REInstruct: Building Instruction Data from Unlabeled Corpus

Shu Chen[1,3], **Xinyan Guan**[1,3], **Yaojie Lu**[1,*], **Hongyu Lin**[1], **Xianpei Han**[1,2,4,*], **Le Sun**[1,2,4]

[1]Chinese Information Processing Laboratory   [2]State Key Laboratory of Computer Science
Institute of Software, Chinese Academy of Sciences
[3]University of Chinese Academy of Sciences
[4]Key Laboratory of System Software, Chinese Academy of Sciences
{chenshu2020,guanxinyan2022,luyaojie,hongyu,xianpei,sunle}@iscas.ac.cn

Abstract

Manually annotating instruction data for large language models is difficult, costly, and hard to scale. Meanwhile, current automatic annotation methods typically rely on distilling synthetic data from proprietary LLMs, which not only limits the upper bound of the quality of the instruction data but also raises potential copyright issues. In this paper, we propose REInstruct, a simple and scalable method to automatically build instruction data from an unlabeled corpus without heavy reliance on proprietary LLMs and human annotation. Specifically, REInstruct first selects a subset of unlabeled texts that potentially contain well-structured helpful and insightful content and then generates instructions for these texts. To generate accurate and relevant responses for effective and robust training, REInstruct further proposes a rewriting-based approach to improve the quality of the generated instruction data. By training Llama-7b on a combination of 3k seed data and 32k synthetic data from REInstruct, fine-tuned model achieves a 65.41% win rate on AlpacaEval leaderboard against text-davinci-003, outperforming other open-source, non-distilled instruction data construction methods. The code is publicly available at `https://github.com/cs32963/REInstruct`.

1 Introduction

Instruction Tuning (Ouyang et al., 2022) is an important step in aligning large language models (LLMs) (Brown et al., 2020; Touvron et al., 2023a). By tuning LLMs on <instruction, response> pairs, instruction tuning enhances zero-shot performance of LLMs across a variety of tasks (Wei et al., 2022; Chung et al., 2022), and paves the way for further preference training such as reinforcement learning from human feedback (RLHF) (Ouyang et al., 2022; Touvron et al., 2023b) and direct preference optimization (DPO) (Rafailov et al., 2023).
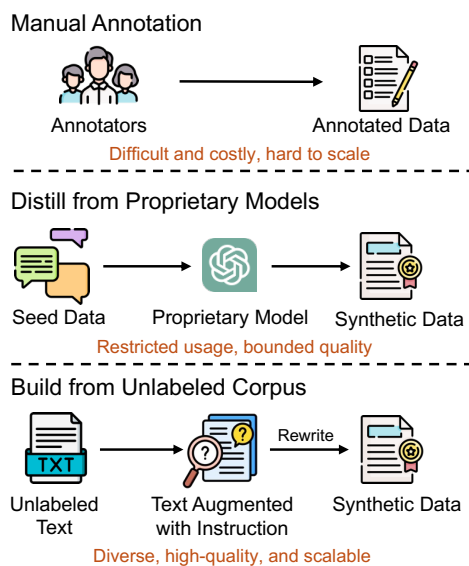

Figure 1: Comparison of our automatic instruction data annotation method and previous methods.

However, it is very difficult and costly to manually annotate high-quality instruction data (Köpf et al., 2023). Previous methods for automatic annotation largely rely on powerful proprietary LLMs like ChatGPT (OpenAI, 2023a), using techniques such as Self-Instruct (Wang et al., 2023) and Evol-Instruct (Xu et al., 2023a). While these methods can generate large amount of instruction data and facilitate research in instruction tuning (Liu et al., 2023a; Chen et al., 2023), the upper bound of the quality of generated data is limited by the power of proprietary LLMs.

In this paper, we propose a simple and scalable method to automatically build instruction data from unlabeled text corpus. The proposed method requires only small amount of human-labeled data, and does not rely heavily on powerful proprietary LLMs. Our key motivation is that unlabeled text corpus contains a subset which is filled with high-quality instruction following information, and converting these texts into proper instruction follow-

---

* Corresponding authors.

ing format is much easier than labeling them from scratch. Specifically, we first employ a set of heuristic rules to obtain unlabeled text containing well-structured helpful and insightful content. Given these content, we then generate related instruction to these high-quality content using an LLM reversely trained on a small set of high-quality human-annotated seed instruction data. Finally, given the generated instruction and the content, we employ a response rewriting process on the content, in order to obtain the high-quality and accurate response of the instruction from the content. To this end, we leverage a rewriting process in the form of generative machine reading comprehension, which is conducted by an addtional LLM trained on a small set of rewriting data.

To demonstrate the effectiveness of our automatic labeling method, we train LLMs jointly on our synthetic data and seed data across different data scales. We find that, Llama-7b (Touvron et al., 2023a) model finetuned with our largest 32k synthetic data and 3k seed data achieves 65.41% win rate on AlpacaEval leaderboard, outperforming other open-source, non-distilled methods with the same pre-trained checkpoint. This demonstrate the effectiveness of REInstruct.

Generally, the main contributions of this paper are:

- We propose a method to automatically build high-quality instruction data from unlabeled corpus, which requires only a small set of seed instruction data and a small set of rewriting data.

- We identify several heuristic rules to obtain high-quality unlabeled text for constructing instruction-following data, and propose a generative machine reading comprehension-based rewriting process to obtain high-quality response for each instruction.

2 Related Work

Instruction Tuning. Early works on instruction tuning focus on traditional NLP tasks, and instruction tuned LLMs show improved zero-shot performance across a variety of NLP tasks (Wei et al., 2022; Chung et al., 2022; Mishra et al., 2022; Sanh et al., 2022). Since the release of ChatGPT (OpenAI, 2023a), there have been increasing interests in general purpose instruction tuning, which expands instructions beyond NLP tasks (Ouyang

et al., 2022; Wang et al., 2023; Taori et al., 2023; Chiang et al., 2023). General purpose instruction tuning serves as the first step to build powerful LLM-based AI assistant (Ouyang et al., 2022; OpenAI, 2023a,b), providing foundation for further alignment training such as RLHF (Ouyang et al., 2022) and DPO (Rafailov et al., 2023).

