Instruction-Tuning Data Synthesis from Scratch via Web Reconstruction

Anonymous ACL submission

Abstract

The improvement of LLMs' instruction-following capabilities depends critically on the availability of high-quality instruction-response pairs. While existing automatic data synthetic methods alleviate the burden of manual curation, they often rely heavily on either the quality of seed data or strong assumptions about the structure and content of web documents. To tackle these challenges, we propose **Web Reconstruction** (WebR), a fully automated framework for synthesizing high-quality instruction-tuning (IT) data directly from raw web documents with minimal assumptions. Leveraging the inherent diversity of raw web content, we conceptualize *web reconstruction* as an instruction-tuning data synthesis task via a novel dual-perspective paradigm—*Web as Instruction* and *Web as Response*—where each web document is designated as either the input or output role to trigger the reconstruction process. Comprehensive experiments show that datasets generated by WebR outperform state-of-the-art baselines by up to 16.65% across four instruction-following benchmarks. Notably, WebR demonstrates superior scalability, data efficiency, and compatibility with existing datasets, enabling enhanced domain adaptation with minimal effort. The data and code will be publicly available.

1 Introduction

Large language models (LLMs) (Brown et al., 2020; Achiam et al., 2023; Dubey et al., 2024) have become integral across a myriad of applications, demonstrating exceptional performance on diverse tasks by effectively following instructions (OpenAI, 2022; Achiam et al., 2023). Their remarkable performance largely stems from supervised fine-tuning (SFT) (Wei et al., 2022; Mishra et al., 2022) on instruction-response pairs. This process empowers LLMs to produce customized outputs when provided with specific instructions, facilitating their adaptation to novel tasks without prior exposure.

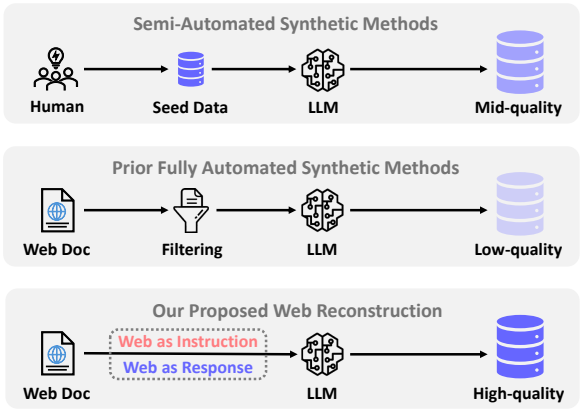

Figure 1: Our proposed Web Reconstruction method surpasses previous techniques by being (1) fully automated, eliminating the need for manual intervention or seed data; (2) minimally reliant on assumptions about the structure and content of web documents; and (3) capable of generating high-quality IT data.

A fundamental challenge in advancing the instruction-following capabilities of LLMs lies in the collection of high-quality instruction-tuning (IT) data. Early approaches primarily rely on human experts to manually generate and curate IT data (Wang et al., 2022; Conover et al., 2023), which is both time-intensive and resource-heavy. To mitigate these limitations, **Semi-Automated Synthetic Methods** (Wang et al., 2023; Taori et al., 2023; Xu et al., 2024a) leverage LLMs to expand small, human-annotated seed datasets using few-shot prompting techniques. While effective, the performance of these methods is highly sensitive to prompt engineering and the careful selection of seed examples (Xu et al., 2024b). More recently, **Fully Automated Synthetic Methods**, such as WebInstruct (Yue et al., 2024) and instruction back-translation (Li et al., 2024d), have emerged as scalable alternatives that eliminate human involvement by synthesizing IT data based on web-scraped documents. These methods, however, often operate under strong assumptions about the structure and

content of raw web data, such as the availability of explicit question-answer pairs or minimal irrelevant content. Consequently, they can only handle a limited scope of web documents, resulting in potential biases that hinder the quality and generalizability of the generated IT data.

As illustrated in Figure 1, we propose **Web Reconstruction (WebR)**, a fully automated framework that transcends these limitations by rethinking how web data can be transformed into high-quality IT datasets. To overcome the reliance on strong assumptions of web data, we conceptualize *web reconstruction* as an instruction-tuning data synthesis task, where raw noisy web documents are reconstructed into human-preferred, response-like outputs. This shift enables WebR to **thoroughly exploit the inherent diversity of raw web documents**. Central to WebR is a novel dual-perspective paradigm—*Web as Instruction* and *Web as Response*—which designates each web document as either the input or output role to trigger the reconstruction process. (1) *Web as Instruction* pairs the raw web with a synthesized rewrite request to serve as a complete instruction, enabling the generation of reorganized and coherent responses. Web rewriting inherently encompasses diverse NLP tasks such as reading comprehension and information extraction, thus compelling LLMs to engage in reasoning and execute intricate instructions during training. (2) *Web as Response*, drawing inspiration from instruction backtranslation (Li et al., 2024d), first produces a latent instruction by treating the raw web content as the response. Subsequently, a contextually appropriate web is reconstructed via two stages: LLM's initial rollout and further refinement. This strategy significantly boosts the LLM's ability to acquire knowledge and perform complex question-answering tasks.

We apply WebR to the Llama3-70B-Instruct and GPT-4o-mini models, creating two 100k-sample IT datasets: WebR-Basic and WebR-Pro. To validate their effectiveness, we train various LLMs, including Llama3-8B-base and Qwen2.5-1.5/3/7/14B-base, and evaluate them on four widely used benchmarks. WebR-Basic consistently outperformed public IT dataset baselines on all benchmarks, achieving a remarkable **16.65%** average improvement. Moreover, WebR-Pro demonstrated superior performance compared to a mixture of prior IT datasets generated using GPT-4o-mini, even with the same data quantity. Interestingly, merging WebR with existing IT datasets yielded further performance gains, highlighting its compatibility and effectiveness. Detailed analyses reveal three key insights: (1) the effectiveness of WebR scales with the size of the base LLM; (2) WebR exhibits superior data efficiency, with performance improving linearly relative to the logarithmic growth of training data; and (3) WebR facilitates effective domain adaptation through the simple inclusion of domain-specific web documents.

2 Related Work

The synthesis of high-quality instruction-tuning (IT) data (Zhou et al., 2023a; Jiang et al., 2023; Xu et al., 2024a) is crucial for improving LLMs' instruction-following capabilities. Previous studies can be broadly classified into three main categories.

Human-Crafted Method primarily involves employing professionals to create instructions, as seen in datasets like SUPER-NI (Wang et al., 2022) and DOLLY (Conover et al., 2023). While these datasets offer high-quality content, their size is constrained by the significant costs associated with manual creation. Alternatively, approaches like ShareGPT (Chiang et al., 2023) and WildChat (Zhao et al., 2024) leverage user interaction logs with LLMs to collect human-generated instructions. However, this method risks incorporating toxic or undesirable content (Zhao et al., 2024).

