rows, (a) the combination of the (Column 2 - Row 2, Column 3 - Row 3) represents the consistent instances where both forms of feedback are decisive, (b) the combination of the (Column 2 - Row 3, Column 3 - Row 2) represents the inconsistent decisive instances, and (c) Column 1 represents the scenario where the rankings feedback is indecisive. We report the average

ratings difference between the scores achieved by the instances belonging to the category (a), (b), and (c) in Table 5a.

We find that the maximum average ratings score difference is achieved by the **consistent** instances for human annotators as well as GPT-3.5-Turbo. This indicates that the quantitative gap between the pair of candidate responses belonging to the consistent instances is higher than the pair of responses belonging to the inconsistent instances, which further highlights that the difference in the quality of the two responses in a consistent instance is more distinguishable than in an inconsistent instance. In addition, we observe that the average ratings score difference for the inconsistent instances is similar when the rankings feedback is decisive and indecisive with ratings feedback being decisive (Column 2 and 3), for both human and AI annotators. This hints towards the potential qualitative closeness in the pair of responses that causes them to receive decisive ratings feedback but contradictory rankings feedback in the annotation process. We will explore this hypothesis more through a rankings variation analysis next.

	Consistent (Ratings & Rankings Decisive)	Inconsistent (Ratings & Rankings Decisive)	Inconsistent (Ratings Decisive)
Human	1.58	1.28	1.29
GPT-3.5-Turbo	1.22	1.12	1.12

(a) Difference in the ratings feedback scores when rankings feedback is decisive.

	Rankings variation score (Human Feedback)
Consistent Instances	0.50
Inconsistent Instances	0.36

(b) Variation in the rankings feedback for consistent and inconsistent examples.

Table 5: Quantitative analysis of the differences between the consistent and inconsistent subsets of the feedback data acquired from humans and AI.

Variation in the Rankings. In this study, our primary objective is to test the hypothesis that the inconsistent instances primarily result from the qualitative similarity between pairs of responses. To operationalize this, we propose that the rankings preference of human annotators will exhibit more significant variability for pairs of responses that are qualitatively close. Conversely, we expect the variation in human preferences to be lower when one of the responses is qualitatively superior. To conduct our analysis, we utilize a dataset comprising 500 rankings feedback instances, each with four human annotations. In the dataset, annotations favoring ‘response 1’ are marked as ‘+1’, those favoring ‘response 2’ are marked as ‘-1’, and annotations indicating ‘equal’ preference are marked as ‘0’. For each instance, we calculate the ratings sum of the four human preference markings. A low score on an instance suggests that the pair of candidate responses are considered ‘close’ by the annotators, leading to an equal likelihood of selecting ‘equal’ or either of the two responses. Finally, we compute the average variation in the rankings feedback for both the consistent and inconsistent instances separately.

The results of our analysis are presented in Table 5b. To quantify average variation in annotator feedback of a single example, we define a variation score metric. For each example, we take the sum of the score for each annotator’s label. Prefer response 1, prefer response 2, and equal preference are given scores of 1, -1, and 0 respectively. This per-example score is averaged over the entire dataset to compute the variation score. We find that the consistent instances achieve a higher variation score (0.5) as compared to the inconsistent instances in the human feedback data (0.36). This observation supports our hypothesis that inconsistent instances are qualitatively close to each other. The variation in the human and AI feedback for the inconsistent instances may arise due to their ‘intrinsic’ preference towards a specific dimension, such as succinctness, coherence, and writing style while evaluating the responses.

Qualitative Analysis. To investigate the source of the consistency problem, we conduct a thorough examination to determine whether it arises from systematic disparities in annotator preferences or from inexplicable noise within the feedback acquisition process. To this end, we ask the human annotators to provide explanations for a few feedback instances.