Automatic Instruction Data labeling. Manually annotating instruction data is very difficult and expensive, making it hard to scale (Köpf et al., 2023). Early instruction datasets are obtained by converting existing task-specific NLP datasets into instruction format (Wei et al., 2022; Chung et al., 2022; Sanh et al., 2022). Recent works have largely rely on powerful proprietary LLMs (Ouyang et al., 2022; OpenAI, 2023a) to generate high-quality instruction data. Alpaca (Taori et al., 2023) use Self-Instruct (Wang et al., 2023) to collect instruction data from InstructGPT (Ouyang et al., 2022) using its in-context learning ability (Brown et al., 2020). WizardLM (Xu et al., 2023a) devise Evol-Instruct to generate more complex and difficult instructions by prompting ChatGPT (OpenAI, 2023a). While these works (Ding et al., 2023; Chiang et al., 2023; Wu et al., 2024) can generate large amount of instruction data and help facilitate research in instruction tuning (Liu et al., 2023a; Chen et al., 2023), they rely on the underlying proprietary LLMs to ensure the diversity, difficulty and quality of instruction data, and fail to address the difficulty of annotating instruction data for these powerful proprietary LLMs.

There have been some effort in automatic instruction dataset annotation without reliance on powerful proprietary LLMs, trying to build instruction data using unlabeled text (Gu et al., 2022). Closest to our work is Humpback (Li et al., 2023a), which proposes to add instructions to unlabeled text and employ an iterative self-rating and selection pipeline to curate high-quality instruction data. However, Humpback relies on the existence of high quality unlabeled text already in the form of instruction response, and leaves the noise in unlabeled text and during instruction generation unaddressed, whereas we tackle the noise directly, resulting in a more efficient and robust pipeline.

Simulated Human Supervision. Since human supervision is difficult and expensive to obtain, recent works have explored the possibility of training LLMs to generate pseudo human supervisions, which usually helps reduce the annotation cost and speed up the alignment process. RLHF (Ouyang
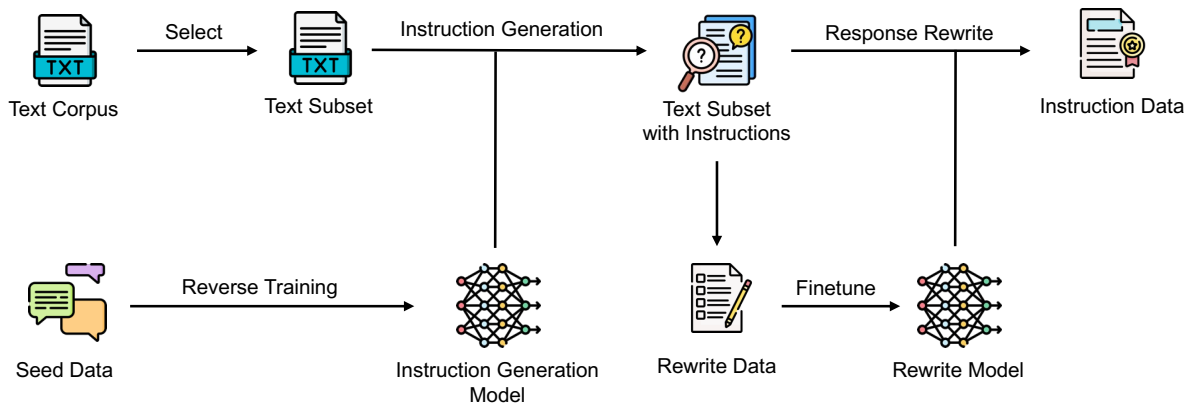
Figure 2: The overall framework of our instruction data annotation method.

et al., 2022; Touvron et al., 2023b) train LLMs on human preference data to provide reward signals for reinforcement learning to align LLMs. CoachLM (Liu et al., 2023b) use expert-annotated instruction revision data to train an LLM for automatic instruction data optimization. Our work is related as we train LLMs to generate instructions for unlabeled text and rewrite the response, which is essentially simulating how humans would annotate instruction data given candidate unlabeled text.

3 Method

The goal of our method is to automatically build instruction data from an unlabeled text corpus. In this section, we first outline how we select unlabeled texts that contain well-structured helpful and insightful contents (§3.1). Next, we describe how to convert these contents into <instruction, response> pairs through instruction generation (§3.2) and response rewriting (§3.3). Finally, we explain how to instruction finetune pretrained LLMs using both seed instruction data and our synthetic data (§3.4). Figure 2 illustrates the overall framework of our method.

3.1 Unlabeled Text Selection

The first step of our method is to select a subset of unlabeled text rich in high-quality instruction following information to steer the direction of model generation. Therefore, we aim to find a subset of text that contains well-structured helpful and insightful content to serve as the source of instruction tuning signals of our annotated data.

While there might be more sophisticated ways to select the desired text, we simply use a set of heuristic rules to filter out low quality text. Using

rule-based filtering has its merits compared with advanced neural methods, as it is fast, stable, and interpretable. The exact sampling rules can be found in Appendix A.1. Some of the major factors and rules we consider include:

1. **Text length**, which is generally related to the quality of the text. Serious and complicated topics often require longer text to elucidate, and longer text usually contains more details. Thus we filter out texts that are too short.

2. **Text structure**, which usually indicates the overall quality of the text and the expertise of the author. Well organized content shows the seriousness of writing and professionalism of the author. We keep text that contains multiple paragraphs starting with verbs in imperative form or present participles.

3. **Text style**, as formal writing usually results in high-quality content. We filter out text with too many first-person pronouns such as "I", "my", "we", and "us", as these pronouns are more common in casual writing.

4. **Promotional text indicators**, since text corpus includes lots of advertisement, we filter out text containing punctuation commonly found in advertisement. We also filter out text with too many all-capitalized words, as they are also common in promotional text.

The resulting subset is a diverse set of high-quality unlabeled text, more suitable for converting into instruction response than randomly sampled text. We find this subset mainly consists of helpful guidance for a particular task, provide solutions for certain problem, or elucidate some complicated

Unlabeled Text:

Traveling is often a joyous experience, where the only moans and groans come from not being able to see everything you wanted to...

These safety tips will help you travel safely on your final trips of the summer.

Dress moderately. Now is not the time to look like you have money, even if that is the case. Dress moderately, and hide anything flashy...

Travel light. For one, it's easier to keep with you at all times—deterring thieves by making it less likely that you'll need to set it down...

