Reformatted Alignment

Anonymous ACL submission

Abstract

The quality of finetuning data is crucial for aligning large language models (LLMs) with human values. Current methods to improve data quality are either labor-intensive or prone to factual errors caused by LLM hallucinations. This paper explores elevating the quality of existing instruction data to better align with human values, introducing a simple and effective approach named REALIGN, which *reformats* the responses of instruction data into a format that better aligns with pre-established criteria and the collated evidence. This approach minimizes human annotation, hallucination, and the difficulty in scaling, remaining orthogonal to existing alignment techniques. Experimentally, REALIGN significantly boosts the general alignment ability, math reasoning, factuality, and readability of the LLMs.

Encouragingly, *without* introducing any additional data or advanced training techniques, and merely by reformatting the response, LLaMA-2-13B's mathematical reasoning ability on `GSM8K` can be improved **from 46.77% to 56.63%** in accuracy. Additionally, a mere 5% of REALIGN data yields a 67% boost in general alignment ability measured by the Alpaca dataset. This work highlights the need for further research into the *science* and *mechanistic interpretability* of LLMs. We have made the associated code, defined criteria, and data publicly accessible to support future studies at `https://anonymous.4open.science/r/ReAlign-9B3D`.

1 Introduction

Alignment has been witnessed to be an effective technique for aligning large language models (LLMs) to human values and human intent (Ouyang et al., 2022), which usually requires fine-tuning on a large amount of synthetic data derived from LLMs (Wang et al., 2023b; Honovich et al., 2023; Peng et al., 2023; Xu et al., 2023) or

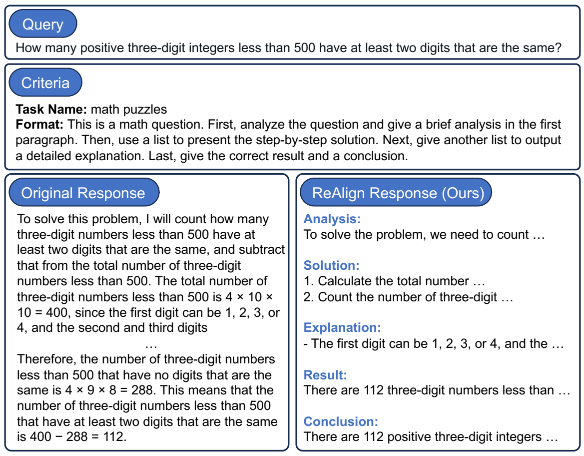


Figure 1: REALIGN realigns the original response with the pre-defined criteria to be a better format. The original response is from the `Open-Platypus` (Lee et al., 2023) dataset. The complete version is shown in Tab. 16.

human-annotated instruction data (Ouyang et al., 2022; Köpf et al., 2023).

Recent studies, notably by Zhou et al. (2023) highlight the critical role of instruction data quality in this process. Numerous works (Wang et al., 2023b; Zhou et al., 2023; Cao et al., 2023; Chen et al., 2023; Li et al., 2023a; Lu et al., 2023) have contributed to enhancing instruction quality by focusing on the diversity and complexity of input queries as well as the quality of responses. These efforts can be divided into two primary approaches. The first approach, advocated by Ouyang et al. (2022) and Touvron et al. (2023), involves the manual creation of high-quality data. Although this method creates complex queries and factually correct and highly readable responses, it is labor-intensive and challenging to scale. The second approach revolves around the automated extraction of high-quality instructions from existing datasets due to their extensive availability (Cao et al., 2023; Chen et al., 2023; Li et al., 2023a; Lu et al., 2023). However, this method inherits the limitations associated with distilled data, such as containing factually incorrect content (Ji et al., 2023; Gudibande et al., 2023) and the format and style of the generated response are often determined by distilled LLMs' preference.

In this paper, instead of focusing on the creation of instruction data from scratch, we investigate how existing instruction data can be made higher quality and better aligned with human values. We propose a simple and effective method, named REALIGN, which is orthogonal to the above existing approaches. Specifically, REALIGN necessitates a base instruction dataset, which can be sourced from extensive existing supervised datasets (e.g., GSM8K (Cobbe et al., 2021)), or publicly available instruction data compiled through various methods (e.g., Self-Instruct (Wang et al., 2023b), Evol-Instruct (Xu et al., 2023), and Self-Alignment (Li et al., 2023b)). The REALIGN process unfolds in three main steps. The first step involves **criteria definition** (§3.1), where humans define their preferences (e.g., the preferred format of responses) in various scenarios in the form of natural language. In this paper, we meticulously define criteria for 46 distinct scenarios. The second step, **retrieval augmentation** (§3.2), broadens the knowledge base for knowledge-intensive tasks like open-domain QA and fact verification. This is achieved by incorporating additional information, thereby improving the factuality and informativeness of responses. The final step, **reformatting** (§3.3), aims to re-align the responses with the pre-established criteria and the collated evidence, guaranteeing outputs that are both structured and substantiated. As demonstrated in Fig. 1, the realigned response provides a better format and a clearer chain of thoughts.

The underlying *philosophy* of REALIGN is to re-coordinate the roles of humans and LLMs in the alignment process, leveraging their complementary strengths – humans articulate their preferences, and LLMs, in turn, reconstruct instructions based on their generative power (e.g., instruction-following ability), without directly using distilled LLM knowledge. Through this collaborative synergy, we expect the generated instruction data to be not only more contextually precise but also more closely aligned with human preferences.

We operationalize this idea on five types of existing instruction data, where three are general datasets (i.e., Open-Platypus (Lee et al., 2023), No Robots (Rajani et al., 2023), and Alpaca (Taori et al., 2023)) and two are mathematical datasets (i.e., GSM8K (Cobbe et al., 2021) and MATH (Hendrycks et al., 2021)). The performance of REALIGN has been validated across various well-established benchmarks, including AlpacaEval (Li et al., 2023c), MT-Bench (Zheng et al., 2023), and Vicuna-Bench (Chiang et al., 2023) for general alignment, as well as GSM8K and MATH for mathematical reasoning. Additionally, it has also been evaluated for factuality and readability, demonstrating its proficiency. In particular, REALIGN significantly boosts math reasoning, even up to 9.86% on GSM8K for LLaMA-2-13B. Notably, we find that only 5% of the REALIGN data yields a 67% boost in general alignment ability compared to the full REALIGN data based on the Alpaca dataset, indicating that only a small amount of data is required to learn style and format.

2 Related Work

2.1 Instruction Creation

Creating instructional data significantly improves LLMs' alignment abilities. High-quality instruction generation traditionally depends on human annotation for tasks like query writing, response drafting, and preference indication. This approach produces premium open-source datasets (e.g., Open-Platypus (Lee et al., 2023) and OpenAssistant (Köpf et al., 2023)) and supports advanced LLMs (e.g., LIMA (Zhou et al., 2023) and LLaMA-2 (Touvron et al., 2023)), but it's hard to scale due to high labor costs and the need for domain-specific expertise. Many studies have explored using LLMs (e.g., GPT-3 (Brown et al., 2020) and GPT-4 (OpenAI, 2023)) to generate instruction data. Techniques like unnatural instructions (Honovich et al., 2023) and self-instruct (Wang et al., 2023b) utilize GPT-3's in-context learning with seed data to generate instructions, while evol-instruct (Xu et al., 2023) generates more complex and varied instructions through ChatGPT. Recently, training with self-generated data has yielded excellent results, achieving self-alignment (Li et al., 2023b; Yuan et al., 2024; Chen et al., 2024). While it can be easily scaled up, this approach inherits the drawbacks of LLMs (e.g., factual errors) (Gudibande et al., 2023). Our proposed method stands out by providing a way to automatically improve data quality with minimal effort and a significant reduction in factual errors.

2.2 Instruction Selection

After the discovery of "quality is all you need" (Zhou et al., 2023; Touvron et al., 2023), instruction selection has been paid attention to, aiming at selecting a small number of the highestquality samples from a large amount of instruction data as a training dataset. Cao et al. (2023) evaluates the dataset's quality by utilizing the evaluation dataset loss to fit the natural language indicators of the dataset. Chen et al. (2023) proposes to use ChatGPT directly to score the data, while Li et al. (2023b) proposes to score the data using the trained model directly to save costs. Lu et al. (2023) proposes to tag samples within SFT datasets based on semantics and intentions and define instruction diversity and complexity regarding tags to rank data. Li et al. (2023a) introduces a self-guided approach that utilizes a new indicator, Instruction-Following Difficulty (IDF), to score data by identifying gaps in a model's responses versus its autonomous generation capability. Liu et al. (2023) trains two scorers to evaluate the complexity of the instruction and the quality of the response, respectively, and then uses the embedding distance to determine the diversity to select high-quality data. However, the above works usually mine from distilled datasets because the large scale of distilled datasets is available, thereby inheriting the drawbacks of distilled data and suffering from the hallucination of LLMs.