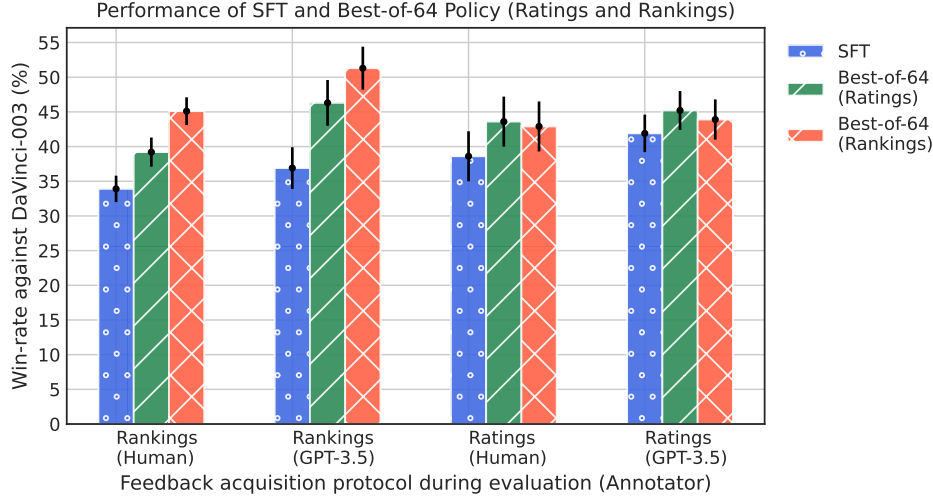


Figure 3: Win-rate against DaVinci-003 of Supervised finetuned (SFT) Alpaca-7B LLM and Best-of-64 policy that leverages the reward models trained with the ratings and rankings feedback protocols, independently, to select from the SFT responses. The gap between SFT and Best-of-64 (Rankings) is higher than SFT and Best-of-64 (Ratings) using the rankings feedback protocol for model evaluation. However, the gap between SFT and Best-of-64 (Ratings) is slightly higher than the gap between SFT and Best-of-64 (Rankings) using the ratings feedback protocol for model evaluation. The trend holds for humans and GPT-3.5-Turbo as the evaluators. The error bars represent the 95% confidence interval.

In Table 9, we showcase a pair of candidate responses, both expected to align with the provided instructions. The annotators responsible for rankings favored 'response 2' due to its density, while perceiving 'response 1' as dull. At the same, despite these divergent preferences, the individual responses received conflicting rating scores: 'response 2' incurred a substantial penalty for its density and perceived non-compliance with task instructions, whereas 'response 1' garnered an average score for only a partial adherence to the guidelines. This indicates that differences in the preferences of the humans while annotating for different feedback protocols played a significant role in their decision-making. We provide a few more examples for the consistent and inconsistent instances with their explanations in the Appendix §C.

5 ALIGNMENT AND EVALUATION

In the previous sections, we established that the feedback data obtained from both humans and AI suffer from a consistency problem. Nonetheless, it is crucial to examine the impact of these forms of feedback on language model alignment and subsequent model evaluation.

5.1 SETUP

Reward Modeling. We choose low-rank adaptation [17] (LoRA) of Alpaca-7B as the reward model.¹ We use the objective function defined in Eq. 2 to train the reward model on the rankings feedback data. Similar to [29], a single batch only contains all the pairs of candidate responses for a particular instruction. Further, 70% of the feedback data is used for training and the rest is for validation. We provide more details on the training settings in Appendix §D. We train the reward models on the rankings and ratings feedback for 5 epochs with early stopping where the validation loss is the criteria. We train each reward model for three random seeds.

Evaluation. Post-training, we evaluate the models on 553 unseen instructions, which includes 188 instructions from the Open Assistant (OASST) evaluation, 129 from the helpful evaluation

¹<https://github.com/tloen/alpaca-lora>

released by Anthropic [3], 80 from Vicuna evaluation [10], and 156 from Koala evaluation [14]. We employ a Best-of- n ($n = 64$) policy to evaluate the effect of various feedback data on model alignment. For this, we first generate n candidate responses of length 200 from the SFT model (Alpaca-7B) with the unseen instructions. Subsequently, the output of Best-of- n would be the one that achieves the maximum score with the trained reward model. Finally, we compute the win-rate of the SFT model and the Best-of- n policies against DaVinci-003 (reference model) on these unseen instructions. Here, we use the DaVinci-003 responses collected in AlpacaEval [21]. Here, we vary the choice of the feedback protocols as ratings and rankings during evaluation and the choice of the annotators as humans and ChatGPT-3.5-Turbo. Each pairwise comparison or response rating is performed by a crowd-worker, where we paid \$18 per hour for the annotations. We spent \$1000 on collecting evaluation annotations from humans.

5.2 COMPUTING WIN-RATES

Post-alignment, we evaluate the Best-of- n policies (ratings, rankings) against a reference model on a set of unseen user instructions \mathcal{U} . Since we aim to understand the effect of the choice of the feedback acquisition protocol on the complete alignment pipeline, we collect ratings and rankings for the responses from the alignment policy and the reference models from humans and AI as annotators. We compute the win-rate against the reference model as the fraction of instances for which the Best-of- n policy response is preferred over the reference model under a particular feedback protocol. In case of a ‘tie’ judgment by the annotators, both the policy and reference model get half a point. Formally, we define per-example score for rankings as

$$\mathcal{E}(u, y_{\mathcal{P}}, y_{\mathcal{F}}) = \begin{cases} 1 & r(u, y_{\mathcal{P}}, y_{\mathcal{F}}) = y_{\mathcal{P}} \\ 0.5 & r(u, y_{\mathcal{P}}, y_{\mathcal{F}}) = tie \\ 0 & r(u, y_{\mathcal{P}}, y_{\mathcal{F}}) = y_{\mathcal{F}} \end{cases} \quad (5)$$

and for ratings as

$$\mathcal{E}(u, y_{\mathcal{P}}, y_{\mathcal{F}}) = \begin{cases} 1 & a(u, y_{\mathcal{P}}) > a(u, y_{\mathcal{F}}) \\ 0.5 & a(u, y_{\mathcal{P}}) = a(u, y_{\mathcal{F}}) \\ 0 & a(u, y_{\mathcal{P}}) < a(u, y_{\mathcal{F}}) \end{cases} \quad (6)$$

where $u \in \mathcal{U}$ is the unseen input instruction, $y_{\mathcal{P}} = \mathcal{P}_n(g_{\theta}, p_{\theta}, u, \{v_1, \dots, v_n\})$ is output from the Best-of- n policy that leverages the rankings reward model, $y_{\mathcal{F}}$ is the response from the reference model, $r(u, y_{\mathcal{P}}, y_{\mathcal{F}})$ is the preference between the pair of responses decided by the annotator. Win-rate metric w is computed as the average per-example score:

$$w = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \mathcal{E}(u, y_{\mathcal{P}}, y_{\mathcal{F}}) \quad (7)$$

5.3 RESULTS

We train the reward models on the ratings feedback data and the rankings feedback data collected from GPT-3.5-Turbo (§3.2). We present the win-rates for the SFT and the Best-of- n policies on the unseen instructions evaluated by humans and GPT-3.5-Turbo in Figure 3.