...

When you practice these simple safety tips, you drastically improve the safety of your travels. Remember these tips every time you travel and always be on the lookout for new safety tips. Your family and identity are important, so guard them carefully.

Generated Instruction:

Give me some safety tips for my final trips of the summer.

Rewritten Response:

Here are some safety tips for your final trips of the summer:
1. Dress moderately: Avoid drawing attention to yourself by dressing like everyone else. Avoid excessive jewelry or flashy clothing styles that may attract unwanted attention.
2. Travel light: Carrying less weight will make it easier to keep track of your belongings and reduce the chances of becoming a target for thieves.

...

Remember, these safety tips are meant to enhance your overall safety and protect your family and identity. Stay alert and continue to explore new safety tips to ensure a safe and enjoyable travel experience.

Table 1: An instance of our selected unlabeled text, along with generated instruction and rewritten response. Some texts are omitted for length issues.

Answer the question based on the web text provided. Your answer must be helpful, detailed, and polite. Hide the fact that your answer is actually based on the web text, i.e. answer directly.

Web Text:
[Selected unlabeled text]

Question:
[Reversely generated instruction]

Figure 3: Prompt for collecting rewritten responses from ChatGPT.

for reversely training $\mathcal{M}_{\text{reverse}}$ is:

$$\mathcal{L}_{\text{reverse}} = \sum_{(x_s, y_s) \in \mathcal{D}_{\text{seed}}} - \log p(x_s | y_s) \quad (1)$$

where x_s represents instruction and y_s represents response in human-annotated seed instruction data.

3.3 Response Rewriting

Using selected unlabeled text as responses for newly generated instructions is suboptimal. The distribution mismatch between unlabeled text and instruction response results in some noise in the instruction-response pairs, mostly in the form of irrelevant information that does not relate to the newly generated instruction. In addition, the unlabeled text is written from a human perspective, and may differ in the style of a helpful AI assistant. To reduce such noise, we employ a response rewriting process to improve the quality of unlabeled text as instruction response.

Our rewriting process is formed similar to generative machine reading comprehension, in which the rewriting model $\mathcal{M}_{\text{rewrite}}$ is given the unlabeled text as the context, the instruction as the question, and is required to generate a response as the answer. To obtain $\mathcal{M}_{\text{rewrite}}$, we train another LLM on a small set of rewriting data $\mathcal{D}_{\text{rewrite}} = \{(x_g, y_u, y_r)\}$, which contains tuples of unlabeled text y_u, the generated instruction x_g, and the rewritten response y_r, i.e. $\mathcal{M}_{\text{rewrite}}$ is trained with the following objective:

$$\mathcal{L}_{\text{rewrite}} = \sum_{(x_g, y_u, y_r) \in \mathcal{D}_{\text{rewrite}}} - \log p(y_r | y_u, x_g) \quad (2)$$

To collect $\mathcal{D}_{\text{rewrite}}$ for training rewriting model, we use ChatGPT as a proxy for human annotators

concepts. Such texts are more likely to have a proper instruction, and help reduce the noise during instruction generation. The natural diversity inherited from unlabeled corpus also resembles our daily needs, which is more suitable for instruction tuning an LLM-based AI assistant. Table 1 shows an instance of our selected unlabeled text. More examples can be found in Appendix D.

3.2 Instruction Generation

To obtain instruction-response pairs, we reversely train an LLM on a small set of human-annotated seed instruction data $\mathcal{D}_{\text{seed}} = \{(x_s, y_s)\}$, and use the trained model $\mathcal{M}_{\text{reverse}}$ to add instructions to the selected unlabeled text (Li et al., 2023a; Köksal et al., 2024). The generated instruction is answerable by the unlabeled text or by part of it. This is similar to the back-translation setting in machine translation (Sennrich et al., 2016). The objective

to rewrite selected unlabeled texts based on generated instructions. We ask ChatGPT to hide the fact that the response is generated based on some given unlabeled text (see prompt in Figure 3). We find ChatGPT returns valid responses grounded in the given text in most cases, demonstrating the quality of our selected text and the validity of generated instructions. However, there are two cases when ChatGPT fails to rewrite the response properly:

1. ChatGPT fails to hide the fact that the response is generated based on the given unlabeled text, and produces responses including text segments such as "Based on the given web text", which is a leakage from our rewrite prompt. These failure cases are filtered out when training the rewriting model.

2. ChatGPT decides that the unlabeled text does not contain enough information to generate a proper response and refuses to answer the instruction. These responses usually start with "I am sorry" or "I apologize". We keep these failure cases when training rewriting model, as they provide additional defense against noise in our unlabeled text selection and instruction generation. This is similar to the inclusion of unanswerable questions in machine reading comprehension (Rajpurkar et al., 2018). After response rewriting, these cases can be filtered out using simple rules (Appendix A.2).

Both the instruction generation model and the rewriting model are trained at a small scale, with data size similar to seed instruction data. They are kept fixed when scaling up the size of generated data without further annotation cost. We collect the generated instruction x_g and the rewritten response y_r to construct our final synthetic data $\mathcal{D}_{synthetic} = \{(x_g, y_r)\}$. Table 1 shows an instance of generated instruction and the rewritten response.

3.4 Joint Fine-tuning

For the final instruction tuning, we train LLMs jointly on the seed instruction data and our synthetic data. To balance two sources of training data, we up-sample seed data according to the size of synthetic data, as it proves useful in back-translation (Edunov et al., 2018). We append different prompts to these two types of instruction data for differentiation, similar to the setting of tagged back-translation (Marie et al., 2020). Specifically,

we use the same prompt from Humpback (Li et al., 2023a), i.e. we append Prompt $\text{Pr}_{seed} = $ "Answer in the style of AI Assistant." after the seed instructions and append Prompt $\text{Pr}_{aug} = $ "Answer with knowledge from web." after our generated instructions. During inference, we prepend Pr_{seed} before Pr_{aug}, as is shown to be beneficial for the final performance (Li et al., 2023a). The objective of joint fine-tuning is:

$$
\begin{aligned}
\mathcal{L}_{\text{finetune}} = & \lambda \sum_{(x_s, y_s) \in \mathcal{D}_{\text{seed}}} - \log p(y_s | \text{Pr}_{seed}, x_s) \\
& + \sum_{(x_g, y_r) \in \mathcal{D}_{\text{synthetic}}} - \log p(y_r | \text{Pr}_{aug}, x_g)
\end{aligned}
\tag{3}
$$

where λ represents upsample ratio of seed data.