2.3 Instruction Tuning

Instruction tuning aims to reinforce the model's instruction-following capabilities and align LLMs to human values. Early instruction tuning was designed to improve cross-task generalization capabilities, in which they usually scale up the quantity and the diversity of tasks (Mishra et al., 2022; Wei et al., 2022a; Sanh et al., 2022; Wang et al., 2022). Recent works no longer explicitly define tasks, but extend to more generalized capabilities, especially for scenarios of real-world questions (Wang et al., 2023b; Honovich et al., 2023; Peng et al., 2023; Xu et al., 2023). Differently, our work utilizes the future of the task to design a better format for it, which further improves the quality of the data.

3 REALIGN

Given a base instruction dataset $\mathcal{D} = \{(q_1, r_1), \cdots, (q_n, r_n)\}$, where q and r are the input query and response respectively, REALIGN aims to improve the quality of responses by three steps as shown in Fig. 2: (1) Criteria Definition: defining the criteria including tasks and formats for each task, (2) Retrieval Augmentation: retrieving relevant external information for the knowledge-intensive tasks, and (3) Reformatting: reformatting the original response based on the guidance consisting of hand-written format and the retrieved information. An overview of our method is shown in Fig. 2.

Group	Tasks
Generation	question generation; story generation; poem generation; email generation; data generation; text-to-text translation
Brainstorming	advice giving; recommendations; how-to generation; planning
Code	code correction; code simplification; explain code; text-to-code translation; code-to-code translation; language learning questions; code language classification; code-to-text-translation
Rewriting	instructional rewriting; language polishing; paraphrasing; text correction
Extraction	information extraction; keywords extraction; table extraction
Summarization	title generation; text summarization; note summarization
Conversation	open qa; closed qa; fact verification; value judgment; roleplay; explain answer
Education.	natural language tutor; exam problem tutor; ai tutor; math puzzles; fill in the blank
Classification	general classification; ordering; sentiment analysis; language classification; topic classification
Others	rejecting; others

Table 1: The category of tasks. "Education." denotes Specialized Educational Dialog.

Email Generation
It is an email-writing task. Here is a general guideline for creating a well-structured and professional email:

1. Subject Line: Write a clear and concise subject line that accurately summarizes the content of your email ...
2. Salutation: Begin your email with a formal salutation such as "Dear [Recipient's Name]," ...
3. Introduction: Start your email with a brief introduction ...
4. Body: This is the main content of your email ...
5. Politeness and Tone: Maintain a polite and respectful tone throughout your email ...
6. Closing: Conclude your email with a closing remark, such as "Thank you," or "Best regards," followed by your name ...
7. Signature: Include your full name, job title, and contact information (e.g., phone number, email address) ...
8. Attachments: If you need to include attachments, mention them ...
9. Proofread: Before sending the email, proofread it for any grammatical or spelling errors ...

The best emails are short, direct, professional, and scannable for the recipient. Follow a formal business email structure unless you have an established casual rapport with the recipient.

Table 2: An example of the format for the "email generation" task.

3.1 Criteria Definition

The predefined criteria consist of the tasks and the corresponding formats:

Tasks. Clearly defining tasks is crucial to subsequently devising tailored formats, as the optimal format varies across distinct tasks. In this paper, we follow Li et al. (2024) to define 46 different tasks $\{T_1, \cdots, T_{N=46}\}$, categorized into 10 major groups, as shown in Tab. 1. The detailed description for each task is shown in Tab. 11, §B. We also train a task classifier $\texttt{C}$, detailed in §C.

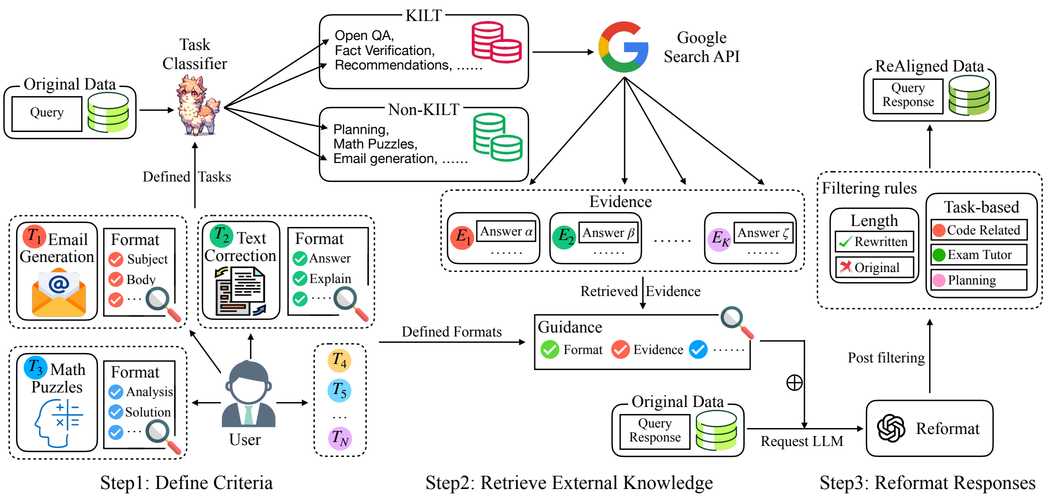


Figure 2: An overview of our REALIGN including three steps. KILT denotes Knowledge Intensive Language Tasks.

Format. Due to the distinct formatting requisites associated with diverse tasks, we meticulously devised tailored formats $\{F_1, \cdots, F_{N=46}\}$ for each task based on the task definition and description, encompassing considerations such as organizational structure, section content, and output modality. Specifically, we summarize the formatting deficiencies of GPT-4, Claude-2, and Bard in each task, and investigate user preferences for formatting regarding these tasks. Based on this information, we design tailored formats, which are more readable than the generic format. Each format has a task name and a detailed format description. We show an example of a format for “email generation” in Tab. 2 (The complete version is shown in Tab. 17).

In this step, we input query q_i to the task classifier $\mathtt{C}$ (detailed in §C) to acquire the category t_i:

$$t_i = \mathtt{C}(q_i),$$

and then obtain the corresponding format f_i.

3.2 Retrieval Augmentation

Knowledge-intensive language tasks (KILT), such as open-domain QA and fact verification, usually require large and external knowledge sources as the evidence to ensure the factuality (Petroni et al., 2021). Thus, we follow Petroni et al. (2021) to choose five knowledge-intensive tasks and use the query q_i to retrieve relevant information as our evidence. The tasks for retrieval augmentation are shown in Tab. 11. Specifically, we follow Chern et al. (2023) and use the Google Search API as our retriever $\mathtt{R}$ provided by Serper[1] to retrieve the most relevant search snippets included in the API's answer. We then parse the response to obtain different types of snippets such as answer boxes, knowledge graphs, and organic search results. Finally, we choose the top-k snippets and filter them as our evidence $E_i = e_{i1}, \cdots, e_{ik}$:

$$E_i = \mathtt{R}(q_i).$$

We show an example of a knowledge-intensive language task in Tab. 18, demonstrating that retrieval augmentation enables the response more factual and informative.

3.3 Reformatting

3.3.1 Rewriting

In this step, we leverage large language models (e.g., ChatGPT) to rewrite the response r_i based on the given format f_i and retrieved evidence E_i (for knowledge-intensive tasks). Since certain queries have additional requirements (e.g., specific formatting or specified information), an adaptive rewriting strategy is employed. This approach involves initially using LLMs to determine whether the format matches the query requirements. Subsequently, if it matches, the LLMs rewrite the response accordingly. We divide the tasks into two categories:

Non-knowledge-intensive tasks For the non-knowledge-intensive tasks, we decide to rewrite a part of the tasks. This decision stems from the observation that certain tasks are not amenable to a standardized format, exemplified by instances such as story generation and poem generation (see Tab. 11 for details). We guide LLMs to rewrite the original responses r_i, organizing the query q_i, original response r_i, and the format f_i together via the prompt in Tab. 14:

$$\hat{r}_i = \mathtt{LLM}(q_i, r_i, f_i),$$

[1] https://serper.dev/

where $\hat{r}_i$ is the reformatted response.

Knowledge-intensive tasks. For the knowledge-intensive tasks, we additionally utilize the retrieved evidence E_i compared to non-knowledge-intensive tasks. Specifically, We guide LLM to rewrite the original response r_i, organizing the query q_i, original response r_i, format f_i, and the retrieved evidence E_i together via the prompt in Tab. 15:

$$\hat{r}_i = \mathtt{LLM}(q_i, r_i, f_i, E_i).$$

3.3.2 Post-processing

Length filtering. We find that LLMs sometimes fail to reformat and only output the changed sentences, whose output length plummets. To filter out the data that fails to be reformatted, we keep the original response instead of using the reformatted response that is less than half the length of the original response.

Task-based filtering. To mitigate the problem of error propagation in task classification, we design filtering rules for specific tasks: (i) For code-related tasks (e.g., "`code correction`"), the keyword matching rule is employed to ascertain whether both the original and the reformatted versions contain code. If only one of the original responses or the reformatted response incorporates code, it signifies a failure in reformatting, and the original response is retained. (ii) For the "`exam problem tutor`" task, reformatted responses that do not contain the accurate result will not be accepted. (iii) For the "`planning`" task, if the query does not contain a planning-related keyword (e.g., plan or planning), the original answer is retained.

Finally, we could acquire the REALIGN dataset $\hat{\mathcal{D}} = \{(q_1, \hat{r}_1), \cdots, (q_n, \hat{r}_n)\}$.