Semi-Automated Synthetic Method uses LLMs to generate synthetic IT datasets by starting with a small set of human-annotated seed data and expanding them through few-shot prompting. Notable methods include Self-Instruct (Wang et al., 2023), Alpaca (Taori et al., 2023), and Evol-Instruct (Xu et al., 2024a). While these techniques enable large-scale data generation, the diversity of the synthesized data is often constrained by the quality and variety of seed examples (Li et al., 2024a).

Fully Automated Synthetic Method utilizes LLMs to synthesize IT data from scratch, drawing from web-scraped documents. For instance, WebInstruct (Yue et al., 2024) extracts question-answer (QA) pairs from web documents to construct instruction-response datasets. Nevertheless, this approach depends on the explicit presence of QA pairs within the raw web corpus, which is not always guaranteed. Similarly, backtranslation (Li et al., 2024d; Nguyen et al., 2024) treats web documents as natural responses and employs LLMs to infer the corresponding latent user instructions.

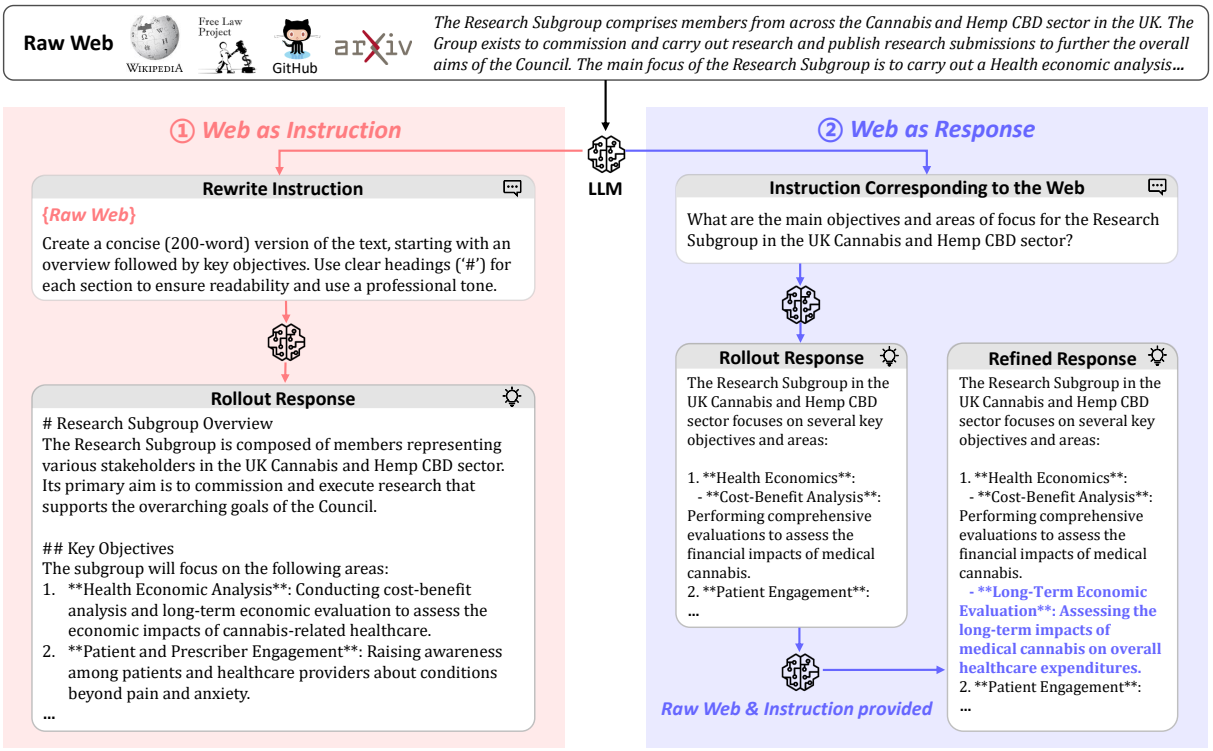

Figure 2: Overview of the proposed **Web Reconstruction** (WebR) framework. Leveraging an off-the-shelf LLM, WebR transforms raw web documents into high-quality instruction-response pairs. It strategically assigns each document as either an instruction or a response to trigger the process of web reconstruction.

However, web documents often contain irrelevant content or unsuitable expressions, making them suboptimal as response candidates.

3 Web Reconstruction

Prior fully automated synthetic methods often rely on strong assumptions about the structure and content of raw web documents—such as the presence of explicit question-answer pairs, minimal irrelevant content, or appropriate expressions—necessitating complex preprocessing steps like retrieval and filtering. In contrast, we introduce the **Web Reconstruction** (WebR) framework, which leverages a powerful, off-the-shelf LLM to overcome these limitations by directly reconstructing unstructured and noisy web content into high-quality, response-like outputs. As shown in Figure 2, WebR comprises two core strategies: (1) *Web as Instruction*, where raw web content is concatenated with a synthesized rewrite request to serve as a complete instruction, guiding the generation of a reorganized, coherent response; (2) *Web as Response*, where a latent instruction is inferred by treating raw web content as a response, enabling reconstruction through the LLM's initial rollout and subsequent refinement. By adopting this dual-branch approach, WebR efficiently generates high-quality instruction-response pairs, ensuring contextually appropriate outputs while eliminating the need for extensive preprocessing.

3.1 Web as Instruction

Raw web documents often contain disorganized or irrelevant information that hinders direct usability. Even when dealing with well-structured content, further refinement is often required to meet human-preferred formats and stylistic conventions. A natural approach to reconstructing web content is to rewrite it according to specific requirements, such as style, format, structure, etc. To ensure diverse and realistic rewriting scenarios, we leverage a powerful LLM to generate a detailed rewrite request tailored to the original document's content (See prompt in Figure 8). The request, along with the raw web content, are concatenated to form a comprehensive instruction. In addition to whole-document transformations, we further enhance task diversity by randomly (50% probability) generating rewrite requests that target *specific sections* of the web content rather than the entire document, as shown in Figure 9. This simulates real-world text manipulation scenarios where users may need to

extract and modify only certain portions of a text. The curated instructions are then processed by the LLM to produce reconstructed web content. Notably, the complexity of rewrite requests naturally encompasses various NLP tasks, such as summarization, information extraction, and semantic understanding. Addressing these tasks requires LLM to demonstrate advanced reasoning and comprehension abilities, thereby enhancing its proficiency in instruction-following, contextual understanding, and reasoning (as verified in Table 3).

