RESCUE: Ranking LLM Responses with Partial Ordering to Improve Response Generation

Anonymous ACL submission

Abstract

Customizing LLMs for a specific task involves distinguishing effective responses from erroneous ones. This skill can be developed using supervised fine-tuning with extensive human preference data. However, obtaining expert-annotated preference data is expensive for most tasks. In this paper, we present a novel method to optimize LLMs using ranking metrics. This method trains the model to prioritize the best responses from a pool of candidates created for a particular task. Rather than a traditional full ordering, we advocate for a partial ordering, as achieving consensus on the perfect order of candidate responses can be challenging. Our partial ordering is more robust, less sensitive to noise, and can be achieved with limited human annotations or through heuristic methods. We test our system's improved response generation ability using benchmark datasets, including the latest multi-document question answering task. We conduct ablation studies to understand crucial factors, such as how to gather candidate responses for specific tasks, determine their most suitable order, and balance supervised fine-tuning with ranking metrics. Our approach, named RESCUE, suggests a promising avenue for enhancing LLMs' contextual understanding via response ranking.[1]

1 Introduction

A significant advantage of large language models (LLMs) is their ability to explain their predictions (Ziegler et al., 2020; Alkhamissi et al., 2023; Ludan et al., 2023; Li et al., 2023; Ye et al., 2023). For example, LLMs may suggest lab tests to doctors based on patient symptoms (Peng et al., 2023) or help financial analysts evaluate risks in their investment portfolios (Romanko et al., 2023), providing explanations for each. As LLMs increasingly assist in decision-making across domains, examining the quality of their explanations becomes crucial. Previous studies indicate that the quality of model explanations can vary, and they may mislead about the true reason behind a model's prediction (Burns et al., 2022; Turpin et al., 2023). Therefore, it is vital to improve LLMs' explanation quality, along with enhancing their task accuracy.

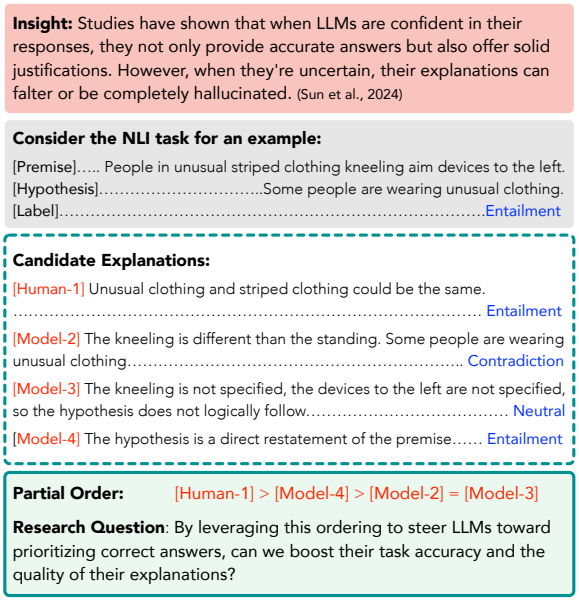

Figure 1: Accurate model predictions often come with high-quality explanations. We build on this insight to rank candidate responses, where human responses are placed above model responses with correct labels, which in turn are prioritized over incorrect responses. This hierarchy encourages the LLM to generate explanations that lead to accurate labels.

We focus on LLM responses that consist of *a predicted label* and *a detailed explanation*. LLMs should provide not only accurate labels but also sound rationales to support their predictions. Explanations can be generated through the model's self-reflection, using methods such as chain-of-thoughts or tree-of-thoughts (Yao et al., 2022; Wei et al., 2023; Yao et al., 2023; Shinn et al., 2023). Explanations can also be embedded in prompts to guide LLMs in new tasks via in-context learning (Ye et al., 2023). In this study, we take a step further by exploring methods to train an open-source LLM to effectively rank candidate responses, which we acquire from diverse sources. Ranking responses

[1] We plan to release our source code and models publicly.

enables the LLM to differentiate between sound and flawed explanations for a specific task, thereby enhancing response generation.

Interestingly, accurate model predictions often come with high-quality explanations. Studies have shown that when LLMs are confident in their responses, they not only provide accurate answers but also offer solid justifications. On the flip side, when they're uncertain, their explanations can falter or be completely hallucinated (Singh et al., 2023; Liu et al., 2023a; Sun et al., 2024). Our paper builds on this insight to rank candidate responses. We place human responses above model responses with correct labels, which in turn are prioritized over incorrect responses. This hierarchy encourages the LLM to generate explanations comparable to humans' or, at the very least, to produce explanations that lead to accurate labels.

Our method benefits from requiring minimal expert annotations, which is a frequent challenge in most domain-specific tasks. Unlike reinforcement learning with human feedback (RLHF; Ziegler et al., 2020) or direct preference optimization (DPO; Rafailov et al., 2023), which need extensive expert-annotated data, our approach is cost-effective and practical in resource-constrained situations. We employ a partial ordering of LLM responses, which can be acquired with limited human annotations or through heuristic functions. This study's contributions are summarized as follows:

- We seek to improve LLMs' response generation. In training, we supplement each example with candidate responses, featuring a mix of accurate and inaccurate predictions, and sound and flawed explanations. For tasks with long contexts, we anchor responses in different parts of the context to increase diversity. LLM is trained to prioritize the best responses using the ranking metric.
- We test our system's response generation using the latest benchmarks, and conduct ablation studies to understand crucial factors, such as how to gather candidate responses, determine their most suitable order, and balance supervised fine-tuning with ranking metrics. Our approach, named RESCUE, offers a promising way to improve LLMs' contextual understanding via response ranking.