4 Experiments

4.1 Datasets

For evaluation of general ability, we select two high-quality manual datasets and one distillation dataset for instruction tuning: (1) **`Open-Platypus`** (Lee et al., 2023) is an amalgamation of 11 open-source datasets, carefully curated to enhance LLM performance in STEM and logical domains. It consists of 25k questions, with around 90% written by humans and the rest generated by LLM. (2) **`No Robots`** (Rajani et al., 2023) is a high-quality dataset of 10k instructions and demonstrations created by skilled human annotators. (3) **`Alpaca`** (Taori et al., 2023) is an open-source instruction tuning dataset generated from text-davinci-003 (Ouyang et al., 2022) by the *Self-Instruct* (Wang et al., 2023b) method, containing 52k samples. Additionally, we also choose two manual datasets to evaluate the math reasoning after using REALIGN: (4) **`GSM8K`** (Cobbe et al., 2021) is a high-quality grade school math problems dataset created by human problem writers, consisting of 7.5k training problems and 1k test problems. (5) **`MATH`** (Hendrycks et al., 2021) is a dataset of mathematics competitions problems, including 7.5k for training and 5k for testing.

4.2 Models

We fine-tune two well-known open-source base models: **LLaMA-2-13B** (Touvron et al., 2023) and **Mistral-7B** (Jiang et al., 2023).

4.3 Evaluation

We evaluate REALIGN on general alignment and specific alignment ability including math reasoning, factuality, and readability.

4.3.1 General Alignment

To evaluate the general alignment ability, we follow Wang et al. (2023a) to employ the most widely recognized benchmarks, including: **AlpacaEval** (Li et al., 2023c), **MT-Bench** (Zheng et al., 2023), **Vicuna-Bench** (Chiang et al., 2023). Specifically, we use GPT-3.5 and Auto-J (detailed in §D) as the evaluators for AlpacaEval due to the cost of GPT-4, which has an extremely strong correlation with human (Li et al., 2024; Sun et al., 2024), and GPT-4 for MT-Bench and Vicuna-Bench.

4.3.2 Specific Alignment

We evaluate specific perspectives for alignment, including math reasoning, factuality, and readability.

Math Reasoning. To evaluate math reasoning, we finetune LLaMA-2-13B and Mistral-7B on `GSM8K` and `MATH` training datasets, respectively, and test afterward. The prompt template for training and testing is "`Question:\n{input}\n Answer:\nLet's think step by step.\n`". Since both datasets consist of math problems in the same style, we apply forced rewriting instead of adaptive, which does not require the determination of whether the query and format match but rather mandates a rewriting. We determine the accuracy by extracting the last number from the responses and comparing it directly to the ground truth.

Factuality. To evaluate the factuality, we randomly select 100 cases from the Natural Questions dataset (NQ) (Kwiatkowski et al., 2019), a public

Model	Dataset	AlpacaEval		MT-Bench			Vicuna-Bench	Overall
		GPT-3.5 (%)	Auto-J	First	Second	Average		
LLaMA-2-13B	Open-Platypus	55.71	**4.93**	6.69	**5.16**	**5.94**	8.28	6.18
	+ REALIGN	**58.20**	4.81	**6.89**	4.86	5.88	**8.45**	**6.24**
	No Robots	44.25	4.56	5.80	5.15	5.48	7.31	5.44
	+ REALIGN	**48.13**	**4.65**	**6.04**	**5.20**	**5.62**	**7.51**	**5.65**
	Alpaca	46.08	4.65	5.55	4.16	4.86	6.55	5.17
	+ REALIGN	**49.19**	**4.74**	**5.83**	**4.71**	**5.27**	**6.84**	**5.44**
Mistral-7B	Open-Platypus	59.63	5.15	7.29	5.88	6.58	**8.96**	6.66
	+ REALIGN	**61.33**	**5.15**	**7.43**	**6.18**	**6.80**	8.86	**6.74**
	No Robots	44.22	4.62	5.95	**4.94**	5.44	7.32	5.45
	+ REALIGN	**48.26**	**4.76**	**6.14**	4.79	**5.46**	**7.68**	**5.68**
	Alpaca	51.24	4.77	6.06	**5.26**	5.66	7.14	5.67
	+ REALIGN	**52.67**	**4.82**	**6.50**	5.03	**5.76**	**7.33**	**5.79**

Table 3: The results of the general alignment ability on the original datasets and the REALIGN datasets. **Bold** indicates the best result on each dataset. For AlpacaEval, GPT-3.5 denotes the winning rate obtained by using GPT-3.5 as the evaluator. Auto-J denotes the quality of the model's responses evaluated in a point-wise manner using Auto-J (Li et al., 2024). For Overall, we calculate the average of AlpacaEval's winning rate for GPT-3.5 divided by 10, the results for Auto-J, the average MT-Bench results, and the results for Vicuna-Bench.

Q&A dataset rich in fact-based queries and their verified answers. We employ both GPT-4 and human evaluation in our experiment. Specifically, GPT-4 is used to rate these instances on a factuality scale of 1 to 10, considering the question, the response, and the ground truth (referred to as the factuality score). The evaluation prompt is shown in Tab. 20. For the human evaluation, we compare the model's response to the ground truth based on the question, assigning a True or False. The results we present are the proportion of responses that were assigned as True.

Readability. To evaluate the readability, we compare a model trained on the original dataset against another model on the dataset enhanced with REALIGN, using human and GPT-4 evaluations on the Vicuna-Bench dataset (Chiang et al., 2023). Since the vicuna bench contains fewer complex questions (e.g., code and math), the judge can focus on the format rather than the result. We design an evaluation prompt prioritizing readability, refer to Tab. 19, and randomize response positions to eliminate bias.

4.4 Results

REALIGN Improves General Alignment Ability. From Tab. 3, we can see an increase in almost all three datasets and benchmarks on both the LLaMA-2-13B and Mistral-7B models, showing that REALIGN can significantly improve models' response quality and conversation ability. Additionally, from the results of MT-Bench, we can see that REALIGN can improve the performance of the second turn of conversations on half the datasets even though it only rewrites the first turn of the instruction data.

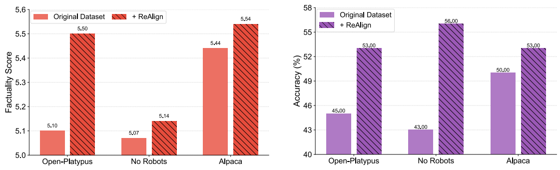
Model	Dataset	GSM8K	MATH	Overall
LLaMA-2-13B	GSM8K	46.77	5.02	25.90
	+ REALIGN	**56.63**	**5.46**	**31.05**
	MATH	14.48	6.14	10.31
	+ REALIGN	**25.17**	**7.14**	**16.16**
Mistral-7B	GSM8K	61.25	10.64	35.95
	+ REALIGN	**68.16**	**12.70**	**40.43**
	MATH	28.35	13.18	20.77
	+ REALIGN	**38.21**	**15.30**	**26.76**

Table 4: The results of math reasoning on GSM8K, MATH and them + REALIGN. We test models on both GSM8K and MATH test sets. We report the accuracy by exact matching. **Bold** indicates the best result.

REALIGN Can Boost Math Reasoning. As shown in Tab. 4, REALIGN can boost the math reasoning on both datasets, even up to 9.86% on GSM8K using LLaMA-2-13B. Remarkably, REALIGN enhances generalization, demonstrated by cross-domain performance boosts. Specifically, training models using the MATH dataset yields notable improvements in the GSM8K test results, and vice versa. For instance, it has been observed that training on the MATH dataset can augment GSM8K performance by 10.69% based on LLaMA-2-13B. We explore the possible reasons in §E.

REALIGN Can Enhance Factuality. Fig. 3 shows REALIGN elevates the factuality, highlighting its efficacy. This improvement is probably due to the addition of retrieval augmentation.

REALIGN Can Improve Readability. As shown in Fig. 4, we see that REALIGN can improve

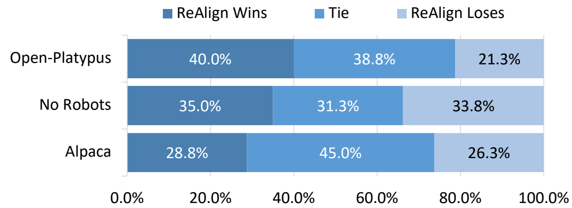


(a) GPT-4 Evaluation (b) Human Evaluation

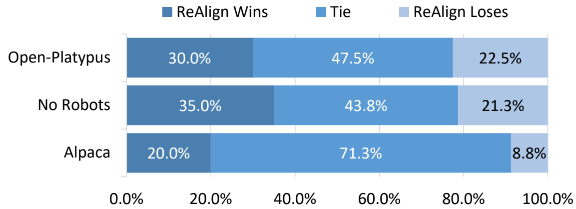
Figure 3: The results of the factuality score evaluated by GPT-4 and human.