3.2 Web as Response

Inspired by instruction backtranslation (Li et al., 2024d), we propose an alternative approach to reconstruct web content by treating the web as a response. Specifically, we utilize a LLM to predict a latent instruction for which the raw web content would serve as an ideal response, as illustrated in Figure 10. To further enhance diversity, specific segments of web content are treated as responses (with a 50% probability), as depicted in Figure 11. Unlike traditional back-translation methods, which directly treat latent instructions and raw web content as instruction-response pairs, **our approach introduces a two-stage refinement process**. First, we generate an initial response by rolling out an LLM prediction for the latent instruction. Next, we refine this response using both the raw web content and the latent instruction to produce a more accurate and comprehensive output, as shown in Figure 12. The initial rollout ensures that the response exhibits human-like fluency and natural language style, while the subsequent refinement step integrates critical information from the raw web, ensuring that the final response is both precise and thorough. This dual-stage process significantly enhances the LLM's performance in knowledge acquisition and question-answering tasks, as demonstrated by the improvements reported in Table 3. The generated instruction as well as the refined response are finally paired as IT data.

3.3 Dataset Construction Details

Following existing work (Li et al., 2024d; Yue et al., 2024), we construct our dataset by sampling raw web documents from three diverse and representative domains: 70% from the English subset of Common Crawl (Computer, 2023) (general domain), 15% from OpenWebMath (Paster et al., 2024) (math domain), and 15% from GitHub (Computer, 2023) (code domain). To enable large-scale creation of diverse synthetic data for various scenarios, we adopt a persona-driven instruction synthesis strategy inspired by Ge et al. (2024). Initially, an LLM generates personas for the raw web documents (see template in Figure 7), which guide the subsequent instruction synthesis for our proposed Web Reconstruction process. The ratio of *Web as Instruction* to *Web as Response* is set to 2:1, following insights from the ablation study presented in Table 3. To enhance diversity and eliminate redundancy, we apply MinHash (Broder, 1997) deduplication based on n-gram features of instructions. We configure the signature size to 128 and the similarity threshold to 0.7. The final synthesized dataset comprises 100,000 instruction-response pairs.

To evaluate the effectiveness of WebR in generating high-quality IT datasets, we use WebR to construct datasets with two LLMs: the open-source `Llama3-70B-Instruct` (Dubey et al., 2024) (temperature=0.6, top-p=0.9) and the proprietary `GPT-4o-mini` (Achiam et al., 2023) (temperature=0.7, top-p=1.0). The resulting datasets, **WebR-Basic** (from Llama3) and **WebR-Pro** (from GPT-4o-mini), differ in their generative capabilities and quality. A comparative analysis of the average token lengths is presented in Appendix C, while a detailed cost analysis of WebR is provided in Appendix D. Notably, the overall expenditure for calling GPT-4o-mini API is **$38.57**.

3.4 Dataset Analysis

Coverage. We evaluate the coverage of WebR in the embedding space using a comparative analysis with existing datasets. Specifically, we leverage the `all-mpnet-base-v2` embedding model[1] to compute the input embeddings of instructions and utilize t-SNE (Van der Maaten and Hinton, 2008) to project these embeddings into a two-dimensional space for visualization. To provide meaningful baselines, we include human-curated ShareGPT (Chiang et al., 2023), semi-automated synthetic UltraChat (Ding et al., 2023), and Alpaca (Taori et al., 2023). As shown in Figure 3, the t-SNE visualization reveals that WebR-Basic comprehensively spans the primary regions covered by ShareGPT, UltraChat, and Alpaca. This observation highlights that WebR-Basic effectively captures a diverse range of topics, aligning well with both human-curated and synthetic datasets.

[1] https://huggingface.co/sentence-transformers/all-mpnet-base-v2

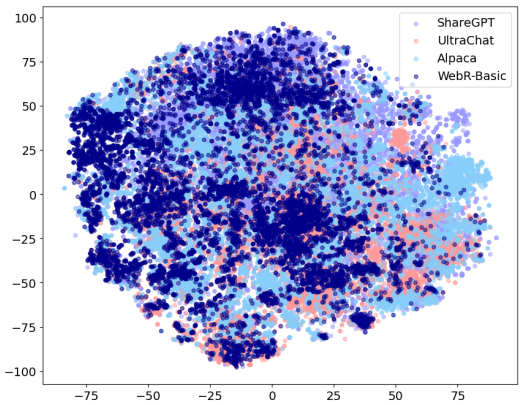

Figure 3: This figure compares the t-SNE plot of WebR-Basic with those of ShareGPT, UltraChat, and Alpaca, each of which is sampled with 10,000 instructions.

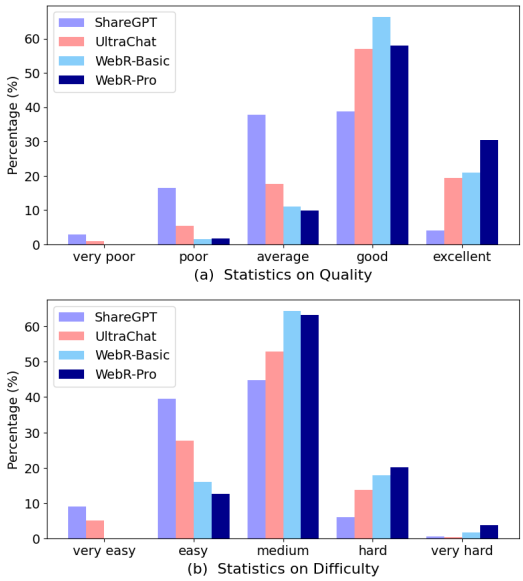

Figure 4: Statistics of instruction quality and difficulty.

Quality and Difficulty. Following Magpie (Xu et al., 2024b), we use the Qwen2.5-72B-Instruct model to evaluate the quality and difficulty of each instruction, categorizing them into five levels. As depicted in Figure 4, synthetic data generally demonstrates higher quality and greater difficulty compared to human-crafted instructions. In particular, WebR-Basic and WebR-Pro exhibit superior distributions in both quality and difficulty metrics, surpassing UltraChat in these aspects.