Best-of- n policies outperforms SFT. We find that randomly sampling from the n ($= 64$) candidate responses from the SFT model (labeled as ‘SFT’ in the figure) achieves a win-rate of 33.9% against DaVinci-003 using pairwise judgments from the humans. Further, we observe that the Best-of-64 achieves a win-rate of 39.2% and 45.1% against DaVinci-003 using the reward models trained with ratings and rankings feedback, respectively. We observe a similar trend for the reward models trained with the ratings feedback using humans and AI as the evaluators. This indicates that the feedback collected at scale from AI LLMs is useful for improving the alignment of the SFT.

Evaluation Inconsistency. To our surprise, we observe a *evaluation inconsistency* phenomenon where the choice of the feedback protocol during evaluation favors the alignment algorithm that uses the same feedback protocol. Specifically, we find that the gap between the win-rate of the Best-of-n (rankings) policy and the SFT (11.2%) is larger than the gap between the Best-of-n (ratings) policy and SFT (5.3%) as judged by the human annotators under the rankings protocol. However, the gap between the win-rate of the Best-of-n (ratings) policy and SFT (5%) is slightly larger than the gap between the Best-of-n (rankings) policy and SFT (4.3%) as judged by the human annotators under the ratings protocol. Interestingly, we observe a similar evaluation inconsistency when GPT-3.5-Turbo is used as the evaluation annotator.

This indicates that the annotators (human and AI) perceive the quality of the policy responses differently under the different feedback protocols. It ties back to our observations on the consistency problem in the feedback acquisition protocols. In summary, the feedback and evaluation inconsistency problems have severe implications for the practitioners as the protocol for feedback acquisition dictates each step of the alignment pipeline.

6 RELATED WORK

Learning from Human Feedback. In a typical Reinforcement Learning setting, a policy is optimized to maximize a reward function, which is explicitly defined by a human. [11] introduced the Reinforcement Learning from Human Feedback (RLHF) framework, which initially learns a reward model network from a dataset of human preferences. This reward model is then used to replace a human-written reward function during RL optimization. [37] and [29] introduced an RLHF framework for aligning LMs with human preferences. They first learn a reward model trained on rankings of human feedback sampled from LM outputs, which is then used to optimize an LM with an RL algorithm. In contrast, [36] and [32] bypass the learning of a reward model and directly tune an LM using human preference data. Although we do not perform reward-based optimization experiments beyond Best-of-n sampling, we expect our findings to hold similarly for RLHF or other methods of learning from human feedback.

Learning from AI Feedback. While RLHF has shown promising results, the collection of human preferences remains a significant cost for aligning language models. Prior work has demonstrated that LMs are capable of providing feedback [19, 55, 13] that can be used to improve their own capabilities [25, 1]. [4] introduced the Reinforcement Learning from AI Feedback (RLAIF) framework, which replaces the human feedback step of RLHF with an automatic AI feedback generation procedure. Our work compares human feedback with AI feedback in both rankings and ratings settings.

Feedback Data Collection. While RLHF and RLAIF have proven to be highly effective methods for aligning LLMs, additional challenges remain, including the optimization of feedback collection and reward modeling [9]. Previous works have utilized both human and AI feedback in both ratings [22, 20] and rankings [37, 29, 39, 13] settings on machine-written responses [3] and human-written responses [27]. However, there has not been work that systematically compares the effects of collecting ratings versus rankings feedback. While prior works have investigated the use of rich forms of feedback such as fine-grained feedback [48] and system-level natural language feedback [52], the focus of our work is solely on ratings and rankings feedback.

7 CONCLUSION

In this study, we conducted a rigorous examination of feedback acquisition protocols aimed at aligning large language models. Our investigation revealed a notable challenge within the realm of sparse feedback protocols, specifically ratings and rankings, which manifests as an issue of consistency. We observed instances where the preferences assigned to a pair of responses were in conflict with each other. Importantly, our research determined that this consistency problem was not limited to any particular group; both human annotators and AI systems exhibited this inconsistency. We delved into the factors contributing to this inconsistency and shed light on its practical consequences in the context of model alignment and evaluation, notably highlighting the emergence of evaluation inconsistency. Our findings underscore the necessity of a critical approach to curating feedback data collected through various protocols. Moreover, they emphasize that the trends observed in evaluations are significantly influenced by the choice of feedback protocols. Future research endeavors

may explore the cognitive underpinnings of the consistency problem, expand the array of feedback protocols to include denser feedback, and investigate the implications of these choices on the alignment algorithm and subsequent evaluation procedures.