2 Related Work

Learning from Human Preferences Aligning LLM responses with human preferences ensures the models' outputs are helpful, safe, and adhere to societal norms (Bai et al., 2022b; Liu et al., 2023b; Honovich et al., 2023; Wang et al., 2023; Rafailov et al., 2023; Hejna et al., 2023). This research often involves humans performing pairwise or k-wise comparisons on model outputs, which are used to train a reward model (Ziegler et al., 2020; Bai et al., 2022a; Ouyang et al., 2022; Ramamurthy et al., 2023; Zhu et al., 2023). Recently, Rafailov et al. (2023) introduce a new parameterization of the reward model in RLHF that enables extraction of an optimal policy in closed form. Unlike other methods, our study focuses on domain-specific tasks. We guide LLMs to make accurate predictions and generate sound explanations using the limited expert annotations available for those tasks.

Reasoning LLMs can improve their reasoning through trial and error and self-improvement. For example, chain-of-thoughts (Wei et al., 2023) allows LLMs to break down complex tasks step by step into more manageable parts. Tree-of-thoughts (Yao et al., 2023) employs task decomposition via a tree structure, guiding LLMs through various steps and consider multiple thoughts within each step. Reflexion (Shinn et al., 2023) combines dynamic memory and self-reflection to refine reasoning skills. However, pinpointing specific reasoning errors remains a practical challenge. The distinction between sound and flawed explanations can often be subtle and unclear during self-reflection.

Ranking Metrics A ranking objective allows the model to prioritize the best candidates (Yuan et al., 2023), improving its performance in tasks like abstractive summarization and question answering. For example, the BRIO training paradigm (Liu et al., 2022) fine-tunes BART and T5 models to generate reference summaries while using a ranking mechanism to score candidate summaries. This approach could be especially beneficial in retrieval augmented generation (Hopkins and May, 2011; Lewis et al., 2021; Nakano et al., 2022). We believe that explanations grounded on incorrect documents should be discounted and those grounded in reference documents be promoted. Our method leverages this insight to enhance the model's ability to generate contextually accurate explanations.

3 Our Approach: RESCUE

Let $x \sim \mathcal{D}$ represent the prompt or context given to the model, and y denote the model's response to prompt x. The response y comprises two parts: a brief justification and a predicted label, separated by the special symbol '####'. For example, in the natural language inference task, it might be "*Un-*

usual clothing and striped clothing could be the same. #### Entailment." Supervised fine-tuning (SFT; Eq. (1)) is a primary method to improve task accuracy by training the model to generate human-written responses y^*. However, since the model has only been exposed to high-quality human responses, its noise robustness remains unvalidated. Prior studies (Ziegler et al., 2020; Touvron et al., 2023) suggest that model performance can plateau quickly, potentially leading to overfitting.

$$\mathcal{L}_{\text{SFT}}(\theta) = -\log \pi_\theta(y^*|x) \tag{1}$$

We proposed to guide the model to prioritize valid responses over flawed ones and contextually accurate responses over inaccurately grounded ones, using a ranking metric as illustrated in Eq. (2). Here, $(x, y_0, y_1, b) \sim \mathcal{S}$ includes a prompt x, two candidate responses, and a binary variable b, where y_b should be scored higher than y_{1-b}. $\mathcal{S}$ represents a diverse set of candidate responses obtained from various sources. For example, responses could be acquired from open-source LLMs like Llama-2 or close-source LLMs like GPT-3.5, GPT-4 or Claude. Human-annotated responses can also be included in the collection when they are available.

$$\begin{aligned}\mathcal{L}_{\text{Rank}}(\theta) = -\mathbb{E}_{(x,y_0,y_1,b)\sim\mathcal{S}}[\\ \max\{0, \log \pi_\theta(y_b|x) - \log \pi_\theta(y_{1-b}|x)\}]\end{aligned} \tag{2}$$

We initiate $\pi_\theta(y|x)$ from a base model $\rho(y|x)$ and subsequently fine-tune it for a specific task with candidate responses. Particularly, $\pi_\theta(y|x)$ is used to loosely represent length-normalized probability $\pi_\theta(y|x) = \frac{1}{|y|^\lambda} \sum_{t=1}^{|y|} \log \pi_\theta(y_t|x, y_{<t})$, where $\lambda > 0$ is the scaling factor for length normalization. Our approach, RESCUE, uses a hyperparameter α to balance the impact of supervised fine-tuning and the ranking metric, as shown in Eq. (3).

$$\mathcal{L}_{\text{RESCUE}}(\theta) = \mathcal{L}_{\text{SFT}}(\theta) + \alpha \mathcal{L}_{\text{Rank}}(\theta) \tag{3}$$

Ranking Metrics vs. Rewards A reward model $r(x, y_i)$ assigns scores to a given prompt x and its corresponding response y_i. As shown in Eq. (4), it allocates the *full probability mass* to the response y_b chosen by human labelers. For this model to function, humans need to provide accurate pairwise preference judgments. Nonetheless, achieving a consensus among human labelers regarding the perfect order of LLM responses can be a daunting task. The labelers often struggle to provide consistent, fine-grained labels (Touvron et al., 2023). As a result, allocating the entire probability mass, i.e., $\log \mathcal{P}_\theta(y_{b'}|x)$ to an incorrectly labeled response $y_{b'}$ can mislead the model and hinder the effective training of the reward model.