4 Experimental Setup

4.1 Baselines

We compare the family of IT datasets generated by WebR with ten state-of-the-art (SOTA) open-source IT datasets, categorized as follows: (1) Human-crafted data: **ShareGPT** (Chiang et al., 2023) and **WildChat** (Zhao et al., 2024) are exemplary human-written datasets containing 112K and 652K high-quality multi-round conversations between humans and GPT, respectively. (2) Semi-automated synthetic data: **Alpaca** (Taori et al., 2023), **Evol-Instruct** (Xu et al., 2024a), and **UltraChat** (Ding et al., 2023) represent widely-used synthetic datasets generated with semi-automated techniques. (3) Mixed data: **Tulu V2 Mix** (Ivison et al., 2023) and **OpenHermes 2.5** (Teknium, 2023) are crowd-sourced datasets that aggregate diverse open-source IT datasets, featuring 326K and 1M conversations, respectively. (4) Fully automated synthetic data: **WebInstruct** (Yue et al., 2024) consists of QA pairs extracted from web corpora, from which we sample 100k examples for our experiments. Additionally, we reproduce **Back Translation** (Li et al., 2024d) using the same source web data as our WebR, based on Llama3-70B-Instruct. Furthermore, **Magpie** (Xu et al., 2024b) synthesizes IT data by prompting Llama3-70B-Instruct with its chat template, from which we similarly sample 100k examples.

4.2 Models and Training Settings

For instruction tuning (IT), we train Llama3-8B-base (Dubey et al., 2024) and Qwen2.5-1.5/3/7/14B-base (Qwen Team, 2024) on various IT datasets. We adhere to the official instruction templates provided by each model. To ensure a fair comparison, we use consistent training hyper-parameters across different baseline datasets. The comprehensive implementation details are listed in Appendix A.

4.3 Evaluation Benchmarks and Metrics

We evaluate the performance of the fine-tuned models using four widely adopted instruction-following benchmarks: AlpacaEval 2 (Li et al., 2023), Arena-Hard (Li et al., 2024c), MT-Bench (Zheng et al., 2023), and IFEval (Zhou et al., 2023b). For AlpacaEval 2, we report the length-controlled win rate (LC), which ensures robustness against verbosity. For Arena-Hard, we report the win rate (WR) against the baseline model. For MT-Bench, we provide the average score, using GPT-4-turbo as the evaluation judge. For IFEval, we report two metrics: prompt-level strict accuracy (*Pr. (S)*) and instruction-level strict accuracy (*Ins. (S)*). More evaluation details are listed in Appendix B.

IT Data	#Data	Human Effort	Response Generator	Alpaca Eval 2	Arena Hard	MT Bench	IFEval Pr. (S)	IFEval Ins. (S)	Avg.
ShareGPT	112k	High	ChatGPT	9.89	6.49	6.34	38.52	42.26	22.70
WildChat	652k	High	GPT-3.5 & 4	14.62	8.73	6.60	39.53	45.66	23.03
Tulu V2 Mix	326k	Mid	Mix	9.91	5.41	5.76	37.69	41.05	19.96
OpenHermes 2.5	1M	Mid	Mix	12.89	8.20	6.51	38.82	43.52	21.99
Alpaca	52k	Low	Davinci-003	4.21	1.24	3.75	20.21	23.56	10.59
Evol Instruct	143k	Low	ChatGPT	7.19	5.58	5.77	39.00	44.25	20.36
UltraChat	208k	Low	ChatGPT	8.29	4.06	5.88	29.66	33.06	16.19
WebInstruct	100k	No	Qwen-72B	3.03	1.62	5.03	18.85	20.42	9.79
Back Translation	100k	No	Llama3-70B	5.24	2.81	3.74	26.85	29.61	13.65
Magpie	100k	No	Llama3-70B	23.62	13.98	6.26	33.83	43.07	24.15
WebR-Basic	100k	No	Llama3-70B	**25.33**	**16.50**	**6.95**	**41.40**	**50.69**	**28.17**
IT Mix	100k	Mid	GPT-4o-mini	30.19	27.81	7.30	43.07	47.13	31.10
WebR-Pro	100k	No	GPT-4o-mini	**34.17**	**30.92**	**7.50**	**43.55**	**51.77**	**33.58**
(IT + WebR-Pro) Mix	100k	Mid	GPT-4o-mini	35.00	34.23	7.50	48.06	53.23	35.60
(IT + WebR-Pro) Merge	200k	Mid	GPT-4o-mini	**35.40**	**35.12**	**7.59**	**49.72**	**53.97**	**36.36**

Table 1: Instruction-following performance comparison of various instruction-tuning (IT) data, based on Llama3-8B.

IT Data	MMLU	ARC	WinoGrande	MATH	GSM8K	HumanEval	Avg.
WildChat	58.46	72.62	49.43	19.34	60.25	42.55	50.44
OpenHermes 2.5	60.08	75.65	51.22	24.18	64.70	44.43	53.38
Magpie	58.58	71.53	51.93	16.12	57.39	40.85	49.40
WebR-Basic	60.85	76.27	52.91	20.28	55.57	40.10	51.00
IT Mix	57.44	73.56	50.36	22.00	61.87	45.12	51.73
WebR-Pro	**61.15**	74.92	**53.20**	24.94	60.69	48.73	53.94
(IT + WebR-Pro) Mix	60.69	**77.63**	50.67	26.34	64.90	**50.61**	55.14
(IT + WebR-Pro) Merge	61.02	76.27	52.72	**28.36**	**66.41**	**50.61**	**55.90**

Table 2: Performance comparison of downstream tasks (Knowledge, Reasoning, Math, Code) based on Llama3-8B.

5 Experimental Results

5.1 Main Results

WebR Outperforms Prior Baselines. Table 1 highlights the performance of Llama3-8B-base fine-tuned with datasets generated by WebR, compared to those fine-tuned with baseline datasets. A general trend emerges: IT datasets requiring higher human effort tend to exhibit better performance than those with lower or no human effort. Nevertheless, our WebR-Basic, which entirely eliminates human effort in dataset creation, significantly and consistently surpasses the SOTA Magpie dataset across all four benchmarks with a **16.65%** average improvement. To ensure a fair and more challenging comparison, we deduplicated and randomly sampled 100k instructions from baseline datasets of varying human effort levels (high, mid, and low) and generated responses using GPT-4o-mini, naming this synthesized strong baseline "**IT Mix**." Even under the same response generator, WebR-Pro consistently outperforms IT Mix, achieving an average improvement of **7.97%**. These results validate that datasets generated by WebR possess superior quality, enabling significantly enhanced instruction-following performance.

WebR Further Enhances Prior Baselines. To explore the potential synergy between WebR and existing datasets, we merged IT Mix and WebR-Pro using two strategies: (1) random sampling of 50k data points from each dataset and (2) direct concatenation. As shown in Table 1, both merged datasets deliver further performance improvements over their individual components, establishing new SOTA results. This can be attributed to the complementary strengths of the datasets: IT Mix offers broader data coverage, while WebR-Pro provides higher quality and more challenging instructions, as evidenced in Figure 3 and Figure 4.