8 LIMITATIONS

While there are many different ways to collect feedback, our focus is on relative and absolute feedback. Future work should explore the impact of collecting richer forms of feedback. Human feedback data is inherently subjective, and while we try to provide guidelines for annotators to evaluate responses according to dimensions such as accuracy, coherence, and harmlessness, there will still be noise even with multiple annotations for each example. Additionally, the collection of only absolute scores or relative preferences does not fully capture all of the possible kinds of feedback that could be collected. More rich forms of feedback data should be explored in future work.

Our analysis is primarily focused on the impact of different feedback collection methods on the downstream performance of LMs as evaluated by win-rate against DaVinci-003. Future work should investigate the impact of different feedback collection and conversion methods on helpfulness, harmfulness, and hallucinations. Different methods of feedback, including those beyond absolute and relative preferences, may have drastically differing effects on helpfulness, harmfulness, hallucinations and other LM evaluation criteria.

Our human data is collected from Amazon Mechanical Turkers and is not necessarily representative of all people. Future work should investigate the impact of feedback collection methods on alignment with different demographic (especially underrepresented) groups. Some methods of feedback collection may amplify bias in the annotator demographics in the trained reward models more than others. Additionally, while we focus on open-ended response domains, our findings on the impact of different feedback collection methods may differ in more application specific domains such as mathematical reasoning or coding tasks. For instance, in a mathematical proof task, binary feedback may be more helpful since a step in a proof is always correct or incorrect.

ACKNOWLEDGEMENT

Hritik Bansal is supported in part by AFOSR MURI grant FA9550-22-1-0380. We also want to thank Ashima Suvama and Nikil Selvam for their helpful suggestions.

REFERENCES

- [1] Afra Feyza Akyürek, Ekin Akyürek, Aman Madaan, Ashwin Kalyan, Peter Clark, Derry Wijaya, and Niket Tandon. RL4f: Generating natural language feedback with reinforcement learning for repairing model outputs. *arXiv preprint arXiv:2305.08844*, 2023.
- [2] Amanda Askell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, Andy Jones, Nicholas Joseph, Ben Mann, Nova DasSarma, et al. A general language assistant as a laboratory for alignment. *arXiv preprint arXiv:2112.00861*, 2021.
- [3] Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.
- [4] Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*, 2022.
- [5] Yoshua Bengio, Réjean Ducharme, and Pascal Vincent. A neural probabilistic language model. *Advances in neural information processing systems*, 13, 2000.
- [6] Yonatan Bitton, Hritik Bansal, Jack Hessel, Rulin Shao, Wanrong Zhu, Anas Awadalla, Josh Gardner, Rohan Taori, and Ludwig Schimdt. Visit-bench: A benchmark for vision-language instruction following inspired by real-world use, 2023.

- [7] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.
- [8] Jon Ander Campos and Jun Shern. Training language models with language feedback. In *ACL Workshop on Learning with Natural Language Supervision*. 2022., 2022.
- [9] Stephen Casper, Xander Davies, Claudia Shi, Thomas Krendl Gilbert, Jérémy Scheurer, Javier Rando, Rachel Freedman, Tomasz Korbak, David Lindner, Pedro Freire, Tony Wang, Samuel Marks, Charbel-Raphaël Segerie, Micah Carroll, Andi Peng, Phillip Christoffersen, Mehul Damani, Stewart Slocum, Usman Anwar, Anand Siththaranjan, Max Nadeau, Eric J. Michaud, Jacob Pfau, Dmitrii Krashennnikov, Xin Chen, Lauro Langosco, Peter Hase, Erdem Bıyık, Anca Dragan, David Krueger, Dorsa Sadigh, and Dylan Hadfield-Menell. Open problems and fundamental limitations of reinforcement learning from human feedback, 2023.
- [10] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. See <https://vicuna.lmsys.org> (accessed 14 April 2023), 2023.
- [11] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.
- [12] Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. Free dolly: Introducing the world’s first truly open instruction-tuned llm, 2023. URL <https://www.databricks.com/blog/2023/04/12/dolly-first-open-commercially-viable-instruction-tuned-llm>.
- [13] Yann Dubois, Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. AlpacaFarm: A simulation framework for methods that learn from human feedback. *arXiv preprint arXiv:2305.14387*, 2023.
- [14] Xinyang Geng, Arnav Gudibande, Hao Liu, Eric Wallace, Pieter Abbeel, Sergey Levine, and Dawn Song. Koala: A dialogue model for academic research. Blog post, April 2023. URL <https://bair.berkeley.edu/blog/2023/04/03/koala/>.
- [15] Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. Chatgpt outperforms crowd-workers for text-annotation tasks. *arXiv preprint arXiv:2303.15056*, 2023.
- [16] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.
- [17] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021.
- [18] Frederick Jelinek. Interpolated estimation of markov source parameters from sparse data. In *Proc. Workshop on Pattern Recognition in Practice*, 1980, 1980.
- [19] Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield Dodds, Nova DasSarma, Eli Tran-Johnson, et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022.
- [20] Kimin Lee, Hao Liu, Moonkyung Ryu, Olivia Watkins, Yuqing Du, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, and Shixiang Shane Gu. Aligning text-to-image models using human feedback. *arXiv preprint arXiv:2302.12192*, 2023.
- [21] Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. AlpacaEval: An automatic evaluator of instruction-following models. https://github.com/tatsu-lab/alpaca_eval, 2023.