$$\mathcal{L}_{\text{Reward}}(r) = -\mathbb{E}_{(x,\{y_i\}_i,b)\sim\mathcal{S}}\left[\log \frac{e^{r(x,y_b)}}{\sum_i e^{r(x,y_i)}}\right] \tag{4}$$

In contrast, our proposed ranking metrics offer greater flexibility and robustness to inconsistencies in human preferences. Our model not only prioritizes y_b over other potential responses using the equation $\max\{0, \log \mathcal{P}_\theta(y_b|x) - \log \mathcal{P}_\theta(y_{1-b}|x)\}$, but further allows minor deviations. For example, the model can still assign a high probability to a less-favored response $\log \mathcal{P}_\theta(y_{1-b}|x)$, provided its probability difference from the top response $\log \mathcal{P}_\theta(y_b|x) - \log \mathcal{P}_\theta(y_{1-b}|x)$ remains minimal. We also advocate for a partial ordering of LLM responses, partitioning them into groups. This group ordering provides a hierarchical perspective, enabling the model to understand the relative importance of each group in a broader context.

4 Ranking LLM Responses

Candidate responses for a given prompt x, can be organized into a strict order. OpenAI has employed a team of trained human labelers to rank sets of model outputs from best to worst to train a reward model (Ziegler et al., 2020; Ouyang et al., 2022). However, this method is quite expensive. We propose two cost-effective approaches to establish a Partial Ordering (PO) of responses.

Our first method, **(PO) Human Prioritization**, posits that human responses should take priority over model responses, as they offer valid rationales and accurate labels. **(PO) Label Prioritization** places responses with correct labels above those with incorrect labels, irrespective of whether they are human or model-generated. This is because rationales resulting in correct labels are more valuable than those leading to incorrect labels. The latter may contain flawed reasoning that misguides their predictions. Lastly, **(PO) Human-Label Hybrid** employs a fine-grained grouping. It places human responses above model responses with correct labels, which are then prioritized over responses with incorrect labels. This hierarchy is designed to motivate the LLM to generate rationales comparable to humans' or, at a minimum, to produce rationales that lead to accurate labels.

Partial Orderings (PO) of responses offer enhanced flexibility and noise robustness. For example, in developing Llama-2, Touvron et al. (2023)

noted that even human labelers struggle to decide between two similar model responses, with annotations for such responses often hinging on subjective judgement and nuanced details. By utilizing a partial order, we only incorporate the most clear-cut pairs of model outputs in the ranking metric, thereby improving the quality of response pairs used in model fine-tuning.

For comparison, we examine two full ordering (FO) approaches. **(FO) Similarity** embeds each candidate response into a vector, which are then ranked based on their Cosine similarity to the vector representing the human response. The second approach **(FO) GPT-3.5-Turbo** leverages the GPT-3.5-Turbo-0613 model to rank candidate responses. We instruct it to prioritize candidates with the same labels as the human response, but allowing it to decide whether this criterion is met. We compare full and partial ordering approaches in §6.

5 Collecting Candidate Responses

We enrich each example with a set of candidate responses, targeting a mix that includes both accurate and inaccurate predictions, along with explanations that are both sound and flawed. We incorporate human annotations into the mix when available. For tasks with long contexts, we anchor responses in different parts of the context to increase diversity. This enriched dataset is used to train our LLM to improve its response generation. Next, we outline two strategies for generating candidate responses.

5.1 Responses Generated by Various LLMs

We focus on the textual entailment task (Bowman et al., 2015; Chen et al., 2017; Camburu et al., 2018; Kumar and Talukdar, 2020) to illustrate our strategy. Specifically, the Stanford NLI dataset identifies relationships between sentence pairs as *entailment*, *contradiction*, or *neutral*. The e-SNLI dataset expands on SNLI by adding human-annotated explanations for these relationships, explaining why sentences are classified in certain ways (Camburu et al., 2018). Similarly, we require LLMs to both *predict and rationalize* their predictions. Our approach then learns to prioritize accurate predictions and their model explanations, while downplaying explanations for inaccurate predictions.

We gather diverse responses for this task from both open-source and proprietary LLMs. Specifically, we sample three responses from `Llama-2-7b`, setting the temperature to 0.8 for diversity, and one from `GPT-3.5-Turbo-0613`, plus a human explanation, making five responses per prompt in total.

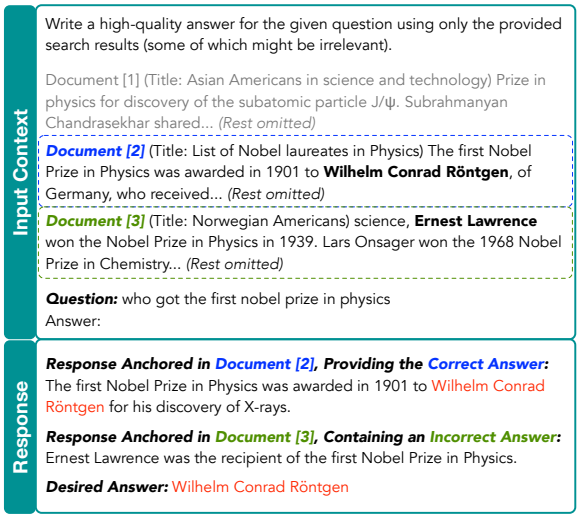

Figure 2: For the Multi-doc QA task, we anchor responses in different parts of the context to produce a diverse set of answers. We generate five candidate responses per instance, one from the gold passage and four from random distractors.

Each response features a brief explanation of the model's reasoning and a predicted label, as shown in Figure 1.