Dataset	Response Len.	REALIGN %
Open-Platypus	224.92 → 206.91	28.5%
No Robots	211.99 → 211.54	15.9%
Alpaca	65.51 → 72.38	29.9%
GSM8K	130.59 → 327.65	100%
MATH	243.73 → 375.35	100%

Table 5: The datasets analysis includes the changes between original datasets and them + REALIGN. Response Len. is the average number of tokens of the responses. REALIGN % denotes the percentage of successful reformatting after the adaptive rewriting.



(a) GPT-4 Judgments



(b) Human Judgments

Figure 4: The readability win-rate of the original dataset + REALIGN against the original dataset based on LLaMA-2-13B, judged by GPT-4 and human.

the readability of three datasets, especially in the Open-Platypus dataset (i.e., 18.7% improvements in GPT-4 judgments). It demonstrates that designing different formats for different tasks and reformatting them can improve readability. In addition, human tends to provide more ties for judgments compared to GPT-4. A possible reason is that REALIGN can provide better structure, causing GPT-4 to be limited to surface formats ignoring content and deep structure. In contrast, humans can read more carefully not being limited to surface formats.

4.5 Analysis

4.5.1 Datasets Analysis

First, we compare the change in the length of responses (i.e., the number of tokens) between the original datasets and the REALIGN datasets, finding that the math reasoning datasets GSM8K and MATH become longer and the other datasets are not significant changes (see Tab. 5). Second, we calculate the percentage of responses for which the adaptive rewriting method selects rewrite by edit distance (the results are shown in Tab 5). Specifically, we compute the edit distance on a word basis, then divide the edit distance by the length of the longest of the original and rewritten responses to obtain the edit rate, and finally record those with an edit rate greater than 0.2 as rewritten. For GSM8K and MATH datasets, all data are ReAligned as adaptive rewriting was not applied to them.

4.5.2 Why Does REALIGN Boost Math?

A series of experiments and analyses yield several important insights (see the complete version in §E):

(1) A well-organized format is more beneficial than merely providing step-by-step explanations. As shown in Tab. 7, merely providing a step-by-step explanation is insufficient without a well-organized format.

(2) Length is not all you need. To analyze the impact of the length of reasoning steps, we hypothesize that the longer the response, the more extensive the reasoning steps involved. As shown in Tab. 7 and Tab. 9, length is not the determining factor; rather, a well-organized format can lead to more substantial gains.

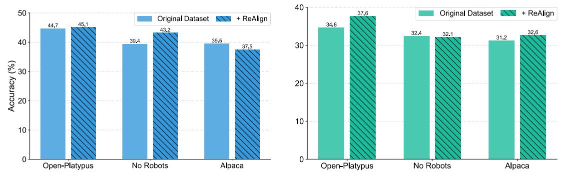
(3) Human value is the most important principle in designing formats. As shown in Tab. 8, formats that align with human habits and are easier to understand yield better performance. Therefore, we also advocate that the development of large language models should move closer to user values.

4.5.3 Ablation Studies

We rewrite two variants of the Open-Platypus dataset and train them based on LLaMA-2-13B for ablation studies:

(1) **W/o Retrieval Augmentation**: We remove the retrieval augmentation from REALIGN and rewrite all tasks without evidences. As shown in Tab. 6, the general alignment ability, knowledge ability, and factuality score (FS) are reduced, indicating the effectiveness of retrieval augmentation. Interestingly, the FS metrics are higher without RAG than in the original dataset, suggesting that REALIGN also has the potential to improve the factuality.

(2) **W/o Adaption**: We remove the adaptive rewriting from REALIGN and use force rewriting. Tab. 6



(a) BBH (b) AGIEval

Figure 5: The results of the knowledge abilities, including the Big Bench Hard (BBH) (3-shot), and AGIEval (zero-shot). We evaluate the abilities across the Open-Platypus, No Robots, and Alpaca datasets, based on LLaMA-2-13B.

Dataset	General Align.	Know. Ab.	FS
Open-Platypus	6.18	39.65	5.1
+ REALIGN	**6.24**	**41.35**	5.5
W/o RA	6.18	40.6	5.3
W/o Adaption	6.17	39.8	**5.6**

Table 6: Ablation study results show removing retrieval augmentation ("W/o RA") and removing adaptive rewriting ("W/o Adaption") in REALIGN. "General Align." and "Know. Ab." denotes the average results of general alignment ability and Knowledge Ability. FS denotes Factuality Score. **Bold** denotes the best.

shows the general alignment and knowledge ability decrease. This may be because forced rewriting, while making the responses more structured, ignores the question's requirements, weakening the instruction-following ability. In addition, FS has increased, probably because forced rewriting leads to a larger amount of REALIGN data, introducing more retrieved knowledge and boosting factuality.

4.5.4 Alignment Tax

When the model is fine-tuned on the REALIGN dataset, a question worth exploring is whether there is a drop in knowledge ability even as alignment ability improves. To evaluate the knowledge ability, we follow (Mitra et al., 2023) to employ the following benchmarks: **Big Bench Hard (BBH)** (Suzgun et al., 2022) and **AGIEval** (Zhong et al., 2023), which is multiple choices knowledge-intensive QA task. As shown in Fig. 5, we can see that REALIGN has little effect on the knowledge-based tasks, indicating that our approach does not impair the knowledge in the original dataset. It is worth noting that in some cases REALIGN will also provide a significant boost to knowledge, such as Open-Platypus on AGIEval. Possible reasons are that a well-defined format can facilitate the accuracy of the knowledge-based tasks (Wei et al., 2022b) and that retrieving external information can augment knowledge.

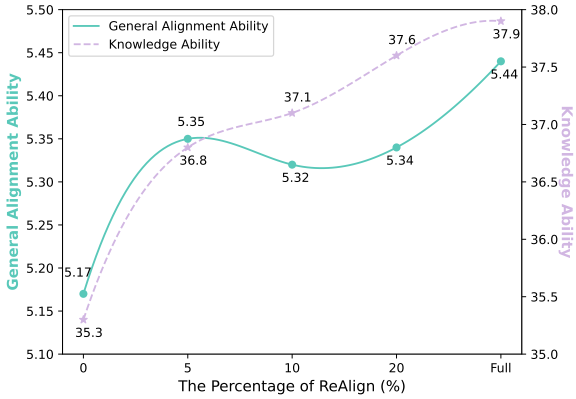


Figure 6: The scaling trends in REALIGN data percentage, including general alignment ability and knowledge ability. We conduct the experiment in the Alpaca dataset based on LLaMA-2-13B.

4.5.5 The Scaling Law of REALIGN

We experiment to explore the impact of the number of REALIGN. Specifically, we randomly sample a $k\%$ ($k = 0, 5, 10, 20, \text{Full}$, with Full being 29.9%) of REALIGN Alpaca data, and fill in the remainder with original responses. The original Alpaca dataset corresponds to 0%. Interestingly, we find that only 5% of the REALIGN data yields a 67% boost in general alignment ability compared to the entire REALIGN data (see Fig. 6). This suggests that only a small amount of data is required to learn style and format, to expose the knowledge and capabilities that were already acquired during pretraining (Zhou et al., 2023). Additionally, the knowledge capability continues to improve as the amount of REALIGN data improves.

4.5.6 Case Study

We show a case from the MT-Bench test set in Tab. 10. This example shows that the response given by the REALIGN model has a better format.

5 Conclusion

In this work, we propose REALIGN, a simple and effective method for alignment, which automatically improves the quality of the existing instruction datasets while minimizing labor costs and hallucinations. We create five new high-quality datasets from Open-Platypus (Lee et al., 2023), No Robots (Rajani et al., 2023), Alpaca (Taori et al., 2023), GSM8K (Cobbe et al., 2021), and MATH (Hendrycks et al., 2021) and high-quality manual-written natural language formats. Experiments demonstrate that REALIGN significantly boosts general alignment ability, math reasoning, factuality, and readability without impairing knowledge ability. Last, we release the code, defined criteria, and data to facilitate future research.

Limitations

First, our approach relies on the ability to reformatting models, which is currently less effective in open-source models (e.g., LLaMA2 (Touvron et al., 2023)) but more costly in closed-source models (e.g., GPT-4 (OpenAI, 2023)). Second, the task categories we define cannot cover all tasks in reality, as real questions may be more complex and involve multiple tasks. Therefore, it is necessary to define more tasks and formats for a wide range of diverse and regional scenarios. Third, applying REALIGN only to single-turn conversations has the potential to hurt the alignment ability of the second-turn conversations, hence extending REALIGN to multi-turn conversation would also be valuable. Last, we explore the reasons behind REALIGN's success superficially, and thus will further explore the science and mechanistic interpretability behind it in the future.

Ethics Statement

We take ethical considerations very seriously. In this paper, both the datasets and models are publicly available and have been widely adopted by researchers. We ensure that the findings and conclusions of this paper are reported accurately and objectively.

References

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *NeurIPS*.

Yihan Cao, Yanbin Kang, and Lichao Sun. 2023. Instruction mining: High-quality instruction data selection for large language models. *arXiv Preprint*.

Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, and Hongxia Jin. 2023. Alpagasus: Training a better alpaca with fewer data. *arXiv Preprint*.

Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. 2024. Self-play fine-tuning converts weak language models to strong language models. *arXiv preprint arXiv:2401.01335*.