4 Experiments

4.1 Setup

Dataset. We use C4 (Raffel et al., 2020) as our unlabeled text corpus, since it is a free open-source dataset commonly used in pre-training LLMs. C4 contains around 400M text segments cleaned from Common Crawl Project[1]. For the seed instruction data used in training instruction generation model and final instruction tuning, we use 3k high-quality (rank 0) instruction data obtained from the first round of conversations in OpenAssistant (Köpf et al., 2023). We collect 4k rewriting data from gpt-3.5-turbo for training the rewriting model.

Baselines. We mainly compare our method with other instruction tuned models that do not rely on powerful proprietary models:

- OASST$_{3K}$ (Köpf et al., 2023): Llama model tuned with our 3k seed instruction data.

- Guanaco (Dettmers et al., 2023): Llama model tuned with 9k instruction data from all turns in OpenAssistant dataset.

- Dromedary$_{ours}$ (Sun et al., 2023): Llama model trained with 40k randomly sampled instruction data from the original Dromedary dataset.

- Humpback$_{ours}$ (Li et al., 2023a): We re-implemented Humpback using C4 and Llama-7b under our settings. More details and discussions can be found in Appendix B.

	Model	#Annotated Data	#Fine-tuning Data	AlpacaEval(%)
Proprietary	GPT-3.5-Turbo-0301	-	-	89.37
	GPT-4	-	-	95.28
Distilled	Baize-v2-7B	100k	100k	63.85
	Vicuna-7B-v1.1	70k	70k	64.41
	Vicuna-7B-v1.3	125k	125k	76.84
Non-distilled	Guanaco-7B	9k	9k	46.58
	Dromedary$_{ours}$	195	40k	18.52
	Humpback$_{ours}$	3k	3k+32k	49.19
	OASST-7B$_{3K}$	3k	3k	61.70
Ours	REInstruct-7B$_{8K}$	3k+4k	3k+8k	63.78
	REInstruct-7B$_{16K}$	3k+4k	3k+16k	64.75
	REInstruct-7B$_{32K}$	3k+4k	3k+32k	**65.41**

Table 2: Win rate of instruction tuned LLMs against text-davinci-003 on AlpacaEval leaderboard.

Model	Hellaswag	PIQA	Winogrande	ARC_E	ARC_C
Baize-v2-7B	73.21	79.54	68.98	69.44	44.37
Vicuna-7B-v1.1	74.64	78.62	**70.17**	72.01	43.77
Vicuna-7B-v1.3	73.92	79.22	69.30	71.89	44.45
REInstruct-7B$_{32K}$	**77.04**	**79.76**	69.77	**73.19**	**48.04**

Table 3: Results (Accuracy) of different instruction-tuned models on commonsense reasoning.

We also compare with models trained with instruction data distilled from proprietary LLMs (Chiang et al., 2023; Xu et al., 2023a,b) to show the competitiveness of our method.

Evaluation. We use AlpacaEval (Li et al., 2023b) to evaluate the instruction following capability of our instruction tuned LLMs. AlpacaEval consists of 805 test instructions that covers a variety of tasks. The evaluation metric is the win rate against reference response by text-davinci-003, judged by GPT-4 (OpenAI, 2023b).

Implementation Details. We use Llama-7b (Touvron et al., 2023a) as our pre-trained base LLM for data construction, which is fine-tuned for instruction generation and response rewriting. For training these two models, we use a batch size of 128, a learning rate of 2e-5 and a cosine learning rate schedule for 10 epochs, which gradually reduces the learning rate to zero. The up-sample (Edunov et al., 2018) ratio for seed data during joint-tuning is chosen according to the growing size of newly generated data, keeping the ratio of generated data to seed data roughly 2:1 (see Appendix A.3). All instruction tuning experiments are conducted with the same batch size of 128 and a constant learning rate of 2e-5. We use 4 or 8 A100-80G GPUs for most of our training and carry out full-parameter tuning. During instruction gen-

eration and response rewriting, we use greedy decoding with repetition penalty (Paulus et al., 2018; Klein et al., 2017) of 1.05.

4.2 Main Results

We select unlabeled text of size 8k, 16k, and 32k for our experiments, and train Llama-7b jointly on seed data and our synthetic data across different data scales. Table 2 shows the results of our method and other models on AlpacaEval. We can see that:

1) *REInstruct can achieve better performance than other open-source, non-distilled methods.* Our method, which learns from only a few seed data and automatically built instructions, surpasses most non-distilled methods and achieves competitive performance with some distilled methods, such as Baize-v2-7B and Vicuna-7B-v1.1.

2) *By automatically building high-quality instruction data from an unlabeled corpus, REInstruct can lead to scalable improvement on instruction following.* Experiments with 8k, 16k, and 32k data show continuous performance gains as data volume grows.

To better verify the effectiveness of our model, we further compare its performance with other distilled models on commonsense reasoning. The evaluation is conducted using lm-eval (Gao et al., 2023). As shown in Table 3, our method achieves

Figure 4: Sunburst visualization of verb-noun structure of seed instructions and generated instructions on our selected unlabeled text.

better or competitive performance on five common-sense reasoning datasets.

4.3 Quality Analysis of Auto Generated Data

This section analyzes the quality of automatically generated data, including generated instructions and rewritten responses.

4.3.1 Diversity of Generated Instruction

Since our seed instruction data consists of just over 3k data points, we select a subset of 4k unlabeled text from C4 for instruction generation, maintaining a similar data size. Figure 4 shows the sunburst visualization of verb-noun structure of seed instructions and our generated instructions, as used in Self-Instruct (Wang et al., 2023) to show the diversity of tasks. We found our generated instructions are quite diverse and similar to the seed instructions, covering a wide range of queries that may be sent to an AI assistant for help or guidance.

4.3.2 Similarity of Unlabeled Text and Rewritten Response

To show that our rewriting process generates responses based on the unlabeled text and preserves the instruction following signals in it, we measure text similarity between the unlabeled text and the rewritten response. Specifically, we calculate the ratio of words in rewritten response that can be directly copied from unlabeled text. We found that 77.63% of the words in rewritten response also appear in the unlabeled text.