Response Flipping We propose a novel method for collecting diverse responses from LLMs without the need for repetitive response sampling. Our method begins by inverting an LLM's explanation for a given response. For instance, if an LLM suggests, "*The to-go packages may not be from lunch. #### Neutral,*" we flip the explanation to, "*The to-go packages are likely from lunch.*" This reversed explanation then guides the LLM to assign a new label, such as "*#### Entailment.*"

Our method uses `GPT-4-0613` for reversing the explanations, given its extraordinary generation capabilities. The prompt for inversion is: "*Rewrite the sentence to convey the opposite meaning: {Explanation}.*" Afterward, `GPT-3.5-Turbo-0613` is used to predict the appropriate label by combining the original context with the inverted explanation. This method offers an efficient way to generate diverse responses with varying labels.

5.2 Responses Anchored in Various Passages

When dealing with long contexts, we can anchor responses in different parts of the context to produce a diverse set of answers. An LLM can then enhance its performance by discriminating among these answers. For example, in the multi-document question answering task (**Multi-doc QA**; Liu et al. 2023b), the LLM uses 10 to 30 Wikipedia passages as input to answer questions. These questions come from NaturalQuestions-Open (Kwiatkowski et al.,

2019), which contains historical Google queries and their human-annotated answers extracted from Wikipedia. Among the passages given to the model, only one has the answer, the rest are distractors. A retrieval system named Contriever (Izacard et al., 2022) is used to obtain distractor passages, which are most relevant to the question but do not contain the answers.

We use `Llama-2-7b` to generate five diverse candidate responses per instance, one from the gold passage and four from random distractors. Responses containing the desired answer are marked correct, as illustrated in Figure 2. Here, we generate two candidate responses "*The first Nobel Prize in Physics was awarded in 1901 to Wilhelm Conrad Röntgen for his discovery of X-rays.*" and "*Ernest Lawrence was the recipient of the first Nobel Prize in Physics.*" by feeding the model Documents [2] and [3] separately. Our Label-Prioritized approach ranks candidates with the desired answer higher than those without. Human-Label-Hybrid further prefers correct answers anchored in the gold passage. In training, the model receives a question and 10 Wikipedia passages, and learns to differentiate correct from incorrect responses. At test time, the fine-tuned model employs beam search to decode the optimal response.

6 Experiments

We have chosen `Llama-2-7b` as our base model for task-specific training. The `Llama-2` series outperforms other open-source options, such as Falcon (Almazrouei et al., 2023), Mistral (Jiang et al., 2023), Vicuna (Chiang et al., 2023) and MPT (MosaicML, 2023), on a number of tasks. Its `7b` variant requires significantly less GPU memory, which is crucial for specific domains without the specialized infrastructure to serve larger models. We opt for the `Llama-2-7b` over `Llama-2-7b-chat` in this study due to our focus on non-dialogue tasks.

We use AdamW (Loshchilov and Hutter, 2017) with a learning rate of $2e^{-5}$ and a cosine scheduler with a 0.03 warmup rate. Our training utilizes fully sharded data parallelism and BF16 mixed precision training, which is generally faster, consumes less memory, and is preferable for large models. Our experiments are conducted using 4xA100 GPUs, and task-specific training is limited to a single epoch for both supervised fine-tuning and response ranking. This is to mitigate the risk of multi-epoch degradation (Xue et al., 2023) and potential overfitting from repeated exposure to the training data. The batch size is set at B=64, the same configuration

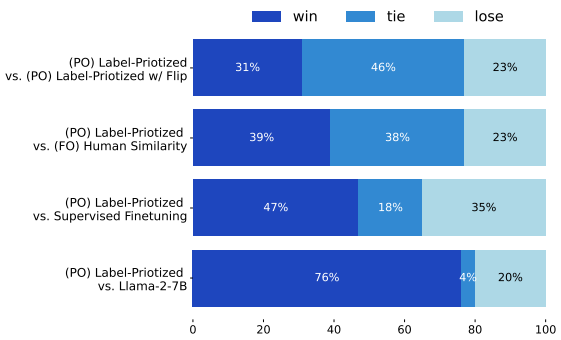

Figure 3: Human evaluation results. Our partial ordering (PO) with label prioritization outperforms the SFT model with an overall win rate of 47%. While SFT shows comparable accuracy in automatic evaluation, it often relies on data artifacts for predictions (Gururangan et al., 2018) and does not yield better explanations. Our PO method also outperforms other methods such as FO Similarity and the base `Llama-2-7b` model.

used for `LLama-2` (Touvron et al., 2023). It is calculated as the product of three factors, $B = g \times b \times D$, combining gradient accumulation steps ($g = 16$), per-GPU batch size ($b = 1$ due to memory constraints), and the number of GPUs ($D = 4$). This strategy allows us to handle a large number of candidates during response ranking.

6.1 Automatic Evaluation of NLI Accuracy

Our goal in this study is to enhance response generation with limited training data, which is a common challenge in real-world scenarios where expert annotations are scarce, often limited to a few thousand examples. We conduct our experiments using the e-SNLI dataset (Camburu et al., 2018), which comprises 549,367 training examples. We intentionally restrict our training to subsets of {2k, 5k, 10k, 20k} samples, approximately 0.4% to 3.6% of the total training set. We report the accuracy of all models on the standard test set of 9,824 examples.