I-Chun Chern, Steffi Chern, Shiqi Chen, Weizhe Yuan, Kehua Feng, Chunting Zhou, Junxian He, Graham Neubig, Pengfei Liu, et al. 2023. Factool: Factuality detection in generative ai–a tool augmented framework for multi-task and multi-domain scenarios. *arXiv preprint arXiv:2307.13528*.

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.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Diego Granziol, Stefan Zohren, and Stephen Roberts. 2022. Learning rates as a function of batch size: A random matrix theory approach to neural network training. *J. Mach. Learn. Res.*, 23:173:1–173:65.

Arnav Gudibande, Eric Wallace, Charlie Snell, Xinyang Geng, Hao Liu, Pieter Abbeel, Sergey Levine, and Dawn Song. 2023. The false promise of imitating proprietary llms. *arXiv Preprint*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *NeurIPS*.

Or Honovich, Thomas Scialom, Omer Levy, and Timo Schick. 2023. Unnatural instructions: Tuning language models with (almost) no human labor. In *ACL*, pages 14409–14428, Toronto, Canada. Association for Computational Linguistics.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Comput. Surv.*, 55(12):248:1–248:38.

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. *arXiv preprint arXiv:2310.06825*.

Mingyu Jin, Qinkai Yu, Haiyan Zhao, Wenyue Hua, Yanda Meng, Yongfeng Zhang, Mengnan Du, et al. 2024. The impact of reasoning step length on large language models. *arXiv preprint arXiv:2401.04925*.

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*, 7:453–466.

Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi-Rui Tam, Keith Stevens, Abdullah Barhoum, Nguyen Minh Duc, Oliver Stanley, Richárd Nagyfi, Shahul ES, Sameer Suri, David Glushkov, Arnav Dantuluri, Andrew Maguire, Christoph Schuhmann, Huu Nguyen, and Alexander Mattick. 2023. Openassistant conversations – democratizing large language model alignment. *arXiv Preprint*.

Ariel N. Lee, Cole J. Hunter, and Nataniel Ruiz. 2023. Platypus: Quick, cheap, and powerful refinement of llms. *arXiv Preprint*.

Junlong Li, Shichao Sun, Weizhe Yuan, Run-Ze Fan, Hai Zhao, and Pengfei Liu. 2024. Generative judge for evaluating alignment. In *ICLR*.

Ming Li, Yong Zhang, Zhitao Li, Jiuhai Chen, Lichang Chen, Ning Cheng, Jianzong Wang, Tianyi Zhou, and Jing Xiao. 2023a. From quantity to quality: Boosting llm performance with self-guided data selection for instruction tuning. *arXiv Preprint*.

Xian Li, Ping Yu, Chunting Zhou, Timo Schick, Luke Zettlemoyer, Omer Levy, Jason Weston, and Mike Lewis. 2023b. Self-alignment with instruction backtranslation. *arXiv Preprint*.

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023c. Alpacaeval: An automatic evaluator of instruction-following models. https://github.com/tatsu-lab/alpaca_eval.

Wei Liu, Weihao Zeng, Keqing He, Yong Jiang, and Junxian He. 2023. What makes good data for alignment? a comprehensive study of automatic data selection in instruction tuning. *arXiv preprint arXiv:2312.15685*.

Keming Lu, Hongyi Yuan, Zheng Yuan, Runji Lin, Junyang Lin, Chuanqi Tan, Chang Zhou, and Jingren Zhou. 2023. #instag: Instruction tagging for analyzing supervised fine-tuning of large language models. *arXiv Preprint*.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2022. Cross-task generalization via natural language crowdsourcing instructions. In *ACL*, pages 3470–3487, Dublin, Ireland. Association for Computational Linguistics.

Arindam Mitra, Luciano Del Corro, Shweti Mahajan, Andres Codas, Clarisse Simoes, Sahaj Agarwal, Xuxi Chen, Anastasia Razdaibiedina, Erik Jones, Kriti Aggarwal, Hamid Palangi, Guoqing Zheng, Corby Rosset, Hamed Khanpour, and Ahmed Awadallah. 2023. Orca 2: Teaching small language models how to reason. *arXiv preprint arXiv:2311.11045*.

Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah. 2023. Orca: Progressive learning from complex explanation traces of gpt-4. *arXiv preprint arXiv:2306.02707*.

OpenAI. 2023. Gpt-4 technical report. *arXiv Preprint*.

Long Ouyang, Jeffrey 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 F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *NeurIPS*.

Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. 2023. Instruction tuning with gpt-4. *arXiv Preprint*.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, Vassilis Plachouras, Tim Rocktäschel, and Sebastian Riedel. 2021. KILT: a benchmark for knowledge intensive language tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2523–2544, Online. Association for Computational Linguistics.

Nazneen Rajani, Lewis Tunstall, Edward Beeching, Nathan Lambert, Alexander M. Rush, and Thomas Wolf. 2023. No robots. https://huggingface.co/datasets/HuggingFaceH4/no_robots.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Tali Bers, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M. Rush. 2022. Multitask prompted training enables zero-shot task generalization. *arXiv Preprint*.

Shichao Sun, Junlong Li, Weizhe Yuan, Ruifeng Yuan, Wenjie Li, and Pengfei Liu. 2024. The critique of critique. *arXiv preprint arXiv:2401.04518*.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, , and Jason Wei. 2022. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv Preprint*.

Guan Wang, Sijie Cheng, Xianyuan Zhan, Xiangang Li, Sen Song, and Yang Liu. 2023a. Openchat: Advancing open-source language models with mixed-quality data. *arXiv preprint arXiv:2309.11235*.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023b. Self-instruct: Aligning language models with self-generated instructions. In *ACL*, pages 13484–13508, Toronto, Canada. Association for Computational Linguistics.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Gary Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Maitreya Patel, Kuntal Kumar Pal, Mehrad Moradshahi, Mihir Parmar, Mirali Purohit, Neeraj Varshney, Phani Rohitha Kaza, Pulkit Verma, Ravsehaj Singh Puri, Rushang Karia, Shailaja Keyur Sampat, Savan Doshi, Siddhartha Mishra, Sujan Reddy, Sumanta Patro, Tanay Dixit, Xudong Shen, Chitta Baral, Yejin Choi, Noah A. Smith, Hannaneh Hajishirzi, and Daniel Khashabi. 2022. Super-naturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks. *arXiv Preprint*.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022a. Finetuned language models are zero-shot learners. *arXiv Preprint*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022b. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. 2023. Wizardlm: Empowering large language models to follow complex instructions. *arXiv Preprint*.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. 2024. Self-rewarding language models. *arXiv preprint arXiv:2401.10020*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric. P Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *arXiv Preprint*.

Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. 2023. Agieval: A human-centric benchmark for evaluating foundation models. *arXiv preprint arXiv:2304.06364*.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. 2023. Lima: Less is more for alignment. *arXiv Preprint*.

A Implementation Details

For retrieval augmentation, we select the top-5 evidence from the retrieval results. For reformatting, We guide gpt-3.5-turbo-1106 to reformat the responses. We set the temperature as 0.3, the top-p as 0.1, and the target length as 2048. Additionally, we generate two rewrite results at a time and choose the longest one, hence n is set to 2. For training, we fine-tune the models based on the LLaMA-2-13B (Touvron et al., 2023) and Mistral-7B (Jiang et al., 2023) for 5 epochs on the `Open-Platypus` dataset, 20 epochs on the `No Robots` and `Alpaca` datasets, and 3 epochs on the `GSM8K` and `MATH` datasets, using the AdamW optimizer with a sequence length of 4,096 tokens. The batch size is 64 for the `Open-Platypus`, `No Robots`, and `Alpaca` datasets, and 128 for the `GSM8K` and `MATH` datasets. The AdamW optimizer's hyperparameters are set as follows: $\beta_1 = 0.9$, $\beta_2 = 0.95$, $\epsilon = 10^{-5}$, and weight decay of 0.1. We employ a cosine learning rate schedule with a maximum learning rate of 5.7×10^{-5} for the `Open-Platypus` dataset, 6.25×10^{-5} for the `No Robots` dataset, 6.55×10^{-5} for the `Alpaca` dataset, and 1×10^{-5} for the `GSM8K` and `MATH` datasets, which decays to 10% of the maximum value. Following Wang et al. (2023a) and Granziol et al. (2022), the learning rate is scaled proportionally to the square root of the batch size. All models are trained on 8 NVIDIA A100 80G GPUs.

B Task Description

The task descriptions mentioned in §3.1 and whether they are retrieved and rewritten are exhibited in Tab. 11.

C Training Details of Task Classifier

In this section, we describe the training details of the task classifier mentioned in §3.1.

In real scenarios, user questions can be quite long and complex (with more than 1k words), while traditional BERT-like models only have a context length of 512 tokens, urging us to train a large language model for classification. Following Li et al. (2024), we convert the classification task into a generation task, which directly generates the task name given a question with the prompt as shown in Tab. 13. Specifically, we manually label about 33 questions for each kind of task from `LIMA` (Zhou et al., 2023), `ShareGPT` (Zheng et al., 2023), and `Alpaca` (Taori et al., 2023) datasets. For tasks with less training data, we use ChatGPT to generate a portion of the questions. Then, we divide questions in a 9:1 train/test split (Tab. 12). We train the task classifier from LLaMA-2-13B (Touvron et al., 2023), and set the max sequence length as 2,048, epochs as 20, and batch size as 16. We set the initial learning rate to 2e-5 and consine decaying to 0 by the end of training with warmup steps as 10. The optimizer is AdamW with $\beta_1 = 0.9, \beta_2 = 0.95$. The loss is only calculated on the output end as well. The accuracy and F1 of the final task classifier on the test set are 78.32% and 81.59%, respectively.