Figure 5: Sunburst visualization of verb-noun structure of generated instructions on randomly sampled unlabeled text.

4.4 Detailed Analysis

This section analyzes the effects of Unlabeled Text Selection and Response Rewriting. We design four ablated variants of REInstruct:

- "w/o Unlabeled Text Selection" directly samples unlabeled text from the corpus at random, rather than using several heuristic rules to obtain high-quality texts (§4.4.1);

- "w/o Response Rewriting" removes the response rewriting step and directly uses unlabeled text as the response (§4.4.2);

	AlpacaEval(%)
OASST-7B$_{3K}$	61.70
REInstruct-7B$_{8K}$	63.78
w/o Unlabeled Text Selection	61.42
w/o Response Rewriting	49.19
Self-Rewrite	56.63
Self-Training	59.76

Table 4: Win rate on AlpacaEval of REInstruct and its ablated variants.

- "Self-Rewrite" uses the LLM only trained on seed data to rewrite the unlabeled text (§4.4.2);

- "Self-Training" uses the self-generated response instead of the rewritten response from unlabeled text (§4.4.3);

We conduct analysis at the scale of 8k synthetic data, and Table 4 shows the win rate on AlpacaEval of these different variants.

4.4.1 Effect of Unlabeled Text Selection

To understand the effect of our unlabeled text selection, we compare it with random sampling from the text corpus. Figure 5 shows the sunburst visualization of instructions generated on randomly sampled unlabeled text. As seen, most instructions are limited to a few text generation tasks, indicating low instruction diversity.

We then conduct an instruction tuning experiment and evaluate on AlpacaEval, using the same process as the original REInstruct but with the randomly sampled unlabeled text. Table 4 shows that using randomly sampled unlabeled text (w/o Unlabeled Text Selection) leads to no significant performance improvement over the baseline OASST-7B$_{3K}$ (61.70% to 61.42%), while our method REInstruct-7B$_{8K}$ improves performance from 61.70% to 63.78%.

Based on the above observation, it is clear that not all unlabeled texts are worth converting into instruction tuning data. And our unlabeled text selection can effectively obtain high-quality unlabeled text for constructing instruction-following data.

4.4.2 Effect of Response Rewriting

To show the necessity of the rewriting process, we conduct instruction tuning without rewriting unlabeled text, i.e. the selected unlabeled text is di-

rectly used as instruction response. As shown in Table 4, using unlabeled text as responses (w/o Response Rewriting) significantly degrades performance. This observation indicates that the content from unlabeled texts are unsuitable to be directly used as accurate instruction responses for model training.

An interesting question is whether we can skip with the external rewriting data and directly employ the LLM instruction-tuned on the seed data to rewrite the unlabeled text. We refer such approach as Self-Rewrite. Table 4 shows that self-rewrite leads to inferior performance. We suspect that LLMs fine-tuned on a small set of seed data with relatively short instructions might struggle with the rewriting task where the rewriting instructions are much longer. Adding longer instructions or other content-grounded generation data to the seed data might help reduce reliance on external rewriting data. We will explore this in future work.

4.4.3 Effect of Instruction Following Signals in Unlabeled Text

In this section, we investigate the role of instruction following signals extracted in our selected unlabeled text. Previous works (Zhou et al., 2023) have shown that LLMs obtain quite reasonable instruction following capability with limited instruction data. Therefore, it is possible that a self-training approach where model is trained on its own responses on a larger set of instructions may lead to better performance.

To understand the difference between self-training and our method, we use instructions reversely generated on our unlabeled text to prompt LLM tuned on seed data. We then finetune a new LLM jointly on the seed data and the self-generated data. In other words, the sampled unlabeled text is only used to produce a diverse set of instructions, and does not provide instruction following signals in the response.

Table 4 shows that the result of self-training is inferior to our method. Our hypothesis is that good instruction tuning requires continuously providing high quality instruction following signals that can improve current output space of the trained LLM. The instruction data generated in a self-training fashion already holds large amount of probability space, and does not provide much training signals.

5 Conclusion

In this paper, we propose REInstruct, a simple and scalable method to automatically build instruction data from unlabeled corpus. Our method only requires small amount of labeled data, and does not heavily rely on proprietary LLMs. Experiment results show that instruction tuning LLMs on a small set of human-labeled data and our synthetic data achieves competitive instruction following performance, demonstrating the quality of our synthetic data. We believe this work provides insights for instruction data annotation and helps foster research in instruction tuning. For future work, we aim to develop more sophisticated neural methods to collect and extract instruction following signals from unlabeled dataset across different modalities.

Limitations

Using heuristic rules to sample unlabeled text has certain limitations. First, it only captures some shallow textual features that are indicators of high-quality text, with limited semantic understanding of the content. Second, it might be hard to generalize to other data distributions such as code, or other modalities such as vision. Finding high-quality unlabeled data might be more subtle in these settings.

While our method leads to more capable LLMs, further alignment training is required to reduce the hallucination and toxicity of the instruction tuned LLMs.

Acknowledgements

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 *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual.*

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. *CoRR*, abs/2307.08701.

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.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Y. Zhao, Yanping Huang, Andrew M. Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models. *CoRR*, abs/2210.11416.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *CoRR*, abs/2305.14314.

Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. 2023. Enhancing chat language models by scaling high-quality instructional conversations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 3029–3051. Association for Computational Linguistics.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 489–500. Association for Computational Linguistics.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2023. A framework for few-shot language model evaluation.

Yuxian Gu, Pei Ke, Xiaoyan Zhu, and Minlie Huang. 2022. Learning instructions with unlabeled data for zero-shot cross-task generalization. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu*

Dhabi, United Arab Emirates, December 7-11, 2022, pages 1617–1634. Association for Computational Linguistics.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. 2017. Opennmt: Opensource toolkit for neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, System Demonstrations*, pages 67–72. Association for Computational Linguistics.

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. *CoRR*, abs/2304.07327.

Abdullatif Köksal, Timo Schick, Anna Korhonen, and Hinrich Schütze. 2024. Longform: Effective instruction tuning with reverse instructions. *Preprint*, arXiv:2304.08460.

Xian Li, Ping Yu, Chunting Zhou, Timo Schick, Luke Zettlemoyer, Omer Levy, Jason Weston, and Mike Lewis. 2023a. Self-alignment with instruction backtranslation. *CoRR*, abs/2308.06259.