- [22] Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let’s verify step by step. *arXiv preprint arXiv:2305.20050*, 2023.
- [23] Chin-Yew Lin and Franz Josef Och. Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pp. 605–612, 2004.
- [24] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.
- [25] Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegrefe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. *arXiv preprint arXiv:2303.17651*, 2023.
- [26] Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*, 2021.
- [27] Stanford NLP. Stanfordnlp/shp · datasets at hugging face. URL <https://huggingface.co/datasets/stanfordnlp/SHP>.
- [28] OpenAI. Gpt-4 technical report, 2023.
- [29] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.
- [30] Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. Instruction tuning with gpt-4. *arXiv preprint arXiv:2304.03277*, 2023.
- [31] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.
- [32] Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint arXiv:2305.18290*, 2023.
- [33] Jérémy Scheurer, Jon Ander Campos, Tomasz Korbak, Jun Shern Chan, Angelica Chen, Kyunghyun Cho, and Ethan Perez. Training language models with language feedback at scale. *arXiv preprint arXiv:2303.16755*, 2023.
- [34] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017.
- [35] Weiyan Shi, Emily Dinan, Kurt Shuster, Jason Weston, and Jing Xu. When life gives you lemons, make cherryade: Converting feedback from bad responses into good labels. *arXiv preprint arXiv:2210.15893*, 2022.
- [36] Feifan Song, Bowen Yu, Minghao Li, Haiyang Yu, Fei Huang, Yongbin Li, and Houfeng Wang. Preference ranking optimization for human alignment. *arXiv preprint arXiv:2306.17492*, 2023.
- [37] Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33:3008–3021, 2020.
- [38] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca, 2023.
- [39] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

- [40] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.
- [41] Amos Tversky. Intransitivity of preferences. *Psychological review*, 76(1):31, 1969.
- [42] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.
- [43] Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, et al. Super-naturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks. *arXiv preprint arXiv:2204.07705*, 2022.
- [44] Yizhong Wang, Hamish Ivison, Pradeep Dasigi, Jack Hessel, Tushar Khot, Khyathi Raghavi Chandu, David Wadden, Kelsey MacMillan, Noah A Smith, Iz Beltagy, et al. How far can camels go? exploring the state of instruction tuning on open resources. *arXiv preprint arXiv:2306.04751*, 2023.
- [45] Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions, 2023.
- [46] Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*, 2021.
- [47] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837, 2022.
- [48] Zeqiu Wu, Yushi Hu, Weijia Shi, Nouha Dziri, Alane Suhr, Prithviraj Ammanabrolu, Noah A Smith, Mari Ostendorf, and Hannaneh Hajishirzi. Fine-grained human feedback gives better rewards for language model training. *arXiv preprint arXiv:2306.01693*, 2023.
- [49] Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. Wizardlm: Empowering large language models to follow complex instructions. *arXiv preprint arXiv:2304.12244*, 2023.
- [50] Kevin Yang, Dan Klein, Asli Celikyilmaz, Nanyun Peng, and Yuandong Tian. Rlcd: Reinforcement learning from contrast distillation for language model alignment. *arXiv preprint arXiv:2307.12950*, 2023.
- [51] Da Yin, Xiao Liu, Fan Yin, Ming Zhong, Hritik Bansal, Jiawei Han, and Kai-Wei Chang. Dynosaur: A dynamic growth paradigm for instruction-tuning data curation, 2023.
- [52] Weizhe Yuan, Kyunghyun Cho, and Jason Weston. System-level natural language feedback, 2023.
- [53] Zheng Yuan, Hongyi Yuan, Chuanqi Tan, Wei Wang, Songfang Huang, and Fei Huang. Rrhf: Rank responses to align language models with human feedback without tears. *arXiv preprint arXiv:2304.05302*, 2023.
- [54] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.
- [55] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *arXiv preprint arXiv:2306.05685*, 2023.

	2	3	4	5	6	7
Humans	45.4	49.7	48.9	46.3	47.7	48.3
GPT-3.5-Turbo	32.4	45.2	41.5	47	48.7	46.7

(a) Average response length

	2	3	4	5	6	7
Humans	34.2	36.6	36.6	35.7	36.7	37.1
GPT-3.5-Turbo	25	34.4	31.4	35.8	37.5	36.5

(b) Average number of unique words

Table 6: Analysis to study the length or number of unique tokens bias in our setup for ratings feedback

	Preferred	Unpreferred		Preferred	Unpreferred
Human	49.5	49.5	Human	37.8	37.7
GPT-3.5-Turbo	49.0	50.2	GPT-3.5-Turbo	37.8	38.0

(a) Average length of the responses

(b) Average number of unique words

Table 7: Analysis to study the bias towards longer responses (number of whitespace words not the subword tokens) or responses with most unique number of whitespace words not the subword tokens. These numbers are calculated for 1800 comparisons from GPT-3.5-Turbo and Humans.