D The Description of Auto-J

Auto-J (Li et al., 2024) is an open-source generative judge designed to evaluate LLMs based on their alignment with human preferences, which is the best critique model besides GPT-4 (Sun et al., 2024). Auto-J stands out due to its generality, being trained on real-world user queries and responses from various LLMs across 58 scenarios. It offers flexibility by enabling both pairwise comparison and single-response evaluation through prompt adjustments. Additionally, Auto-J enhances reliability and encourages human participation in the evaluation process by offering detailed natural language critiques, improving interpretability.

E Why Does REALIGN Boost Math? (Complete Version)

By carefully observing the cases, we speculate that the reason for the improvements in math reasoning may stem from the easier-to-understand format, more detailed explanations (Mukherjee et al., 2023), or length (Jin et al., 2024). To further explore the reasons, we merely incorporate detailed step-by-step explanations without including a complete format. These explanations are generated using gpt-3.5-turbo-1106, with the prompts used detailed in Tab. 21 and Tab. 22. The results are shown in Tab. 7.

Moreover, to explore the impact of various formats, we experiment with two other formats based on REALIGN: (1) The first requires separating natural language and calculation, meaning that the natural language does not include the calculation process and mathematical computations are expressed separately; (2) The second variant requires the use of special markers ‘<<>>’ in equations based on the format of REALIGN, for example,

Test Dataset	Training Dataset	Model: LLaMA-2-13B	Model: Mistral-7B	Overall
GSM8K	GSM8K	46.77	61.25	54.01
	+ Explanation	48.60	53.37	50.99
	+ REALIGN	**56.63**	**68.16**	**62.40**
MATH	MATH	6.14	13.18	9.66
	+ Explanation	**7.30**	13.94	10.62
	+ REALIGN	7.14	**15.30**	**11.22**

Table 7: The results of math reasoning on GSM8K, MATH and them + explanation or REALIGN. We report the accuracy by exact matching. **Bold** indicates the best result.

$<< 1 + 2 = 3 >>$. We merely create these variants on GSM8K due to the cost of ChatGPT API. The results are shown in Tab. 8

From the results of the above experiments (see Tab. 7 and Tab. 8), we can derive insights below:

Insights 1: A well-organized format is more beneficial than merely providing step-by-step explanations. As shown in Tab. 7, we can find that merely providing a detailed explanation is insufficient without a well-structured format and may even result in performance inferior to the original dataset (i.e., The results on GSM8K based on Mistral-7B). For the more complex MATH dataset, detailed explanations still play a significant role. However, A well-organized structure may further enhance their effectiveness.

Insights 2: Length is not all you need. Jin et al. (2024) suggests that longer reasoning steps can enhance math reasoning capabilities. To further analyze the impact of the length of reasoning steps, we hypothesize that the longer the response, the more extensive the reasoning steps involved. Specifically, we calculate the average length that is shown in Tab. 9. We can see that the average length of “GSM8K + Explanation” exceeds that of “GSM8K” by more than double, and is even longer than “GSM8K + ReAlign”. However, its performance is significantly inferior to “GSM8K + ReAlign”. Additionally, the average length of "MATH + Explanation" is shorter than "MATH + ReAlign", yet it demonstrates superior performance on LLaMA-2-13B. These findings suggest that length is not the determining factor; rather, a well-organized format can lead to more substantial gains.

Dataset	Model: LLaMA-2-13B	Model: Mistral-7B	Overall
GSM8K	46.77	61.25	54.01
+ REALIGN (Separate.)	55.57	62.09	58.83
+ REALIGN (‘<<>>’)	**57.01**	63.61	60.31
+ REALIGN	56.63	**68.16**	**62.40**

Table 8: The results of math reasoning on GSM8K and them + different formats of REALIGN. We report the accuracy by exact matching. “Separate.” denotes the first variant that separates natural language and calculation. “<<>>” denotes the second variant that requires the use of special markers ‘<<>>’ in equations based on the format of REALIGN. **Bold** indicates the best result.

Dataset	Response Len.
GSM8K	130.59
+ Explanation	341.61
+ REALIGN	327.65
MATH	243.73
+ Explanation	293.57
+ REALIGN	375.35

Table 9: The response length of original datasets and them + Explanation and REALIGN. Response Len. is the average number of tokens of the responses.

Insights 3: Human value is the most important principle in designing formats. As shown in Tab. 8, we can see that, firstly, expressing natural language and mathematical calculations together (REALIGN) performs better. This approach is more in line with human habits and preferences, making it easier for users to understand. Secondly, adding special markers “<<>>” to equations can diminish performance, potentially because this format does not align with the customary practices of human writing and usage, making it less intuitive to understand. Therefore, the most important principle in designing formats is human value. A good format is not only about superior performance but also about user preference (i.e., formats that are easier for users to understand). Following this principle in designing criteria, we observed a significant improvement in our math reasoning abilities, indicating that formats easier for users to understand can also lead to better performance. Therefore, we also advocate that the development of large language models should move closer to user values.

F Prompts

The reformatting prompts for non-knowledge-intensive tasks and knowledge-intensive tasks are shown in Tab. 14 and Tab. 15. The complete version of Fig. 1 and Tab. 2 are shown in Tab. 16 and Tab. 17, respectively. The example of a knowledge-intensive task is shown in Tab. 18. The evaluation prompts for readability and factuality are shown in Tab. 19 and Tab. 20, respectively. The prompts for generating explanations for GSM8K and MATH datasets are shown in Tab. 21 and Tab. 22.

Query: Benjamin went to a bookstore and purchased a variety of books. He bought 5 copies of a sci-fi novel, each priced at $20, 3 copies of a history book priced at $30 each, and 2 copies of a philosophy book for $45 each. What was the total cost of his purchases?
Response from Original Model: To find the total cost of Benjamin's purchases, I need to add up the prices of all the books he bought. I can use the information given in the problem to set up an equation: Total cost = Price of sci-fi novel * Number of copies + Price of history book * Number of copies + Price of philosophy book * Number of copies Plugging in the values from the problem, I get: Total cost = $20 * 5 + $30 * 3 + $45 * 2 Simplifying, I get: Total cost = $100 + $90 + $90 Adding, I get: Total cost = $280 So, the total cost of Benjamin's purchases was $280.
Response from REALIGN Model: To find the total cost of Benjamin's purchases, we need to add up the prices of the books he bought. Step-by-step solution: 1. Calculate the cost of the sci-fi novels: 5 copies at $20 each = $100 2. Calculate the cost of the history books: 3 copies at $30 each = $90 3. Calculate the cost of the philosophy books: 2 copies at $45 each = $90 Detailed explanation: The total cost of Benjamin's purchases is the sum of the costs of the sci-fi novels, the history books, and the philosophy books: $100 + $90 + $90 = $280. Correct result and conclusion: The total cost of Benjamin's purchases was $280.

Table 10: An example of the response from a model trained on the original `Open-Platypus` dataset (denoted as Original Model) and the response from a model trained on the REALIGN `Open-Platypus` dataset (denoted as REALIGN Model). We highlight the content in the better format given by the REALIGN Model in green compared with the content given by the Original Model in red.