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023b. 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. 2023a. What makes good data for alignment? A comprehensive study of automatic data selection in instruction tuning. *CoRR*, abs/2312.15685.

Yilun Liu, Shimin Tao, Xiaofeng Zhao, Ming Zhu, Wenbing Ma, Junhao Zhu, Chang Su, Yutai Hou, Miao Zhang, Min Zhang, Hongxia Ma, Li Zhang, Hao Yang, and Yanfei Jiang. 2023b. Automatic instruction optimization for open-source LLM instruction tuning. *CoRR*, abs/2311.13246.

Benjamin Marie, Raphael Rubino, and Atsushi Fujita. 2020. Tagged back-translation revisited: Why does it really work? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 5990–5997. Association for Computational Linguistics.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2022. Cross-task generalization via natural language crowdsourcing instructions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 3470–3487. Association for Computational Linguistics.

OpenAI. 2023a. Chatgpt: Optimizing language models for dialogue. https://openai.com/chatgpt.

OpenAI. 2023b. GPT-4 technical report. *CoRR*, abs/2303.08774.

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 *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Arnold Overwijk, Chenyan Xiong, and Jamie Callan. 2022a. Clueweb22: 10 billion web documents with rich information. In *SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022*, pages 3360–3362. ACM.

Arnold Overwijk, Chenyan Xiong, Xiao Liu, Cameron VandenBerg, and Jamie Callan. 2022b. Clueweb22: 10 billion web documents with visual and semantic information. *Preprint*, arXiv:2211.15848.

Romain Paulus, Caiming Xiong, and Richard Socher. 2018. A deep reinforced model for abstractive summarization. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

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. *CoRR*, abs/2305.18290.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 2: Short Papers*, pages 784–789. Association for Computational Linguistics.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal V. 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 Févry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M. Rush. 2022. Multitask prompted training enables zero-shot task generalization. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

Zhiqing Sun, Yikang Shen, Qinhong Zhou, Hongxin Zhang, Zhenfang Chen, David D. Cox, Yiming Yang, and Chuang Gan. 2023. Principle-driven self-alignment of language models from scratch with minimal human supervision. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

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, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971.

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, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

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), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 13484–13508. Association for Computational Linguistics.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022. Finetuned language models are zero-shot learners. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Minghao Wu, Abdul Waheed, Chiyu Zhang, Muhammad Abdul-Mageed, and Alham Fikri Aji. 2024. Lamini-lm: A diverse herd of distilled models from large-scale instructions. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2024 - Volume 1: Long Papers, St. Julian's, Malta, March 17-22, 2024*, pages 944–964. Association for Computational Linguistics.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. 2023a. Wizardlm: Empowering large language models to follow complex instructions. *CoRR*, abs/2304.12244.

Canwen Xu, Daya Guo, Nan Duan, and Julian J. McAuley. 2023b. Baize: An open-source chat model with parameter-efficient tuning on self-chat data. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 6268–6278. Association for Computational Linguistics.

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

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. *CoRR*, abs/2305.11206.

A Implementation Details

A.1 Heuristic Rules to Sample Unlabeled Text

We describe the factors we consider when devising the heuristic rules to select candidate unlabeled texts in section 3.1. Here are the exact rules:

1. Texts shorter than 1200 characters or longer than 3000 characters are filtered out.

2. Texts will be kept if there are 4 to 10 paragraphs starting with verbs in imperative form or present participles, and the number of other paragraphs is less than 2.

3. Texts will be filtered out if they contain more than two words from this list: "we ", "our ", "i ", "i've ", "we've ", "we're ", "my ", "he ", "she ", "us ".

4. Texts with the following punctuation will be filtered out: "...", ". . . ", "™", "#", "&", "*", "®", "@".

5. Texts with more than two all-capitalized words such as "COME" and "NOW" are filtered out.

6. Texts with more than one question mark("?") will be filtered out.

A.2 Rules to Filter Failed Rewritten Data

To filter out responses that fail to hide the fact that the answer is based on the given web text, we remove responses that contain "web text"(leakage from our prompt) and "based on the information provided". To filter out responses that refuse to answer the instruction, we remove responses that contain "sorry" and "i apologize".

A.3 Finetune Details

For instruction tuning on seed data, we use batch size of 128 and train Llama-7b for 200 steps. For joint instruction tuning on seed data and synthetic data, we also use batch size of 128 and finetune models with the number of steps and upsample ratio listed in Table 5.

Model	#Annotated Data	#Fine-tuning Data	Training Steps	Upsample Ratio
REInstruct-7B$_{8K}$	3k+4k	3k+8k	500	1
REInstruct-7B$_{16K}$	3k+4k	3k+16k	800	2
REInstruct-7B$_{32K}$	3k+4k	3k+32k	1000	4

Table 5: Training steps and upsample ratio

A.4 Prompts

Figure 6 shows the system prompt we use for instruction tuning.

A chat between a curious user and an artificial intelligence assistant. The assistant gives helpful, detailed, and polite answers to the user's questions.

User:
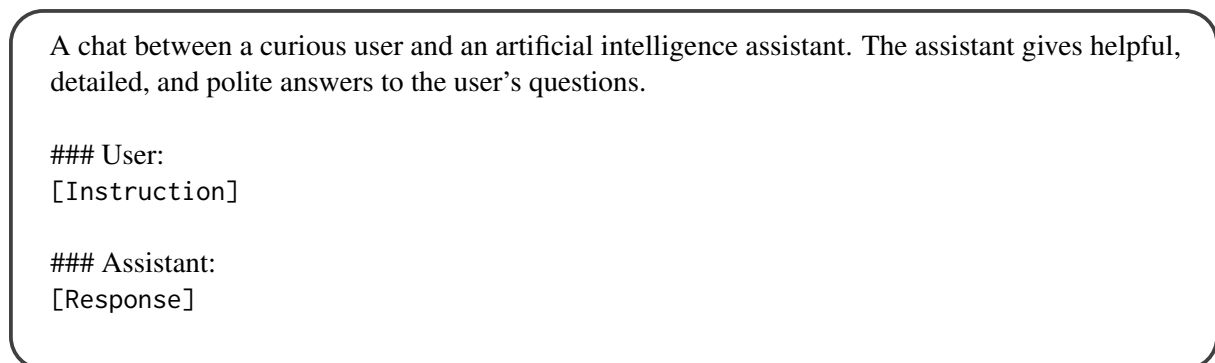[Instruction]