We evaluate a variety of models on this task. In particular, we train the base model with human responses (**SFT**). We also explore two response ranking strategies: full ordering (FO), which ranks candidate model responses by their semantic closeness to human responses (**Similarity**) or as assessed by **GPT-3.5-Turbo**, and partial ordering (PO), which trains the base model to prioritize human responses over those from models (**Human Prioritization**), responses with correct labels over incorrect ones (**Label Prioritization**), and a mix of both (**Human-Label Hybrid**). Both FO and PO rely on our ranking metric detailed in Eq.(3).

Table 1 presents task accuracy across various proportions of training data. We observe that models trained with ranking metrics and incorporating both full and partial ordering strategies outperform those

	System	Proportion of Training Data					w/ Res. Flip.	
		0.4%	0.9%	1.8%	3.6%	AVG	0.4%	0.9%
BASELINE	(SFT) Supervised Finetuning	77.45	85.56	87.33	87.94	84.57	–	–
	(FO) Similarity	81.01	86.69	86.53	86.38	85.15	↑ 5.18	↓ 0.26
	(FO) GPT-3.5-Turbo	82.20	86.62	85.02	86.71	85.14	↑ 3.09	↓ 1.32
OURS	(PO) Human Prioritization	80.70	87.11	87.06	86.26	85.28	↑ 6.10	↓ 1.30
	(PO) Label Prioritization	81.97	87.27	**88.16**	**87.97**	86.34	↑ 5.15	↑ 0.61
	(PO) Human-Label Hybrid	**82.86**	**87.47**	87.33	87.73	**86.35**	↑ 4.88	↑ 0.34

Table 1: Task accuracy of RESCUE on natural language inference, reported on the e-SNLI test set. We observe that models trained with ranking metrics and incorporating both full and partial ordering strategies outperform those trained solely with SFT, especially when working with a few thousand annotated examples. Our partial ordering strategies, namely label prioritization and a hybrid of human and label prioritization, surpass full ordering methods.

trained solely with SFT, especially when working with a few thousand annotated examples. This indicates that training an LLM to rank responses can improve response generation and result in more accurate predictions of textual entailment relationships. The improvement is most notable when using only 0.4% of the total training data, suggesting the advantage of ranking metrics in scenarios with extremely scarce training data.

Our partial ordering strategies, namely label prioritization and a hybrid of human and label prioritization, surpass full ordering methods. This could be because achieving consensus on full ordering of responses is challenging even for humans. This approach may introduce variability in response ranking and destabilizes training. SFT begins to show improvement with 20k or more training examples, although gathering such extensive annotations is often difficult for domain-specific tasks. Additionally, while flipping responses increases answer variety, it might cause a shift in the distribution of ranked responses. We find this technique consistently improves response generation only when training data is limited to 2k examples.

Our models match state-of-the-art performance. E.g., Hsieh et al. (2023) achieved 89.51% accuracy using a 540B LLM with step-by-step distilling. By contrast, our models use only a fraction of the full training set with a 7B model. Without supervised fine-tuning, the base `Llama-2-7b` model yields a significantly lower accuracy of 33.31%. Next, we extend our evaluation to include human assessment of model explanations.

6.2 Human Evaluation of Response Quality

Human evaluation provides a holistic assessment of model responses. We compare several models, including our PO method with label prioritization, SFT, FO method with responses ranked their similarity to human responses, PO model with response flipping, and the base model. These models were trained with varying amounts of training data (0.4% to 3.6%), and the highest performing model across all data proportions was chosen for human evaluation. An annotator evaluated responses for 100 randomly selected samples from the e-SNLI test set, using win, tie and lose to rate each response pair. Evaluations were based on label accuracy and the quality of explanations. A quality explanation should support the predict label with detailed reasoning and show logical coherence.

As Figure 3 illustrates, our partial ordering (PO) with label prioritization outperforms the SFT model with an overall win rate of 47%. This advantage stems from the PO models' ability to distinguish between sound and flawed responses, thus improving response generation. While SFT shows comparable accuracy in automatic evaluation, it often relies on data artifacts for predictions (Gururangan et al., 2018) and does not yield better explanations. Similar to findings from automatic evaluations, adding response flipping does not surpass the original label prioritization method. Our PO method also outperforms other methods such as FO Similarity and the base `Llama-2-7b` model.

6.3 Evaluation of Multi-Document QA

The Multi-Doc QA task involves answering a given question using a set of retrieved documents. Liu et al. (2023c) found that LLMs exhibit a U-shaped curve, depending on where the answer-containing document is located within the input context and highlighting difficulties in accessing relevant information in the middle of long contexts. To mitigate this, we incorporate response ranking. We generate five candidate responses per question, one from the correct document and four from distractors. We then train the base model on 1k examples from the

Position of Gold Document	5 Retrieved Documents				10 Retrieved Documents			
	1st	3rd	5th	AVG	1st	5th	10th	AVG
Base Model (`Llama-2-7b`)	**45.64**	34.19	43.05	40.96	**46.41**	27.17	42.95	38.84
(PO) Label Prioritization	44.88	**42.44**	**53.43**	**46.92**	35.72	**33.43**	**55.11**	**41.42**

Table 2: Answer accuracy for the Multi-QA task. We evaluate two scenarios: the model receives 5 or 10 documents returned by the retriever. We find that the PO method with label prioritization substantially improves model performance, as ranking responses allows the LLM to more effectively identify relevant information, improving the U-shaped curve.