Task name	Description	Retrieval	Rewriting
	Generation		
question generation	Write some questions based on the given description.	✗	✓
story generation	Write a story based on the given description.	✗	✗
poem generation	Write a poem based on the given description.	✗	✗
email generation	Write an email based on the given description.	✗	✓
data generation	Generate data based on the given description.	✗	✓
text-to-text translation	Translate the given text into another language.	✗	✗
	Brainstorming		
advice giving	Respond well to users when they seek advice.	✗	✗
recommendations	Give recommendations to users.	✓	✓
how-to generation	Give relevant and complete answer when users ask 'how to do' something.	✓	✓
planning	Write a plan for an event or activity.	✗	✓
	Code		
code correction	Correct the potential errors in a piece of code.	✗	✓
code simplification	Rewrite a piece of code to make it more concise and easy to understand.	✗	✗
explain code	Write an explanation for a piece of code.	✗	✓
text-to-code translation	Write a piece of code based on the given description.	✗	✓
code-to-code translation	Convert the given code into another programming language.	✗	✓
language learning questions	Write an answer for the given question about programming language learning.	✗	✓
code language classification	Classify the programming language for the given code.	✗	✓
code-to-text-translation	Write a document for the given code.	✗	✓
	Rewriting		
instructional rewriting	Rewrite a given text with a specific instruction.	✗	✓
language polishing	Polish a piece of text to make it more fluent, natural, and readable.	✗	✓
paraphrasing	Paraphrase a given text.	✗	✗
text correction	Correct the potential errors in a piece of text.	✗	✓
	Extraction		
information extraction	Extract one or multiple user-specified categories of information from a piece of text attached in the user's query.	✗	✓
keywords extraction	Extract the keywords from a piece of text.	✗	✓
table extraction	Generate a table include the key information from a piece of text attached in the user's query.	✗	✗
	Summarization		
title generation	Generate a title for the given text or based on a description of the work.	✗	✗
text summarization	Write a summary for a piece of text.	✗	✗
note summarization	Write a note to summarize a piece of text.	✗	✗
	Conversation		
open qa	The user's query is an open domain question with no attached passage or article.	✓	✓
closed qa	Answer the questions that can be directly answered by the attached passage.	✗	✓
fact verification	Verify if the given fact is true or false.	✓	✓
value judgment	Provide a value judgment on a given topic or statement.	✗	✓
roleplay	Pretend to be a specific person, character, profession or identity, and complete the required task on this basis.	✗	✗
explain answer	Explain something the user wants to know.	✓	✓
	Specialized Educational Dialog		
natural language tutor	Write an answer for the given question about natural language learning.	✗	✓
exam problem tutor	Solve an exam question (like fill-in-the-blank, multiple choice, problem solving, etc) with no math involved.	✗	✓
ai tutor	Write an answer for the given question about machine learning, artificial intelligence or language model.	✗	✓
math puzzles	Write an answer with the step-by-step reasoning process for a math question.	✗	✓
fill in the blank	Complete the missing parts with the most appropriate words to make the text coherent and meaningful.	✗	✓
	Classification		
general classification	Classify one or multiple objects given by the user into the specified categories.	✗	✓
ordering	Sort some things, according to some criteria.	✗	✓
sentiment analysis	Identify and categorize the subjective opinions, attitudes, and feelings of the writer towards a particular subject.	✗	✓
language classification	Classify the language for the given text.	✗	✓
topic classification	Extract the high-level topics or themes from a given text, i.e., what kind of topics are discussed in the text.	✗	✓
	Others		
rejecting	Reject to respond when the query is beyond capacity or it violates general ethical and legal rules.	✗	✓
others	You must choose this if none of the other scenarios match the user's query well.	✗	✓

Table 11: Detailed description for each task.

task	train	test	task	train	test	task	train	test
question_generation	30	2	code_language_classification	30	2	roleplay	30	3
story_generation	30	4	code_to_text_translation	30	3	explain_answer	30	4
poem_generation	30	3	instructional_rewriting	30	4	natural_language_learning_tutor	30	2
email_generation	30	3	language_polishing	30	2	exam_problem_solving_tutor	31	2
data_generation	30	3	paraphrasing	30	2	ml_ai_language_model_tutor	30	3
text_to_text_translation	30	3	text_correction	30	2	math_puzzles	30	6
advice_giving	30	4	information_extraction	30	3	fill_in_the_blank	30	3
recommendations	30	2	keywords_extraction	30	2	general_classification	30	4
how_to_generation	30	3	table_extraction	30	3	ordering	30	3
planning	30	2	title_generation	30	2	sentiment_analysis	30	3
code_correction	30	5	text_summarization	30	5	language_classification	30	3
code_simplification	30	2	note_summarization	30	2	topic_classification	30	2
explain_code	30	2	open_qa	30	6	rejecting	30	3
text_to_code_translation	30	4	closed_qa	30	2	others	43	8
code_to_code_translation	30	3	fact_verification	30	2	overall	1395	143
language_learning_questions	31	5	value_judgement	30	2			

Table 12: The task distribution in the training and test set for task classifier.

Classification Prompt
You will receive a user's query. Additionally, you are given some pre-defined tasks below:

[Existing tasks start]
question_generation
story_generation
poem_generation
email_generation
data_generation
advice_giving
recommendations
how_to_generation
planning
instructional_rewriting
language_polishing
paraphrasing
text_correction
code_correction
code_simplification
information_extraction
keywords_extraction
table_extraction
title_generation
text_summarization
note_summarization
explain_code
explain_answer
text_to_text_translation
text_to_code_translation
code_to_code_translation
code_to_text_translation
open_qa
closed_qa
fill_in_the_blank
fact_verification
math_puzzles
language_learning_questions
natural_language_learning_tutor
exam_problem_solving_tutor
ml_ai_language_model_tutor
general_classification
ordering
sentiment_analysis
code_language_classification
language_classification
topic_classification
value_judgement
rejecting
roleplay
default
[Existing tasks end]

Your objective is to choose the most appropriate task that can reflect the high-level intention of this query. You should first clearly give out your choice. Your choice should exactly match one of the task names provided above, without any modification. Do not include the task description in your choice.

Your output should be just the task name.

User's query is below:
[User's query start]
{input}
[User's query end]

Task name:

Table 13: The classification prompt for the task classifier in the training and inference phase.

System Prompt
Please act as a rewriter to modify the format of the AI assistant's response to the user's question presented below.

Please follow the instructions below:

1. Please first determine whether the given format meets the requirements of the user's question, if it does not, then copy the AI assistant's response, if it does, then modify the response's format following the provided format.
2. Your task is limited to altering the format while keeping the original meaning and information intact.
3. Please make sure that the revised response can answer the user's question correctly.
4. Please make sure that the revised response is fluent and has no additional subheadings.

Please first write "Reasoning: <reason>" to provide a brief reasoning you used to modify, and then write "Revised response: <response>" to output your final revised response without any additional information, ensuring its fluency.

Do not output any additional subheadings.

User Prompt
Below is a user's question, the AI assistant's response, and the provided format.

[Question start]
{question}
[Question end]

[Response start]
{response}
[Response end]

[Format start]
{format}
[Format end]

Table 14: Reformatting prompts for non-knowledge-intensive tasks.

System Prompt
Please act as a rewriter to modify the format of the AI assistant's response to the user's question presented below.

Please follow the instructions below:

1. Please first determine whether the given format meets the requirements of the user's question, if it does not, then copy the AI assistant's response, if it does, then modify the response's format following the provided format.
2. Your task is limited to altering the format while keeping the original meaning and information intact.
3. Please make sure that the revised response can answer the user's question correctly.
4. Please make sure that the revised response is fluent and has no additional subheadings.
5. Evidence is the useful information. You should decide for yourself which parts of the evidence to help rewriting the response.

Please first write "Reasoning: <reason>" to provide a brief reasoning you used to modify, and then write "Revised response: <response>" to output your final revised response without any additional information, ensuring its fluency.

Do not output any additional subheadings.

User Prompt
Below is a user's question, the AI assistant's response, the provided format, and the evidences.

[Question start]
{question}
[Question end]

[Response start]
{response}
[Response end]

[Format start]
{format}
[Format end]

[Evidence start]
{evidence}
[Evidence end]

Table 15: Reformatting prompts for knowledge-intensive tasks.

Query How many positive three-digit integers less than 500 have at least two digits that are the same?
Criteria Task Name: math puzzles Format: This is a math question. First, analyze the question and give a brief analysis in the first paragraph. Then, use a list to present the step-by-step solution. Next, give another list to output a detailed explanation. Last, give the correct result and a conclusion.
Original Response To solve this problem, I will count how many three-digit numbers less than 500 have at least two digits that are the same, and subtract that from the total number of three-digit numbers less than 500. The total number of three-digit numbers less than 500 is $4 \times 10 \times 10 = 400$, since the first digit can be 1, 2, 3, or 4, and the second and third digits can be anything from 0 to 9. Now I will count how many of these 400 numbers have no digits that are the same. To have no digits that are the same, the first digit can be any of the 4 choices, but the second digit has to be different from the first, so it has 9 choices. The third digit has to be different from both the first and the second, so it has 8 choices. [* on another note] I am assuming that the second digit can be 0, since the problem does not say otherwise. Therefore, the number of three-digit numbers less than 500 that have no digits that are the same is $4 \times 9 \times 8 = 288$. This means that the number of three-digit numbers less than 500 that have at least two digits that are the same is $400 - 288 = 112$.
REALIGN Response (Ours) Analysis: To solve the problem, we need to count how many three-digit numbers less than 500 have at least two digits that are the same. Solution: 1. Calculate the total number of three-digit numbers less than 500: $4 \times 10 \times 10 = 400$. 2. Count the number of three-digit numbers less than 500 that have no digits that are the same: $4 \times 9 \times 8 = 288$. 3. Subtract the count of numbers with no same digits from the total to find the number of three-digit numbers less than 500 that have at least two digits that are the same: $400 - 288 = 112$. Explanation: - The first digit can be 1, 2, 3, or 4, and the second and third digits can be anything from 0 to 9. - To have no digits that are the same, the first digit can be any of the 4 choices, but the second digit has to be different from the first, so it has 9 choices. The third digit has to be different from both the first and the second, so it has 8 choices. Result: There are 112 three-digit numbers less than 500 that have at least two digits that are the same. Conclusion: There are 112 positive three-digit integers less than 500 that have at least two digits that are the same.

Table 16: A complete example of a query, including the original response and the REALIGN response.