Performance on Downstream Benchmarks. We evaluate the impact of various instruction-tuning datasets on downstream task performance across multiple domains[2]: (1) **Knowledge**: MMLU (Hendrycks et al., 2021a); (2) **Reasoning**: ARC (Clark et al., 2018) and WinoGrande (Sak-

[2]Evaluation settings are aligned with https://opencompass.org.cn.

Setting	Alpaca Eval 2	MT Bench	IFEval Pr. (S)	Avg.	MMLU	ARC	MATH	HumanEval	Avg.
WebR-Pro	34.17	7.50	43.55	**28.41**	61.15	74.92	24.94	48.73	**52.43**
-w/o Persona	33.30	6.93	44.69	28.31	60.98	74.58	24.03	48.50	52.02
-w/o Part	33.89	7.53	42.60	28.01	61.05	72.53	22.73	48.41	51.18
-w/o Refinement	31.61	7.42	44.73	27.92	59.83	74.92	24.36	48.61	51.93
-w/o MinHash	32.43	7.29	43.02	27.58	60.69	74.92	24.82	47.15	51.90
Ratio of Web as Instruction to Web as Response (2 : 1 in WebR)									
1 : 0	29.15	7.10	39.56	25.27	58.79	74.58	25.74	50.00	52.28
1 : 1	33.16	7.39	43.26	27.94	60.60	73.22	25.18	48.78	51.95
1 : 2	32.99	7.33	42.85	27.72	57.76	72.61	25.26	50.00	51.41
0 : 1	33.41	6.68	42.54	27.54	52.68	72.90	23.30	46.95	48.96

Table 3: Ablation study based on Llama3-8B.

Base LLM	IT Data	AlpacaEval 2	Arena-Hard	MT-Bench	IFEval/Pr. (S)	IFEval/Ins. (S)
Qwen2.5-1.5B	IT Mix	10.98	**15.10**	**6.03**	**29.57**	**33.27**
	WebR-Pro	**11.00** (+0.02)	14.03 (-1.07)	5.92 (-0.11)	**29.57** (+0.00)	32.16 (-1.11)
Qwen2.5-3B	IT Mix	**22.36**	26.54	6.95	**43.07**	**44.73**
	WebR-Pro	22.29 (-0.07)	**28.13** (+1.59)	**7.03** (+0.08)	42.38 (-0.69)	44.71 (-0.02)
Qwen2.5-7B	IT Mix	32.59	45.10	7.45	49.35	52.68
	WebR-Pro	**34.90** (+2.31)	**45.66** (+0.56)	**7.62** (+0.17)	**50.55** (+1.20)	**53.35** (+0.67)
Qwen2.5-14B	IT Mix	42.07	59.00	8.10	58.04	60.63
	WebR-Pro	**46.19** (+4.12)	**62.13** (+2.13)	**8.39** (+0.29)	**60.23** (+2.19)	**64.88** (+4.25)

Table 4: Performance comparison across varied scales of base LLMs.

aguchi et al., 2019); (3) **Math**: MATH (Hendrycks et al., 2021b) and GSM8K (Cobbe et al., 2021); (4) **Code**: HumanEval (Chen et al., 2021). As shown in Table 2, models fine-tuned on the WebR datasets outperform those trained on other baselines, demonstrating their effectiveness in improving generalization across diverse downstream tasks, especially in challenging benchmarks like ARC and WinoGrande. Furthermore, the combination of WebR-Pro and IT Mix further validates the complementary strengths of WebR data in aligning models with complex task requirements.

5.2 Ablation Study

Table 3 compares the LLM performance using different settings to construct WebR-Pro.

- **w/o Persona**: removing the author's persona information during instruction generation leads to performance declines across almost all benchmarks.
- **w/o Part**: creating instructions solely from the entire web content, rather than using specific parts, causes notable performance degradation, particularly on IFEval and reasoning-intensive tasks like ARC and MATH.
- **w/o Refinement**: skipping the refinement step for *Web as Response*—by directly adopting the rollout response as the final output—results in a substantial drop in instruction-following performance.
- **w/o MinHash**: eliminating MinHash-based deduplication decreases performance across all benchmarks, highlighting the importance of maintaining dataset diversity.
- **Ratio of *Web as Instruction* to *Web as Response***: varying the ratio of *Web as Instruction* to *Web as Response* data synthesis reveals that **each component contributes uniquely to model capabilities**. Specifically, *Web as Instruction* enhances reasoning and understanding tasks (e.g., ARC and MATH), while *Web as Response* primarily improves instruction-following and question-answering tasks (e.g., IFEval and AlpacaEval 2). The optimal balance is achieved at a ratio of 2:1, which delivers the best overall performance.

6 Analysis

6.1 Impact of Base LLM Scale

Table 4 highlights the impact of base LLM scale on the performance of our proposed WebR method. While WebR-Pro slightly underperforms IT Mix at the 1.5B model scale, its advantages become

Data Proportion	AlpacaEval 2	MATH	HumanEval	MedQA	FinBen	Avg.
IT Mix	30.19	22.00	45.12	38.88	29.20	33.08
WebR-Pro (4.7 gen : 1 math : 1 code)	34.17	24.94	48.73	47.31	29.56	36.94
- 1 gen	34.40	22.52	44.78	44.94	28.97	35.12
- 1 gen : 1 math	34.25	**28.09**	48.23	46.59	29.77	37.39
- 1 gen : 1 math : 1 code	34.59	27.10	**51.39**	46.83	29.34	**37.85**
- 1 gen : 1 math : 1 code : 1 med	32.75	26.22	49.68	**49.98**	29.01	37.53
- 1 gen : 1 math : 1 code : 1 med : 1 fin	33.03	25.38	48.17	45.64	**30.22**	36.49

Table 5: Domain adaptation based on Llama3-8B, with the domain improvements marked in **green**.

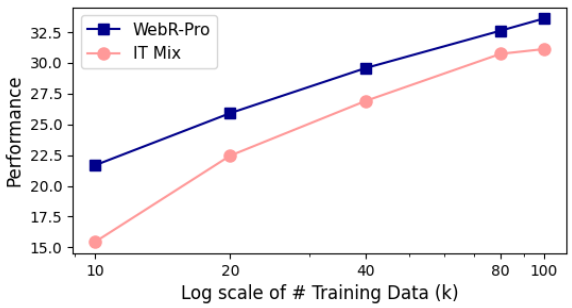

Figure 5: The impact of training data scale on the average instruction-following performance.

increasingly pronounced as the model size grows. For instance, WebR-Pro achieves an average performance improvement of **2.86%** over IT Mix with Qwen2.5-7B and an even more substantial improvement of **5.55%** with Qwen2.5-14B. These results suggest that the advanced synthesis paradigm of WebR better aligns with larger models' capacity to capture complex patterns and utilize reasoning-intensive data. In contrast, smaller models with limited capacity may struggle to fully exploit WebR's potential.