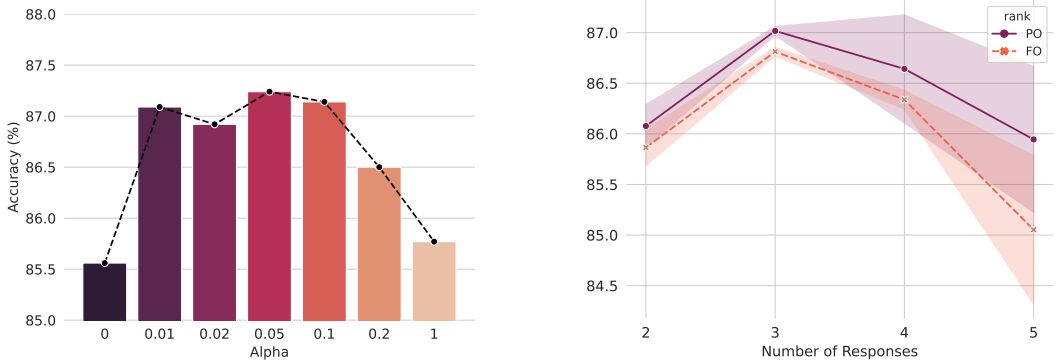

Figure 4: (LEFT) The influence of different α on task accuracy. We find that optimal performance is achieved with an α value between 0.01 to 0.1. (RIGHT) We conduct experiments with a varying number of candidate responses per prompt. Results indicate that performance improvement can be achieved even with 3-4 candidate responses.

training set using our ranking metric (Eq. (2)). SFT is not used due to the absence of human-written explanations for this task. Our method is evaluated on a test set of 665 examples.

Table 2 shows answer accuracy, measured as whether correct answers from the NaturalQuestions annotations appear in the generated responses. We evaluate two scenarios: the model receives 5 or 10 documents returned by the retriever. The correct document is placed either at the beginning (1st position), in the middle (3rd or 5th), or at the end (5th or 10th) of the document set. We find that the PO method with label prioritization substantially improves model performance, as ranking responses allows the LLM to more effectively identify relevant information, improving the U-shaped curve. Our findings also align with those of Liu et al. (2023c), who observed a recency bias in Llama-2-7b. With 20 documents as input, they reported accuracies of about 25% at positions 1, 5, 10, 15, and 42% at position 20. Upon examining the model's responses, we observe that the model often answers questions by copying content, which tends to improve answer accuracy when the answer is located in the middle or end of the context.

7 Discussion

Balancing Coefficient Our approach uses a hyperparameter α to balance the impact of supervised fine-tuning and the ranking metric. Figure 4 shows the influence of different α on task accuracy. We find that optimal performance is achieved with an α value between 0.01 to 0.1. The results indicate that, while supervised fine-tuning is pivotal for RESCUE, integrating the ranking metric enhances the method's robustness to noise.

Number of Candidate Responses We conduct experiments with a varying number of candidate responses per prompt, and the results are shown in Figure 4. In our experiments, we are able to rank up to five candidate responses using four Nvidia A100 GPUs. As the number of candidates increases, so does the demand for additional GPU memory and compute resources. Our experiments indicate that performance improvement can be achieved even with 3-4 candidate responses. Beyond that, RESCUE sees no further gains from increasing the number of responses. This saturation in performance may be attributed to the noise in ranking. Moreover, it highlights the challenges associated with ranking a diverse set of responses differing in length and style of rationales.

Scoring Candidate Responses We identify two characteristics in human responses that distinguish them from model responses. Firstly, they are more concise and to the point. As indicated in Figure 6 (RIGHT), human responses are significantly shorter, averaging 10 fewer tokens per response compared to GPT-3.5's responses. Secondly, we note that LLM responses tend to use more common words, yielding better fluency and generally smoother text

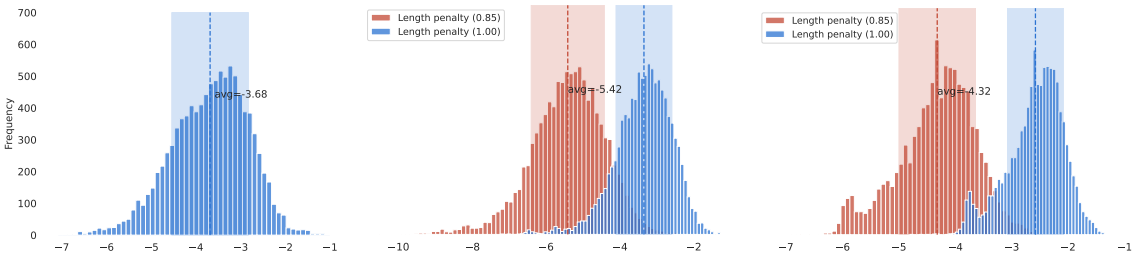

Figure 5: LEFT figure shows the log probabilities of human responses, while MIDDLE and RIGHT figures present those from Llama-2-7B and GPT-3.5-turbo-0613, respectively. We assign a length scaling factor, λ, of 0.85 to all model responses, maintaining a λ of 1.0 for human responses. This approach effectively shifts the log probability score distributions of model responses (colored in red) closer to those of human ones, thereby minimizing margin violations.