Assistant:
[Response]

Figure 6: System prompt for instruction tuning

B Our implementation of Humpback

We use the exact scoring prompt in Humpback (Li et al., 2023a), and run the iterative curation process on our 32k selected unlabeled text. The base model we use is Llama-7b. We found that during the first iteration, about half of our unlabeled text is ranked 5, and the other half is ranked 4. We use texts with score 5 to train the model in the second iteration, and the resulting model produces score 5 for almost all our unlabeled text, thus the performance of the final trained model will be the same as our ablation experiment "w/o Response Rewriting" in Table 4, which has win rate about 50%, much lower than the performance reported in the Humpback paper. Possible reasons for such performance are:

1. Difference in implementation: Since Humpback is not open-source, some exact details are unclear. For example, it is possible that they use Llama-65b model to curate instruction data and use the obtained data to train Llama-7b model for ablation. We also notice other self-alignment works (Sun et al., 2023; Yuan et al., 2024) typically use larger LLMs of scale 65b or 70b.

2. Difference in unlabeled dataset: Since ClueWeb (Overwijk et al., 2022a) used in Humpback (Li et al., 2023a) is not a free open-source dataset, and we are unable to obtain it in a short time, we use the more accessible C4 dataset for our experiment. The difference between CommonCrawl and ClueWeb is well-discussed in Overwijk et al. (2022b), and ClueWeb generally contains higher quality texts that are more likely to satisfy potential information needs.

To the best of our knowledge, there are no public implementations of Humpback on the 7b model scale. On the other hand, it is completely possible to combine self-alignment works such as Humpback (Li et al., 2023a) with our method. We leave that for future investigation.

C Possible Bias from GPT-4 Evaluation

In the case of AlpacaEval, three potential biases of automatic evaluators such as GPT-4 are:

- Prefer longer responses: It is possible that longer responses contribute to the win rate of our method, but since our average response length is not unreasonably longer than other models (Table 6), we believe our results are still valid in the comparison setting.

- Gives more importance to the style of the response than its content (e.g. factuality): More investigation might be needed to understand how this bias affects our reported results. It is possible that our responses are in a more helpful or friendly tone (e.g. trying to provide more information), but has some issues such as factuality.

- Prefer responses from models that are similar: Since we do not directly distill responses from ChatGPT or GPT-4, this bias has less impact on our reported win rate.

	Model	Average Length	AlpacaEval(%)
Distilled	Baize-v2-7B	1127	63.85
	Vicuna-7B-v1.1	1127	64.41
	Vicuna-7B-v1.3	1100	76.84
Non-distilled	Guanaco-7B	1364	46.58
	OASST-7B$_{3K}$	991	61.70
	REInstruct-7B$_{32K}$	1096	**65.41**

Table 6: Average response length on AlpacaEval test data.

D   Examples

We found almost every unlabeled text we selected contains some useful information or knowledge. However, it is possible that average people like us are not capable to judge whether the text contain the best information or knowledge about the discussed topic. Table 7 provides an example that we think might not be good enough. Meanwhile, some texts that are suitable for converting into instruction data might be filtered out by our heuristic rules. We consider finding such texts a major future direction. One possible solution is to use the selected texts as positive examples and random texts as negative examples to train a neural classifier. Table 8 shows an example that might be good for converting into instruction data.

We include more <unlabeled text, generated instruction, rewritten response> triples in Table 9 and Table 10, so that readers can better understand our synthetic data.

Being An Entrepreneur is not at all Glamorous It's stressful. If you think meeting your boss's deadlines or demands is tough, try meeting your own, especially when your personal savings are on the line, workload can be intense and it's frustrating.

Remember the time when you had the great idea to start your business? You did everything possible to turn that vision into a reality. It was exciting, nerve-wracking and rewarding all at the same time.

Don't fear failure; people we try to avoid the negative consequences of failure; BUT we also lose out on the chance of success.

Create a wall of positivity, a motivational wall start printing out quotes, pictures that motivates you and reminds you of your passion.

Surround yourself with like-minded people. Try to spend time with people who have a positive outlook on life, they can be very inspiring, and avoid negative people.

Take care of yourself, recharge your energy by doing something that makes you happy.

Have other goals to achieve in your life other than work, that will make you happy and gives you strength and confidence.

Remember you have adopted a lifestyle of taking chances, while someone else has declined that challenge, don't lose your entrepreneurial spirit. It's what got you where you are in the first place, and it will help you every day as you live in this non-stop world.

Table 7: An example of selected unlabeled text that might not be suitable for converting into instruction data

When considering the legacy you leave behind, have you thought of how your leadership skills will affect that legacy? Leadership skills are a core component of a long lasting legacy. Here are some tips for improving your leadership skills.

Leadership begins and ends with vision. People enjoy following someone that has vision and is able to communicate that vision in easy-to-understand ways. This may be done in story telling or in personal interactions with your team. Each leader has their own personality and leadership style, so use your strengths to communicate your vision.

It's easy to spot a person that is passionate about something. It comes across in the way they talk about it, the level of detail in which they explain it and through their facial and body expressions. Leaders share their passion with others and others respond by aligning their activities to support that passion driven vision. If you're excited about something, others will be too.

It's important to practice what you preach. If you don't model the morals and integrity you are asking of others, it will undermine your leadership. If you're going to talk the talk, you should walk the walk. You will quickly find that others will begin to emulate your behavior so be sure that you are behaving in a way that you want your team to behave.

You should surround yourself with people that are better than you. Use their strengths to your advantage, challenge them to be even better, and reward them as they excel.

If you expect people to excel, you need to communicate progress along the way. Communicate daily with your inner circle to ensure that their progress is properly aligned with your vision. Be transparent, talk openly about things that are working and things that aren't. Analyze everything to discover things that can be improved. Pivot when things are not working well. Communicate with team members in your outer circle on a monthly basis. Let them know how things are progressing, talk about things that are working and things that aren't. A well informed team will always perform better than those that are held in the dark.