6.2 Impact of Training Data Scale

Figure 5 illustrates the impact of training data scale on model performance. The results clearly underscore the superior **data efficiency** of WebR-Pro compared to IT Mix: (1) With only 10k training samples, WebR-Pro achieves a striking **40.26%** performance improvement over IT Mix, highlighting its exceptional capability to elicit latent potential from LLMs even with limited data. (2) WebR-Pro exhibits a more consistent and pronounced linear performance increase with respect to the logarithmic growth in training data, consistently outperforming IT Mix across all data scales. These results strongly validate the efficacy of WebR in efficiently leveraging training data to unlock and enhance the capabilities of LLMs.

6.3 Domain Adaptation

We explore the potential of our proposed WebR framework for domain adaptation by incrementally incorporating domain-specific web documents into the training data. Starting with general-domain content, we progressively add domain-specific materials from math, code, medicine, and finance, assessing performance across relevant benchmarks. For the medical and financial domains, we utilize raw web documents from IndustryCorpus2 (Shi et al., 2024), and evaluate using MedQA (Jin et al., 2021) and FinBen (Xie et al., 2024) benchmarks. As shown in Table 5, WebR demonstrates strong adaptability across domains. Compared to the IT Mix baseline, incorporating domain-specific data consistently improves performance, with math and code data yielding significant gains in MATH (28.09) and HumanEval (51.39), and medical and financial domains showing strong results on MedQA (49.98) and FinBen (30.22). These results highlight WebR's ability to **incorporate specialized knowledge while maintaining competitive general-domain performance**. Furthermore, the process of collecting domain-specific web documents is straightforward, underscoring WebR's practicality.

7 Conclusion

In this paper, we present **Web Reconstruction** (WebR), a fully automated framework for synthesizing high-quality instruction-tuning (IT) datasets. Harnessing the richness of raw web content, we conceptualize *web reconstruction* as an instruction-tuning data synthesis task via a novel dual-perspective paradigm—*Web as Instruction* and *Web as Response*—where each web document is designated as either the input or output role to trigger the reconstruction process. Extensive experiments show that WebR-generated datasets consistently outperform state-of-the-art baselines across four instruction-following benchmarks and six diverse downstream tasks.

Limitations

While WebR can already obtain satisfactory performance, there are several areas for improvement and future exploration. Firstly, the current implementation of WebR focuses on single-turn data synthesis. Expanding this framework to support multi-turn conversations could further enhance its applicability to complex, interactive tasks. Second, due to constraints in time and computational resources, the size of the constructed WebR-Basic and WebR-Pro datasets is currently limited to 100k samples. However, given the vast availability of web documents—numbering in the trillions—the WebR framework has significant potential for scaling to create large-scale IT datasets, which could further boost performance. Finally, WebR does not incorporate advanced data selection techniques, such as Instruction Following Difficulty (IFD) (Li et al., 2024b), as part of its post-processing pipeline. Incorporating such techniques in future work could refine data quality and further enhance the capabilities of LLMs.

Ethics Statement

This study adheres strictly to the ethical principles established by the research community. The utilized IT datasets are reported to be safe and free of content that may contain discrimination, personally identifiable information, or any other undesirable behaviors. We have meticulously designed and curated our instructions for LLMs to ensure that all tasks are restricted to web reconstruction. This approach minimizes the risk of generating content that could raise ethical concerns.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Andrei Z Broder. 1997. On the resemblance and containment of documents. In *Proceedings. Compression and Complexity of SEQUENCES 1997 (Cat. No. 97TB100171)*, pages 21–29. IEEE.

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*.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. 2021. Evaluating large language models trained on code.

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.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *Preprint*, arXiv:1803.05457.

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*.

Together Computer. 2023. Redpajama: An open source recipe to reproduce llama training dataset.

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. 2023. Free dolly: Introducing the world's first truly open instruction-tuned llm.

Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Zhi Zheng, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. 2023. Enhancing chat language models by scaling high-quality instructional conversations. *arXiv preprint arXiv:2305.14233*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Tao Ge, Xin Chan, Xiaoyang Wang, Dian Yu, Haitao Mi, and Dong Yu. 2024. Scaling synthetic data creation with 1,000,000,000 personas. *arXiv preprint arXiv:2406.20094*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021a. Measuring massive multitask language understanding. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

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

Hamish Ivison, Yizhong Wang, Valentina Pyatkin, Nathan Lambert, Matthew Peters, Pradeep Dasigi, Joel Jang, David Wadden, Noah A. Smith, Iz Beltagy, and Hannaneh Hajishirzi. 2023. Camels in a changing climate: Enhancing lm adaptation with tulu 2. *Preprint*, arXiv:2311.10702.

Yuxin Jiang, Chunkit Chan, Mingyang Chen, and Wei Wang. 2023. Lion: Adversarial distillation of proprietary large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3134–3154, Singapore. Association for Computational Linguistics.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2021. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14):6421.

Haoran Li, Qingxiu Dong, Zhengyang Tang, Chaojun Wang, Xingxing Zhang, Haoyang Huang, Shaohan Huang, Xiaolong Huang, Zeqiang Huang, Dongdong Zhang, et al. 2024a. Synthetic data (almost) from scratch: Generalized instruction tuning for language models. *arXiv preprint arXiv:2402.13064*.

Ming Li, Yong Zhang, Zhitao Li, Jiuhai Chen, Lichang Chen, Ning Cheng, Jianzong Wang, Tianyi Zhou, and Jing Xiao. 2024b. From quantity to quality: Boosting LLM performance with self-guided data selection for instruction tuning. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7602–7635, Mexico City, Mexico. Association for Computational Linguistics.

Tianle Li, Wei-Lin Chiang, Evan Frick, Lisa Dunlap, Banghua Zhu, Joseph E Gonzalez, and Ion Stoica. 2024c. From live data to high-quality benchmarks: The arena-hard pipeline.

Xian Li, Ping Yu, Chunting Zhou, Timo Schick, Omer Levy, Luke Zettlemoyer, Jason E Weston, and Mike Lewis. 2024d. Self-alignment with instruction backtranslation. In *The Twelfth International Conference on Learning Representations*.

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023. Alpacaeval: An automatic evaluator of instruction-following models.

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)*, pages 3470–3487, Dublin, Ireland. Association for Computational Linguistics.