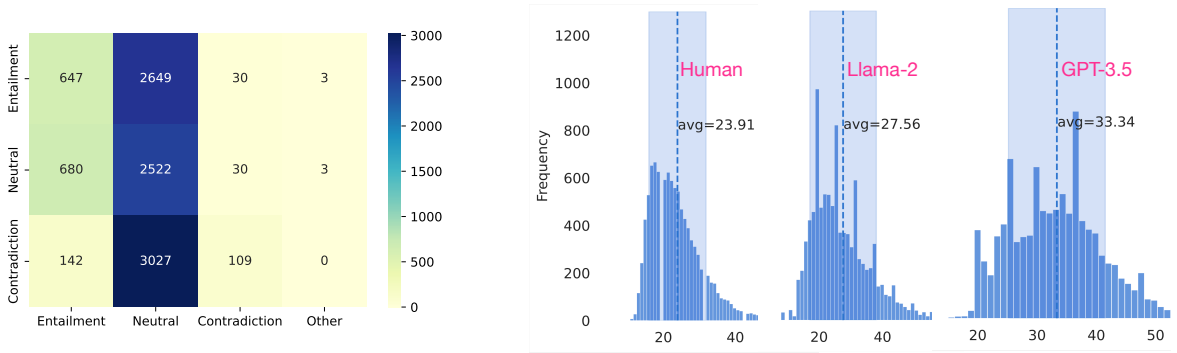

Figure 6: (LEFT) The confusion matrix for the Llama-2-7B base model, where the x-axis represents the labels predicted by Llama-2-7B, and the y-axis represents human labels. The results show Llama-2-7B's tendency to predict neutral labels, as indicated by the dark bar in the middle. (RIGHT) Candidate responses differ in length. We show the distribution of responses from human annotators, Llama-2-7B, and GPT-3.5-turbo-0613 models. Human responses are the shortest, while GPT-3.5's are notably longer, containing on average 10 more tokens per response compared to human responses.

compared to human responses. These characteristics present challenges in ranking responses from diverse sources. Human responses, due to their brevity and unique word choice, often have lower length-normalized log probabilities than model responses. This discrepancy leads to many margin violations during training using Eq. (2), and more parameter updates to ensure human responses score higher than model outputs.

To mitigate this, we assign a length scaling factor λ of 0.85 to all model responses, including those from Llama-2-7B and GPT-3.5-turbo-0613, maintaining a λ of 1.0 for human responses. This effectively shifts the log probability score distributions for model responses closer to human ones (Figure 5), reducing margin violations. We are also exploring adjusting the margin size and curriculum learning, which gradually increases the difficulty of training samples to reduce violations, as potential directions for future research.

Central Tendency Bias LLMs such as Llama-2-7B and GPT-3.5 exhibit a central tendency bias (Goldfarb-Tarrant et al., 2020) in natural language inference. These models often predict *Neutral* labels, leaning towards the "center" of possible labels. Figure 6 presents the confusion matrix, with the x-axis representing predicted labels by Llama-2-7B and the y-axis showing human labels. The results show Llama-2-7B's tendency to predict neutral labels (indicated by the dark bar in the middle) and its avoidance of extreme labels like *Entailment* or *Contradiction*. A plausible reason could be Llama-2-7B's inadequate world knowledge impacting its task accuracy. Moreover, this tendency might originate from the models being trained on human annotations for instruction-following. They frequently give hedging responses to fulfill helpfulness and safety requirements, leading to outputs that are more neutral and less assertive.

8 Conclusion

In this paper, we introduce RESCUE, an approach that trains the LLM to prioritize sound responses over erroneous ones, thereby enhancing overall task accuracy and the quality of explanations. Accurate model predictions often come with high-quality explanations. We build on this insight to rank candidate responses using a partial ordering approach, as achieving consensus on the perfect order of responses is challenging. RESCUE has demonstrated competitive performance on benchmarks.

Limitations

Our approach focuses on optimizing LLMs through ranking metrics and partial ordering of candidate responses. We introduce two innovative strategies for generating candidates: collecting from diverse LLMs and anchoring responses in various parts of the context, showcasing its flexibility across benchmark datasets. We note that organizing candidate responses can benefit from domain-specific criteria, such as sorting recommended lab tests for patients by the relevance of the answer, urgency, and cost. Further, our proposed approach prioritizes the best responses from a set of candidates, thereby improving the task accuracy and the quality of generated explanations. With additional GPU resources, we can improve the variety and representation of candidate responses or categorize them based on domain-specific attributes. Despite existing challenges, our approach offers a promising path for customizing LLMs for specialized applications.

References

Badr Alkhamissi, Siddharth Verma, Ping Yu, Zhijing Jin, Asli Celikyilmaz, and Mona Diab. 2023. Opt-r: Exploring the role of explanations in finetuning and prompting for reasoning skills of large language models. In *Proceedings of the 1st Workshop on Natural Language Reasoning and Structured Explanations (NLRSE)*. Association for Computational Linguistics.

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, Daniele Mazzotta, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. 2023. The falcon series of open language models.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, Ben Mann, and Jared Kaplan. 2022a. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint 2204.05862*.

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

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. 2022. Discovering latent knowledge in language models without supervision.

Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom. 2018. e-snli: Natural language inference with natural language explanations. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 9560–9572.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017. Enhanced LSTM for natural language inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1657–1668, Vancouver, Canada. Association for Computational Linguistics.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Seraphina Goldfarb-Tarrant, Tuhin Chakrabarty, Ralph Weischedel, and Nanyun Peng. 2020. Content planning for neural story generation with aristotelian rescoring. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4319–4338, Online. Association for Computational Linguistics.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel R. Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. *CoRR*, abs/1803.02324.

Joey Hejna, Rafael Rafailov, Harshit Sikchi, Chelsea Finn, Scott Niekum, W. Bradley Knox, and Dorsa Sadigh. 2023. Contrastive preference learning: Learning from human feedback without rl.

Or Honovich, Thomas Scialom, Omer Levy, and Timo Schick. 2023. Unnatural instructions: Tuning language models with (almost) no human labor. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14409–14428, Toronto, Canada. Association for Computational Linguistics.