A ANALYSIS OF THE RESPONSE LENGTH AND NUMBER OF UNIQUE WORDS

B LIST OF TASKS FROM SUPERNI-V2

The list of Super-Natural Instructions [43] tasks used in the feedback data are presented in Table 8.

C QUALITATIVE EXAMPLES WITH HUMAN EXPLANATIONS

We present a few qualitative examples with explanations from the human annotators in Table 9, 10, 11 and 12. In §4.3, we provide the discussion on the inconsistent examples shown in Table 9 and 10. Here, we provide the discussion on consistent instances shown in Table 11 and 12.

Table 10 contains another pair of responses where ratings and rankings feedback were inconsistent. Both responses provide decent answers to the instructions. Explanations of rankings feedback highlighted the differences between the responses, including that the rejected response was “a bit unnatural” compared to the chosen response. On the other hand, explanations of ratings feedback mainly provided insights into why each response was good and thus received a high score. These observations suggest that the inconsistency problem arises from systematic variations in the annotator preferences with the differences in the nature of the feedback protocol.

Table 11 contains an example where rankings and ratings data are consistent with each other. The instruction asks for a question such that every item in a list of given inputs is a correct and reasonable answer to the question. The explanations for ratings feedback for the response 2 which contained a correct response were very simple, usually just indicating that the instruction was followed. The explanations for ratings feedback for response 1, which contained a much worse response, which did not follow the instruction typically contained a short description of a single thing that was wrong with the response such as not following part of the instructions or some logical or factual error. The explanations for rankings feedback contained more detailed analysis of the differences between the two responses and tended to be longer than explanations for ratings feedback. Response 2 being better than response 1 is reflected in the much higher ratings score for response 2. The size of this difference is not captured by rankings feedback.

Table 12 contains another example where ratings and consistent feedback is consistent. The instruction asks for a factual answer to a simple question. Response 1 provides the correct answer and response 2 provides an incorrect answer. For very straightforward, short responses, both rankings and ratings feedback explanations are similar indicating that response 1 is correct while response 2

is incorrect. This straightforward difference is reflected in the very large ratings score gap between the two responses.

D TRAINING SETTINGS

We train the reward models on the rankings and ratings feedback data for 5 epochs with early stopping at the end of every epoch. We use the validation loss for early stopping. We train the reward models on a single Nvidia RTX A6000 GPU with 48GB VRAM.

The reward models optimized using equation 2 are trained with an effective batch size = 16 where the batch size = 4 and the number of gradient accumulation steps = 4. Here, each batch contains all the pairs of candidate responses for a given instruction. Since we have 5 candidate responses for a given instruction in the feedback data, we get 10 pairs of candidate responses (5 choose 2) in a batch. The reward models optimized using equation 1 are trained with an effective batch size = 64 where the batch size = 16 and the number of gradient accumulation steps = 4.

Both the reward models use the AdamW optimizer [24] with a linear warmup of 100 steps to a maximum learning rate followed by a cosine decay. We perform a hyperparameter search over $\{1e-4, 1e-5, 1e-6\}$ for maximum learning rate. We find that $1e-4$ for Alpaca-LoRA reward model architecture. We also apply a weight decay of 0.001 to all the reward models and train them at fp16 precision.