Thao Nguyen, Jeffrey Li, Sewoong Oh, Ludwig Schmidt, Jason Weston, Luke Zettlemoyer, and Xian Li. 2024. Better alignment with instruction back-and-forth translation. *Preprint*, arXiv:2408.04614.

TB OpenAI. 2022. Chatgpt: Optimizing language models for dialogue. *OpenAI*.

Keiran Paster, Marco Dos Santos, Zhangir Azerbayev, and Jimmy Ba. 2024. Openwebmath: An open dataset of high-quality mathematical web text. In *The Twelfth International Conference on Learning Representations*.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Winogrande: An adversarial winograd schema challenge at scale. *arXiv preprint arXiv:1907.10641*.

Xiaofeng Shi, Lulu Zhao, Hua Zhou, and Donglin Hao. 2024. Industrycorpus2.

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`.

Teknium. 2023. Openhermes 2.5: An open dataset of synthetic data for generalist llm assistants.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

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

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Kuntal Kumar Pal, Maitreya Patel, Mehrad Moradshahi, Mihir Parmar, Mirali Purohit, Neeraj Varshney, Phani Rohitha Kaza, Pulkit Verma, Ravsehaj Singh Puri, Rushang Karia, Savan Doshi, Shailaja Keyur Sampat, Siddhartha Mishra, Sujan Reddy A, Sumanta Patro, Tanay Dixit, and Xudong Shen. 2022. Super-NaturalInstructions: Generalization via declarative instructions on 1600+ NLP tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5085–5109, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jason Wei, Maarten Bosma, Vincent 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 *International Conference on Learning Representations*.

Qianqian Xie, Weiguang Han, Zhengyu Chen, Ruoyu Xiang, Xiao Zhang, Yueru He, Mengxi Xiao, Dong Li, Yongfu Dai, Duanyu Feng, Yijing Xu, Haoqiang Kang, Ziyan Kuang, Chenhan Yuan, Kailai Yang, Zheheng Luo, Tianlin Zhang, Zhiwei Liu, GUOJUN XIONG, Zhiyang Deng, Yuechen Jiang, Zhiyuan Yao, Haohang Li, Yangyang Yu, Gang Hu, Huang Jiajia, Xiao-Yang Liu, Alejandro Lopez-Lira, Benyou Wang, Yanzhao Lai, Hao Wang, Min Peng, Sophia Ananiadou, and Jimin Huang. 2024. Finben: An holistic financial benchmark for large language models. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, Qingwei Lin, and Daxin Jiang. 2024a. WizardLM: Empowering large pre-trained language models to follow complex instructions. In *The Twelfth International Conference on Learning Representations*.

Zhangchen Xu, Fengqing Jiang, Luyao Niu, Yuntian Deng, Radha Poovendran, Yejin Choi, and Bill Yuchen Lin. 2024b. Magpie: Alignment data synthesis from scratch by prompting aligned llms with nothing. *arXiv preprint arXiv:2406.08464*.

Xiang Yue, Tuney Zheng, Ge Zhang, and Wenhu Chen. 2024. Mammoth2: Scaling instructions from the web. *Advances in Neural Information Processing Systems*.

Wenting Zhao, Xiang Ren, Jack Hessel, Claire Cardie, Yejin Choi, and Yuntian Deng. 2024. Wildchat: 1m chatGPT interaction logs in the wild. In *The Twelfth International Conference on Learning Representations*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

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. 2023a. LIMA: Less is more for alignment. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023b. Instruction-following evaluation for large language models. *Preprint*, arXiv:2311.07911.

A Implementation Details

Our implementation is based on the alignment-handbook repo[3]. The training procedure was executed on 4 NVIDIA A800 GPUs, each equipped with 80GB of memory. The duration required to train a single instance of the model, specifically the Llama3-8B-base, was approximately 9 hours. The specific hyperparameters used during training are detailed in Table 6. Notably, all models were trained using the same set of hyperparameters, except for the maximum sequence length, which was set to 2048 for the 14B LLMs to mitigate computational bottlenecks.

Hyperparameter	Value
Batch size	128
Learning rate	2e-5
Epoches	4
Max length	4096 (2048 for 14B LLMs)
Optimizer	AdamW
Scheduler	cosine
Weight decay	0
Warmup ratio	0.1

Table 6: Training hyperparameters for Llama3-8B-base and Qwen2.5-1.5/3/7/14B-base.

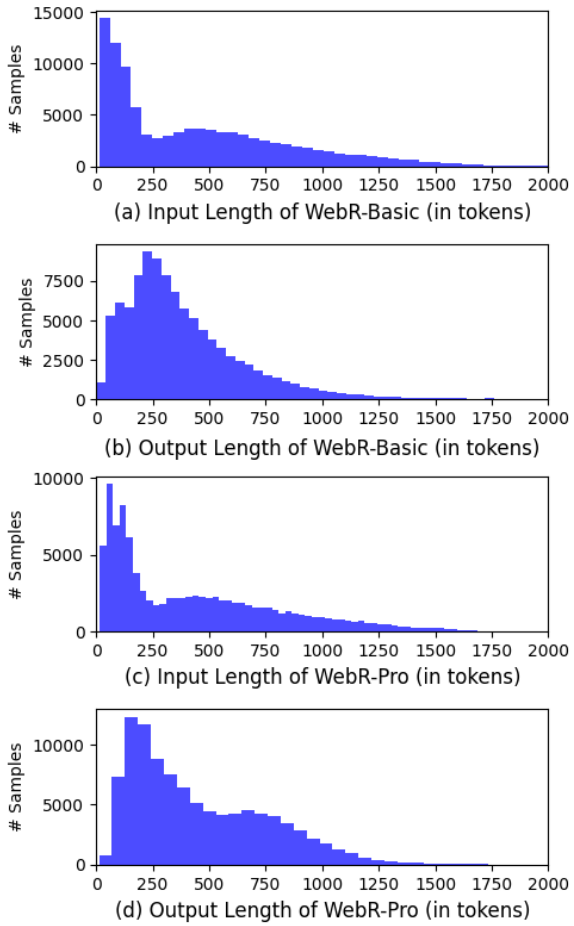

Figure 6: Lengths of instructions and responses in WebR-Basic and WebR-Pro.

B Evaluation Details

Table 7 lists the evaluation details for AlpacaEval 2 (Li et al., 2023), Arena-Hard (Li et al., 2024c), MT-Bench (Zheng et al., 2023), and IFEval (Zhou et al., 2023b). AlpacaEval 2 comprises 805 questions from 5 datasets, and MT-Bench spans 8 categories with a total of 80 questions. Arena-Hard is an enhanced version of MT-Bench, featuring 500 well-defined technical problem-solving queries. IFEval consists of 541 samples, each containing 1 to 3 verifiable constraints. Evaluation metrics are reported in accordance with each benchmark's protocol.