Table 8: An example of filtered out unlabeled text that might be suitable for converting into instruction data

Unlabeled Text	Scratched windowpanes can diminish the beauty of the windows in your home. Learn how to remove these marks with the following tips. Once the glass is damage free and clean, carefully maintain the windowpanes so that they do not become scratched in the future.
	Apply a small amount of soapy water to a damp sponge. Wipe each window pane with the sponge to remove dirty residue. The residue may be covering some of the scratches, so removing it will make it easier for you to spot all of the damage on each windowpane. Rinse the sponge out with plain water and move it over the same spots. Put on a pair of rubber gloves and a face mask to protect your skin from chemicals and prevent breathing in the ammonia.
	Add equal amounts of ammonia and warm water to a spray bottle. Shake the contents. Spray the surface of each windowpane with the cleaning solution. Wipe each windowpane off with a lint-free cloth. The ammonia will eliminate any fine particles that remain on the glass panes. Dry each pane off with another lint-free cloth.
	Pour a few drops of cerium oxide onto a buffing pad. Cerium oxide is a type of polish that is often used to clean jewelry. Place the pad directly over one of the scratches. Press down on the back of the pad as you move it around with small, circular motions. Add a few more drops of cerium oxide as needed. The scratch will fade as you polish it.
	Remove the remaining scratches in the same manner. Spray the window panes, once again, with the ammonia and water solution. Eliminate residue that the cerium oxide left behind with a lint-free cloth. Dry each pane well with another cloth. The windowpanes will be damage-free.
	Protect the glass from becoming scratched in the future by always using soft cloths and nonabrasive cleaning supplies. Remove dirty residue from the glass as soon as it becomes apparent to prevent it from drying on the glass and damaging its surface. When you open or close a window in your home, grasp the frame that surrounds the glass rather than pushing up or down on the pane. These simple steps will protect the windowpanes so that they continue to look great for years to come.
	If you have cracks or holes in your windows, contact a company like Action Glass to repair them for you.
Generated Instruction	How do I remove scratches from windowpanes?
Rewritten Response	To remove scratches from windowpanes, you can follow these steps:
	1. Start by applying a small amount of soapy water to a damp sponge. Wipe each window pane with the sponge to remove dirty residue. This will help reveal any scratches that may be hidden underneath.
	2. Rinse the sponge out with plain water and repeat the process on the same spots. It's important to wear rubber gloves and a face mask to protect yourself from chemicals and prevent breathing in ammonia.
	3. Create a cleaning solution by mixing equal amounts of ammonia and warm water in a spray bottle. Shake the contents and spray the surface of each windowpane with the cleaning solution. Use a lint-free cloth to wipe each windowpane off.
	4. To further remove any remaining particles, add a few drops of cerium oxide onto a buffing pad. Cerium oxide is a type of polish commonly used to clean jewelry. Place the pad directly over a scratch and press down on it while moving in small, circular motions. Repeat this step if needed and add more drops of cerium oxide as required.
	5. Finally, remove any remaining residue left by the cerium oxide with a lint-free cloth. Make sure to dry each pane thoroughly with another lint-free cloth.
	By following these steps, you should be able to effectively remove scratches from your windowpanes and restore their beauty. Additionally, it's important to protect the glass from becoming scratched in the future by using soft cloths and nonabrasive cleaning supplies. Always remove dirty residue as soon as possible to prevent it from damaging the surface of the glass. When opening or closing a window, grasp the frame instead of pushing on the pane, as this can help protect the windowpanes.

Table 9: An example of selected unlabeled text, along with generated instruction and rewritten response

Unlabeled Text	If you are going under the knife then you will need to prepare yourself for any cosmetic surgery procedure in Oregon. Preparing your home, your life, and yourself for plastic surgery will help to make the entire process less stressful. Here are some tips that you should consider.

Talk to your doctor about recommendations for the specific procedure that you have chosen. For example, anyone who has chosen to have a tummy tuck may need to lose a certain amount of weight before surgery. Those who are having breast implants may need to stop taking hormonal drugs. Depending upon the specific surgery you have chosen you may need to prepare in certain ways.

Stop drinking alcohol. Talk with your doctor regarding his policy on smoking cigarettes. By doing this you will clear your system of nicotine and this will help to facilitate the healing process. It is also important that you avoid taking aspirin and any other types of restricted drugs at least two weeks before you have surgery.

Prepare your home properly. You may need to rest up before while afterwards and therefore you need to prepare yourself for this. Make sure you fill prescriptions in advance, make sure that you have clean linens, find somebody to walk your dog and take care of your pets, and so on.

Stock up your fridge as well. When you get back from the hospital the last thing that you really want to have to worry about is going out and getting food. Instead, that all necessary food up in advance and get loads of healthy snack foods that you can use to help you rehabilitate.

Plan exactly how you intend to get to the hospital and back again once the surgery has been done. While you could use a taxi, it would be better to get a friend or family member to make sure you get there and back again safely. It is better to have someone drive you back home and stay with you the night of surgery.

Next Post Next Can Phoenix Dentists Show Good Habits To Promote Healthy Teeth? |
| **Generated Instruction** | What preparation should I do if I am going under the knife in Oregon? |
| **Rewritten Response** | If you are going under the knife in Oregon, there are several preparations you should consider to make the process less stressful. Firstly, it is important to talk to your doctor about any recommendations they may have for your specific procedure. This could include weight loss, stopping certain medications, or any other precautions that may be necessary.

Additionally, it is crucial to stop drinking alcohol and discuss your doctor's policy on smoking cigarettes. This will help clear your system of nicotine and promote a faster healing process. It is also advisable to refrain from taking aspirin and other restricted drugs at least two weeks before the surgery.

Preparing your home is another important aspect of the process. You may need to rest and recuperate both before and after the surgery, so it is essential to have everything in order. This includes filling prescriptions in advance, ensuring you have clean linens, arranging for someone to take care of your pets, and planning for any necessary household tasks.

Furthermore, it is recommended to stock up your fridge with necessary food items before the surgery. Having easy-to-prepare meals and healthy snacks available will save you the hassle of having to leave the house immediately after the procedure.

Lastly, it is crucial to plan your transportation to and from the hospital. While you could rely on public transportation or a taxi, it is safer to have a friend or family member drive you there and back. It is also advisable to have someone stay with you overnight after the surgery for support and assistance.

By following these preparation steps, you can ensure a smoother experience during your cosmetic surgery in Oregon. |

Table 10: An example of selected unlabeled text, along with generated instruction and rewritten response