Mark Hopkins and Jonathan May. 2011. Tuning as ranking. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1352–1362, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alex Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. 2023. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 8003–8017. Association for Computational Linguistics.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. Unsupervised dense information retrieval with contrastive learning.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b.

Sawan Kumar and Partha Talukdar. 2020. NILE : Natural language inference with faithful natural language explanations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8730–8742, Online. Association for Computational Linguistics.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2021. Retrieval-augmented generation for knowledge-intensive nlp tasks.

Jierui Li, Szymon Tworkowski, Yingying Wu, and Raymond Mooney. 2023. Explaining competitive-level programming solutions using llms.

Genglin Liu, Xingyao Wang, Lifan Yuan, Yangyi Chen, and Hao Peng. 2023a. Prudent silence or foolish babble? examining large language models' responses to the unknown.

Hao Liu, Carmelo Sferrazza, and Pieter Abbeel. 2023b. Chain of hindsight aligns language models with feedback.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023c. Lost in the middle: How language models use long contexts. *CoRR*, abs/2307.03172.

Yixin Liu, Pengfei Liu, Dragomir Radev, and Graham Neubig. 2022. Brio: Bringing order to abstractive summarization.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Josh Magnus Ludan, Yixuan Meng, Tai Nguyen, Saurabh Shah, Qing Lyu, Marianna Apidianaki, and Chris Callison-Burch. 2023. Explanation-based finetuning makes models more robust to spurious cues.

MosaicML. 2023. Introducing mpt-30b: Raising the bar for open-source foundation models. Accessed: 2023-06-22.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, Xu Jiang, Karl Cobbe, Tyna Eloundou, Gretchen Krueger, Kevin Button, Matthew Knight, Benjamin Chess, and John Schulman. 2022. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint 2112.09332*.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback.

Cheng Peng, Xi Yang, Aokun Chen, Kaleb E Smith, Nima PourNejatian, Anthony B Costa, Cheryl Martin, Mona G Flores, Ying Zhang, Tanja Magoc, Gloria Lipori, Duane A Mitchell, Naykky S Ospina, Mustafa M Ahmed, William R Hogan, Elizabeth A Shenkman, Yi Guo, Jiang Bian, and Yonghui Wu. 2023. A study of generative large language model for medical research and healthcare.

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.

Rajkumar Ramamurthy, Prithviraj Ammanabrolu, Kianté Brantley, Jack Hessel, Rafet Sifa, Christian Bauckhage, Hannaneh Hajishirzi, and Yejin Choi. 2023. Is reinforcement learning (not) for natural language processing: Benchmarks, baselines, and building blocks for natural language policy optimization. *arXiv preprint 2210.01241*.

Oleksandr Romanko, Akhilesh Narayan, and Roy H. Kwon. 2023. Chatgpt-based investment portfolio selection.

Noah Shinn, Federico Cassano, Beck Labash, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning.

Aniket Kumar Singh, Suman Devkota, Bishal Lamichhane, Uttam Dhakal, and Chandra Dhakal. 2023. The confidence-competence gap in large language models: A cognitive study.

Lichao Sun, Yue Huang, Haoran Wang, Siyuan Wu, Qihui Zhang, Chujie Gao, Yixin Huang, Wenhan Lyu, Yixuan Zhang, Xiner Li, Zhengliang Liu, Yixin Liu, Yijue Wang, Zhikun Zhang, Bhavya Kailkhura, Caiming Xiong, Chaowei Xiao, Chunyuan Li, Eric Xing, Furong Huang, Hao Liu, Heng Ji, Hongyi Wang, Huan Zhang, Huaxiu Yao, Manolis Kellis, Marinka Zitnik, Meng Jiang, Mohit Bansal, James Zou, Jian Pei, Jian Liu, Jianfeng Gao, Jiawei Han, Jieyu Zhao, Jiliang Tang, Jindong Wang, John Mitchell, Kai Shu, Kaidi Xu, Kai-Wei Chang, Lifang He, Lifu Huang, Michael Backes, Neil Zhenqiang Gong, Philip S. Yu, Pin-Yu Chen, Quanquan Gu, Ran Xu, Rex Ying, Shuiwang Ji, Suman Jana, Tianlong Chen, Tianming Liu, Tianyi Zhou, William Wang, Xiang Li, Xiangliang Zhang, Xiao Wang, Xing Xie, Xun Chen, Xuyu Wang, Yan Liu, Yanfang Ye, Yinzhi Cao, Yong Chen, and Yue Zhao. 2024. Trustllm: Trustworthiness in large language models.

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*.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel R. Bowman. 2023. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13484–13508, Toronto, Canada. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models.

Fuzhao Xue, Yao Fu, Wangchunshu Zhou, Zangwei Zheng, and Yang You. 2023. To repeat or not to repeat: Insights from scaling llm under token-crisis. *arXiv preprint arXiv:2305.13230*.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *arXiv preprint arXiv:2305.10601*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*.

Xi Ye, Srinivasan Iyer, Asli Celikyilmaz, Ves Stoyanov, Greg Durrett, and Ramakanth Pasunuru. 2023. Complementary explanations for effective in-context learning.

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.

Banghua Zhu, Jiantao Jiao, and Michael I. Jordan. 2023. Principled reinforcement learning with human feedback from pairwise or k-wise comparisons. *arXiv preprint 2301.11270*.

Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2020. Fine-tuning language models from human preferences. *arXiv preprint 1909.08593*.