C Dataset Analysis

Statistics including token lengths of instructions and responses are illustrated in Figure 6. Tokens are counted using the `tiktoken` library[4]. For WebR-Basic, the average token lengths of instructions and responses are 441.41 and 381.28, respectively. For WebR, the average token lengths of instructions and responses are 439.88 and 457.34, respectively.

D Cost Analysis

Here we analyze the cost-effectiveness of our proposed Web Reconstruction framework. For context, we estimated the budget for data synthesis using the `GPT-4o-mini` API, based on the Batch API's pricing of $0.075 per 1M input tokens and $0.3 per 1M output tokens. Table 8 lists the breakdown of the estimated costs for each step, which demonstrates that the overall expenditure (**$38.57**) is both reasonable and manageable.

Additionally, our main experiment in Table 1 demonstrates that the open-source `Llama3-70B-Instruct` model can achieve satisfactory performance for our proposed Web Reconstruction, significantly outperforming previous IT datasets. Notably, `Llama3-70B-Instruct` can be deployed on only 2 NVIDIA-3090 GPUs, with

[3] https://github.com/huggingface/alignment-handbook

[4] https://github.com/openai/tiktoken

Benchmark	# Exs.	Baseline Model	Judge Model	Scoring Type	Metric
AlpacaEval 2	805	GPT-4 Turbo	GPT-4 Turbo	Pairwise comparison	Length-controlled win rate
Arena-Hard	500	GPT-4-0314	GPT-4 Turbo	Pairwise comparison	Win rate
MT-Bench	80	-	GPT-4/GPT-4 Turbo	Single-answer grading	Rating of 1-10
IFEval	541	-	-	Rule-based verification	Accuracy

Table 7: Evaluation details for AlpacaEval 2 (Li et al., 2023), Arena-Hard (Li et al., 2024c), MT-Bench (Zheng et al., 2023), and IFEval (Zhou et al., 2023b). The baseline model refers to the model compared against.

	# of Samples	Avg. Input Token Length	Avg. Output Token Length	Cost ($)
Generate author's persona	100,000	523	32	4.88
Web as Instruction (instruction)	66,667	711	123	6.02
Web as Instruction (rollout response)	66,667	611	392	10.90
Web as Response (instruction)	33,333	645	91	2.52
Web as Response (rollout response)	33,333	91	522	5.45
Web as Response (refined response)	33,333	1,155	591	8.80
Total	-	-	-	38.57

Table 8: Estimated budget for data synthesis using the `GPT-4o-mini` API.

the option to further reduce hardware requirements through low-bit quantization[5]. This provides an economical alternative for our proposed WebR. In conclusion, our framework demonstrates robustness in leveraging diverse LLMs for data synthesis, confirming its adaptability and effectiveness.

E Prompt Template

Figure 7 shows the prompt template for generating the author persona according to the web content. Figure 8 shows the prompt template for generating the rewrite request based on the whole web content. Figure 9 shows the prompt template for generating the rewrite request based on a specific part of the web content. Figure 10 shows the prompt template for generating the latent instruction corresponding to the whole web content. Figure 11 shows the prompt template for generating the latent instruction corresponding to a specific part of the web content. Figure 12 shows the prompt template for generating a refined response based on the raw web and the instruction.

[5] https://github.com/ollama/ollama

Prompt Template for Author Persona

[Text]
{web}

[Instruction]
The text above is from an English webpage. According to the text, please infer the author's profile (within 30 words).

Figure 7: Prompt template for generating author persona.

Prompt Template for Web as Instruction (all)

[Text]
{web}

[Author of the Text]
{persona}

[Instruction]
The text above is from an English webpage. Imagine that you are a user of an AI assistant, please provide a rewrite request specifically designed based on the text content, to create a new version of the text. You can ask for the rewrite to follow constraints including word/sentence/paragraph length, style, format, structure, etc. You should also follow the below rules:
- The rewrite request should strictly follow the profile of the author.
- The rewrite request should be based on the above text, rather than an isolated instruction.
- The constraints should be detailed and specific.
- Output only the request.
- Do **not** directly use the keyword 'rewrite' and 'new version' in the generated request.
- Make sure the generated request is within **{len_limit}** words.

Figure 8: Prompt template for *Web as Instruction* (generating the rewrite request based on the whole web content).

Prompt Template for Web as Instruction (part)

[Text]
{web}

[Author of the Text]
{persona}

[Instruction]
The text above is from an English webpage. Imagine that you are a user of an AI assistant, please provide a rewrite request specifically designed based on the text content, to create a new version of the text focusing on a specific part of information, rather than global information, in the given text above. You can ask for the rewrite to follow constraints including word/sentence/paragraph length, style, format, structure, etc. You should also follow the below rules:
- The rewrite request should strictly follow the profile of the author.
- The rewrite request should be based on the above text, rather than an isolated instruction.
- The constraints should be detailed and specific.
- Output only the request.
- Do **not** directly use the keyword 'rewrite', 'new version', and 'specific part information' in the generated request.
- Make sure the generated request is within **{len_limit}** words.

Figure 9: Prompt template for *Web as Instruction* (generating the rewrite request based on the specific part of the web content).

Prompt Template for Web as Response (all)

[Text]
{web}

[Author of the Text]
{persona}

[Instruction]
Imagine that you are a user of an AI assistant, please provide the most likely request to which the text above would be a great answer. You should also follow the below rules:
- The request should strictly follow the profile of the author.
- Ensure your request is detailed, specific (including the style, format, and structure of the text), clear, and concise.
- Output only the request.
- Make sure the generated request is within **{len_limit}** words.

Figure 10: Prompt template for *Web as Response* (generating the latent instruction based on the whole web content).

Prompt Template for Web as Response (part)

[Text]
{web}

[Author of the Text]
{persona}

[Instruction]
Imagine that you are a user of an AI assistant, please provide the most likely request to which **a specific part of the text above** would be a great answer. You should also follow the below rules:
- The request should strictly follow the profile of the author.
- Ensure your request is detailed, specific (including the style, format, and structure of the text), clear, and concise.
- Output only the request.
- Make sure the generated request is within **{len_limit}** words.

Figure 11: Prompt template for *Web as Response* (generating the latent instruction based on the specific part of the web content).

Prompt Template for Web as Response (Answer Refinement)

Based on the Provided Information, please improve the Answer to the Question, so that the improved answer is of high quality and factually correct. Only output the improved answer.

[Provided Information]
{web}

[Question]
{request}

[Answer]
{answer}

Figure 12: Prompt template for *Web as Response* (answer refinement).