It is an email-writing task. Here is a general guideline for creating a well-structured and professional email: 1. Subject Line: Write a clear and concise subject line that accurately summarizes the content of your email. This helps the recipient understand the purpose of the email at a glance. 2. Salutation: Begin your email with a formal salutation such as "Dear [Recipient's Name]," or use a more casual salutation if you have an informal relationship with the recipient. 3. Introduction: Start your email with a brief introduction, stating who you are and the reason for writing the email. Be clear and to the point, and avoid unnecessary details. 4. Body: This is the main content of your email. Organize your thoughts into paragraphs or bullet points to make them easier to read. Keep your sentences concise and focused. Use proper grammar, punctuation, and spelling to maintain professionalism. If you need to discuss multiple topics, consider using headings or numbered points to separate them. 5. Politeness and Tone: Maintain a polite and respectful tone throughout your email. Be mindful of the recipient's perspective and use appropriate language. Avoid using excessive capitalization, exclamation marks, or emoticons, as they can come across as unprofessional. 6. Closing: Conclude your email with a closing remark, such as "Thank you," or "Best regards," followed by your name. If you expect a response or need specific action, you can mention it in this section as well. 7. Signature: Include your full name, job title, and contact information (e.g., phone number, email address) in your email signature. This helps the recipient easily identify and contact you if needed. 8. Attachments: If you need to include attachments, mention them in the email body and make sure they are relevant to the email's purpose. Consider compressing large files or using cloud storage services if the attachments are too large to be sent via email. 9. Proofread: Before sending the email, proofread it for any grammatical or spelling errors. Make sure the email conveys your message clearly and effectively. The best emails are short, direct, professional, and scannable for the recipient. Follow formal business email structure unless you have an established casual rapport with the recipient.

Table 17: A complete format for "email generation" task.

Query
Is the given statement true or false?
A robotic vacuum can clean both carpets and hard floors.

Criteria
Task Name: fact verification
Format: This is a fact-verification task. First, give the answer. Then, give an explanation.

Retrieved Evidence
1. Robot vacuums run fully autonomously and automatically. They self-navigate on any flooring type, from hardwoods to carpets, and some models can even be programmed to have a cleaning schedule and to sync to smartphones and smart home devices.
2. iRobot roomba vacuum works on laminate flooring. In fact they work better on hard floor as compared to carpet since their suction and beater bar is not ...
3. Our top pick among the best vacuums for hardwood floors is the iRobot Roomba 694. iRobot is known as a top choice when it comes to robotic ...
4. Because they lack the power to deep clean, though, robot vacuums are best for flooring other than carpet, such as tile, laminate or hardwood, ...

Original Response
True.

REALIGN Response (Ours)
True.
Robot vacuums can clean both carpets and hard floors. They are designed to self-navigate on any flooring type, from hardwoods to carpets, and some models can even be programmed to have a cleaning schedule and to sync to smartphones and smart home devices. Additionally, some specific models like the iRobot Roomba 694 are known as top choices for cleaning hardwood floors. However, it is important to note that robot vacuums may lack the power to deep clean carpets compared to hard floors.

Table 18: An example of a knowledge-intensive task, including the query, the criteria, the retrieved evidence, the original response, and the REALIGN response. We highlight the content that is more factual and knowledge in green.

System Prompt
Please act as an impartial judge and evaluate the quality of the responses provided by two AI assistants to the user question displayed below. You should choose the assistant that follows the user's instructions and answers the user's question better. Your evaluation should consider the readability including clarity, format, and ease of understanding. Well-organized, grammatically correct response is better. Begin your evaluation by comparing the two responses and provide a short explanation. Avoid any position biases and ensure that the order in which the responses were presented does not influence your decision. Do not allow the length of the responses to influence your evaluation. Do not favor certain names of the assistants. Be as objective as possible. After providing your explanation, output your final verdict by strictly following this format: "[[A]]" if assistant A is better, "[[B]]" if assistant B is better, and "[[C]]" for a tie.

User Prompt
[User Question]
{question}

[The Start of Assistant A's Answer]
{answer_a}
[The End of Assistant A's Answer]

[The Start of Assistant B's Answer]
{answer_b}
[The End of Assistant B's Answer]

Table 19: Evaluation prompt for readability.

System Prompt
Please act as an impartial judge and evaluate the factuality of the response provided by an AI assistant to the user question displayed below. Your evaluation should consider correctness. You will be given a reference answer and the assistant's answer. Begin your evaluation by comparing the assistant's answer with the reference answer. Identify and correct any mistakes. Be as objective as possible. After providing your explanation, you must rate the response on a scale of 1 to 10 by strictly following this format: "[[rating]]", for example: "Rating: [[5]]".

User Prompt
[Question]
{question}

[The Start of Reference Answer]
{ref_answer}
[The End of Reference Answer]

[The Start of Assistant's Answer]
{answer}
[The End of Assistant's Answer]

Table 20: Evaluation prompt for factuality.

System Prompt
Please act as a mathematics explanation generator to generate a step-by-step explanation of the answer based on the question presented below.

Please follow the instructions below:
1. Please simply generate a step-by-step explanation, including the reason for each step of the calculation.
2. Please do not change the essence of the answer.

Please write "The explanation: <answer>" to output your explanation without any additional information.

Here is an example for your reference:

Question: Natalia sold clips to 48 of her friends in April, and then she sold half as many clips in May. How many clips did Natalia sell altogether in April and May?

Answer: Natalia sold 48/2 = «48/2=24»24 clips in May.\nNatalia sold 48+24 = «48+24=72»72 clips altogether in April and May.\n#### 72

Explanation:
1. The problem states that Natalia sold clips to 48 friends in April. In May, she sold half as many clips as she did in April. We are asked to find out the total number of clips she sold over these two months.
2. To find out how many clips Natalia sold in May, we take half of the number of clips sold in April. Since she sold clips to 48 friends in April, we calculate half of 48: \n$$\frac{48}{2} = 24$$\nThis step involves simple division, where we divide the number of clips sold in April by 2 to find the number of clips sold in May.
3. Now that we have the number of clips sold in May, we need to add this to the number of clips sold in April to get the total number of clips sold over both months:\n$$48 + 24 = 72$$\nThis step involves addition, where we sum the clips sold in April and May to find the total sales for the two months combined.
4. Natalia sold 72 clips in total during April and May. This final total gives us the answer to the question asked.

User Prompt
Below is a question and the answer:

[Question start]
{question}
[Question end]

[Answer start]
{answer}
[Answer end]

Table 21: Prompts for generating explanations for GSM8K.

System Prompt
Please act as a mathematics explanation generator to generate a step-by-step explanation of the answer based on the question presented below.

Please follow the instructions below:
1. Please simply generate a step-by-step explanation, including the reason for each step of the calculation.
2. Please do not change the essence of the answer.

Please write "The explanation: <answer>" to output your explanation without any additional information.

Here is an example for your reference:

Question: Each row of a seating arrangement seats 7 or 8 people. Forty-six people are to be seated. How many rows seat exactly 8 people if every seat is occupied?

Answer: Let x be the number of rows with 8 people. If we removed a person from each of these rows, then every row would contain 7 people. Therefore, $46 - x$ must be divisible by 7.\n\nThen $x \equiv 46 \equiv 4 \pmod{7}$. The first few positive integers that satisfy this congruence are 4, 11, 18, and so on. However, each row contains at least 7 people. If there were 7 or more rows, then there would be at least $7 \cdot 7 = 49$ people. We only have 46 people, so there must be at most six rows. Therefore, the number of rows with 8 people is $\boxed{4}$.\n#### 4

Explanation:
We have a seating arrangement where each row seats either 7 or 8 people. A total of 46 people need to be seated, and all seats must be filled. The question asks how many rows seat exactly 8 people.
1. We let x represent the number of rows that seat exactly 8 people.
2. To simplify the problem, imagine removing one person from each row that currently seats 8 people. This would convert every 8-person row into a 7-person row. Now, all rows (both the original 7-person rows and the adjusted 8-person rows) would seat 7 people.
3. With this adjustment, the total number of people would be reduced by x (since we removed one person from each 8-person row), making it $46 - x$.
4. Since all rows now hypothetically seat 7 people, the adjusted total, $46 - x$, must be divisible by 7 for it to be a possible total number of people seated in rows of 7.
5. We analyze $46 - x$ in terms of modulo 7. Specifically, we want to find values of x such that $46 - x$ is a multiple of 7. This simplifies to finding x such that $x \equiv 46 \pmod{7}$. Calculating $46 \mod 7$ yields 4, because when 46 is divided by 7, the remainder is 4. This tells us x must be equivalent to 4 modulo 7.
6. The numbers that satisfy $x \equiv 4 \pmod{7}$ are 4, 11, 18, etc. However, we need a practical value of x that fits the total people and the row seating constraint.
7. If there were 7 or more rows of 7 people each, we'd have at least 49 people (since $7 \times 7 = 49$). But we only have 46 people, so there must be fewer than 7 rows in total. The plausible values of x from the list 4, 11, 18, etc., must be reconsidered within this context.
8. Since the only value from our possible x values (4, 11, 18) that is less than 7 and fits the total people count is 4, we conclude that there are 4 rows of 8 people each.

User Prompt
Below is a question and the answer:

[Question start]
{question}
[Question end]

[Answer start]
{answer}
[Answer end]

Table 22: Prompts for generating explanations for MATH.