task001_quoref_question_generation, task003_mctaco_question_generation_event_duration, task006_mctaco_question_generation_transient_stationary, task009_mctaco_question_generation_event_ordering, task015_mctaco_question_generation_frequency, task023_cosmosqa_question_generation, task026_drop_question_generation, task030_winogrande_full_person, task031_winogrande_question_generation_object, task032_winogrande_question_generation_person, task040_qasc_question_generation, task045_miscellaneous_sentence_paraphrasing, task053_multirc_correct_bad_question, task059_ropes_story_generation, task067_abductivenli_answer_generation, task068_abductivenli_incorrect_answer_generation, task071_abductivenli_answer_generation, task072_abductivenli_answer_generation, task074_squad1.1_question_generation, task077_splash_explanation_to_sql, task082_babi_t1_single_supporting_fact_question_generation, task083_babi_t1_single_supporting_fact_answer_generation, task103_facts2story_long_text_generation, task110_logic2text_sentence_generation, task111_asset_sentence_simplification, task132_dais_text_modification, task182_duorc_question_generation, task193_duorc_question_generation, task203_mnli_sentence_generation, task210_logic2text_structured_text_generation, task216_rocstories_correct_answer_generation, task223_quartz_explanation_generation, task246_dream_question_generation, task269_csrg_counterfactual_story_generation, task270_csrg_counterfactual_context_generation, task360_spolin_yesand_response_generation, task389_torque_generate_temporal_question, task393_plausible_result_generation, task394_persianqa_question_generation, task395_persianqa_answer_generation, task418_persent_title_generation, task454_swag_incorrect_answer_generation, task455_swag_context_generation, task466_parsinlu_qqp_text_modification, task489_mwsc_question_generation, task510_reddit_tifu_title_summarization, task511_reddit_tifu_long_text_summarization, task519_aquamuse_question_generation, task550_discofuse_sentence_generation, task572_recipe_nlg_text_generation, task591_sciq_answer_generation, task592_sciq_incorrect_answer_generation, task593_sciq_explanation_generation, task594_sciq_question_generation, task599_cuad_question_generation, task626_xlwic_sentence_based_on_given_word_sentence_generation, task627_xlwic_word_with_same_meaning_sentence_generation, task628_xlwic_word_with_different_meaning_sentence_generation, task645_summarization, task668_extreme_abstract_summarization, task671_ambigqa_text_generation, task672_amazon_and_yelp_summarization_dataset_summarization, task683_online_privacy_policy_text_purpose_answer_generation, task684_online_privacy_policy_text_information_type_generation, task739_lhoestq_question_generation, task743_eurlex_summarization, task821_protoqa_question_generation, task853_hippocorpus_long_text_generation, task857_inquisitive_question_generation, task859_prost_question_generation, task860_prost_mcq_generation, task871_msMarco_question_generation, task897_freebase_qa_topic_question_generation, task955_wiki_auto_style_transfer, task957_e2e_nlg_text_generation_generate, task967_ruletaker_incorrect_fact_generation_based_on_given_paragraph, task1290_xsum_summarization, task1291_multi_news_summarization, task1325_qa_zre_question_generation_on_subject_relation, task1326_qa_zre_question_generation_from_answer, task1345_glue_qqp_question_paraphrasing, task1355_sent_comp_summarization, task1369_healthfact_sentence_generation, task1437_doqa_cooking_question_generation, task1440_doqa_movies_question_generation, task1499_dstc3_summarization, task1519_qa_srl_question_generation, task1530_scitail1.1_sentence_generation, task1540_parsed_pdfs_summarization, task1552_scitail_question_generation, task1553_cnn_dailymail_summarization, task1561_clickbait_new_bg_summarization, task1562_zest_text_modification, task1567_propara_question_generation, task1572_samsum_summary, task1580_eqasc-perturbed_question_generation, task1586_scifact_title_generation, task1594_yahoo_answers_topics_question_generation, task1595_event2mind_text_generation.1

Table 8: Subset of SuperNatural Instructions v2 used in experiments

Example 1: Both responses do not complete the task fully.

Instruction: *You need to create a question containing a blank (.), based on the given context word. Your question must contain two persons –PersonX and PersonY. The expected answer to your question must be PersonX. PersonX and PersonY should not be equally likely to fill the blank. There should be an agreed upon answer to fill in the blank. Your question must contain at least 15 and at most 30 words. You must utilize the given context word while writing the question. Your question must contain only one blank. Make sure that Person X and Person Y have the same gender. In your question, PersonX and PersonY should be used only ONCE and PersonX should appear earlier than PersonY.*

Input: Context Word is ‘peeling’

Response 1: *Person X and Person Y are peeling an orange together. Who is better at peeling oranges, person X or person Y? (Person X should be considered the expert in this scenario).*

Response 2: *PersonX and PersonY are peeling an orange. PersonX has a knife with which he/she is peeling the orange. PersonY is talking to PersonX and is asking for a sip of juice from the orange. PersonX is saying no, and PersonY smiles sheepishly. PersonX eventually gives in and they both sip the juice from the orange. Who is peeling the orange? PersonX*

Rankings Feedback: **Response 2**

Explanations:

Annotator 1 w/ preference for Response 1: *“Neither question contains a blank, but I picked response A as response B explicitly answers the question within the question’s own text.”*

Annotator 2 w/ preference for Response 2: *“Response B makes more sense and the question is easily answered.”*

Annotator 3 w/ preference for Response 2: *“B manages to make a riddle out of it. A is kind of boring in comparison.”*

Ratings Feedback for Response 1: **Score 4**

Explanations:

Annotator 1’ w/ Score 4: *“Response is relevant to the input, but only partially follows complicated instructions. no errors or harm. ”*

Annotator 2’ w/ Score 4: *“It doesn’t include a (.), and though the answer to the question is Person X, it gave the answer away. Person X & Y are otherwise equally likely to be the correct answer.”*

Annotator 3’ w/ Score 5: *“There is 1 answer, but the structure of the response doesn’t fit the instructions. ”*

Ratings Feedback for Response 2: **Score 2**

Explanations:

Annotator 1’’ w/ Score 2: *“This does not meet the task structure requirements, and both Person X and Person Y are peeling oranges. ”*

Annotator 2’’ w/ Score 1: *“I gave it the lowest score because: (1) PersonX and PersonY was used more than once; (2) the expected answer is not just PersonX, but can also be PersonY; (3) word count is greater than 15 - 30 words.”*

Annotator 3’’ w/ Score 2:*“Everything is wrong with it... ”Person X” and ”Person Y” are used a lot more than once, either of them could be the answer, the question is not fill-in-the-blank or 15-30 words, etc. ”*

Table 9: Example explanation for samples with inconsistent feedback data

Example 2: Both responses provide decent answers to the instruction.

Instruction: *Provided the input sentence, you're expected to write a question that involves event "frequency", which refers to how often an event is likely to be repeated. For example, "taking showers" typically occurs 5 times a week, "going to Saturday market" usually happens every few weeks/months, etc. Don't create questions which have explicit mentions of answers in the text. Instead, it has to be implied from what is given. In other words, we want you to use "instinct" or "common sense". Also, the written questions are not required to have a single correct answer.*

Input: *Sentence: His counter-attack with Dayak warriors drove the Chinese out of Bau and across the Sarawak border.*

Response 1: *How often was his counter-attack with Dayak warriors likely to happen?*

Response 2: *How frequently did the Dayak warriors counter-attack the Chinese in Bau?*

Rankings Feedback: **Response 1**

Explanations:

Annotator 1 w/ preference for Response 1: "A does not contain the flagged word. "

Annotator 2 w/ preference for Response 2: "While both responses are a bit unnatural, B fits more closely with the input text in that it asks about this historical event, rather than asking for speculation about the future, as response A does. "

Annotator 3 w/ preference for Response 1: "response A fulfils all the instructions given which makes it better as option B made mention of answers in the text "

Ratings Feedback for Response 1: **Score 6**

Explanations:

Annotator 1' w/ Score 6: "This question is relevant to the input and it follows the instruction of having to do with frequency."

Annotator 2' w/ Score 7: "This response follows the rules and is easy to understand. "

Annotator 3' w/ Score 5: "It is a question that involves frequency, but I'm not sure it's one that can be answered based on instinct or common sense. Though, the input sentence doesn't leave many options."

Ratings Feedback for Response 2: **Score 7**

Explanations:

Annotator 1'' w/ Score 7: "This response meets the task requirements."

Annotator 2'' w/ Score 6: "question is about frequency, followed directions"

Annotator 3'' w/ Score 7: "Question properly seeks frequency of the event."

Table 10: Example explanation for samples with inconsistent feedback data

Example 1: One response is much better than the other	
Instruction: <i>Construct a question that every answer in the list is correct and reasonable for it.</i>	
Input: <i>telephone, electricity, tv, car, computer, microwave</i>	
Response 1: <i>Which home appliances did you purchase in the past year?</i>	
Response 2: <i>What modern technology have been the most influential in your everyday life?</i>	
Rankings Feedback:	Response 2
<i>Explanations:</i>	
<i>Annotator 1 w/ preference for Response 2: "It's quite unlikely that someone had purchased all those items in the last year, and a car and electricity both don't qualify as a home appliance. Response B seems more fitting."</i>	
<i>Annotator 2 w/ preference for Response 2: "Response B makes more sense because people don't really think of cars or telephones as appliances."</i>	
<i>Annotator 3 w/ preference for Response 2: "The list given in the input is not confined to examples of appliances, so I would discard Response A since it is not a natural question to ask and expect this given list as an answer. Response B is a bit unnatural in its tone, but it at least fits with the expected answer from the input."</i>	
Ratings Feedback for Response 1:	Score 3
<i>Explanations:</i>	
<i>Annotator 1' w/ Score 2: "Electricity is not a home appliance, and it's unlikely that a consumer would purchase all of these things in the same year."</i>	
<i>Annotator 2' w/ Score 3: "The response does not fully follow the instruction to construct a question. Not all items in the given list can be considered home appliances. "</i>	
<i>Annotator 3' w/ Score 4: "The answer to the Response's question would vary from person to person."</i>	
Ratings Feedback for Response 2:	Score 7
<i>Explanations:</i>	
<i>Annotator 1'' w/ Score 7: "'All of the input items are modern technological inventions that are influential in human life. This is accurate and concise."</i>	
<i>Annotator 2'' w/ Score 7: "This question fits all of the input values."</i>	
<i>Annotator 3'' w/ Score 7: "perfectly lines up with all the answers in the input"</i>	

Table 11: Example explanation for sample with consistent feedback data

Example 2: One response contains a clear factual inaccuracy while the other does not

Instruction: *Who is the all time leading goal scorer for Manchester United?*

Input: N/A

Response 1: *The all time leading goal scorer for Manchester United is Wayne Rooney with 253 goals.*

Response 2: *The all time leading goal scorer for Manchester United is Sir. Ralph Chapman with 414 goals*

Rankings Feedback: Response 1

Explanations:

Annotator 1 w/ preference for Response 1: “Response A gives the correct answer according to google. B is wrong.”

Annotator 2 w/ preference for Response 1: “Response A is correct, while Response B is not.”

Annotator 3 w/ preference for Response 1: “A is the correct answer.”

Ratings Feedback for Response 1: Score 7

Explanations:

Annotator 1' w/ Score 7: “gives correct and precise info”

Annotator 2' w/ Score 7: “The answer is relevant to the instruction and seems factual. It is coherent and easy to understand. It does not have any grammatical errors and id harmless.”

Annotator 3' w/ Score 7: “This is the perfect answer. It's accurate and written in a complete sentence.”

Ratings Feedback for Response 2: Score 1

Explanations:

Annotator 1'' w/ Score 1: “The answer very wrong...”

Annotator 2'' w/ Score 1: “This should be Wayne Rooney.”

Annotator 3'' w/ Score 1: “Wayne Rooney tops the list of all-time goal-scorers for Manchester United.”

Table 12: Example explanation for sample with consistent feedback data