

---

# MAGPIE: Alignment Data Synthesis from Scratch by Prompting Aligned LLMs with Nothing

---

Zhangchen Xu<sup>♣</sup>    Fengqing Jiang<sup>♣</sup>    Luyao Niu<sup>♣</sup>    Yuntian Deng<sup>◇</sup>

Radha Poovendran<sup>♣</sup>    Yejin Choi<sup>♣◇</sup>    Bill Yuchen Lin<sup>◇</sup>

<sup>♣</sup>University of Washington    <sup>◇</sup>Allen Institute for AI

 <https://magpie-align.github.io/>

 <https://hf.co/magpie-align>

## Abstract

High-quality instruction data is critical for aligning large language models (LLMs). Although some models, such as Llama-3-Instruct, have open weights, their alignment data remain private, which hinders the democratization of AI. High human labor costs and a limited, predefined scope for prompting prevent existing open-source data creation methods from scaling effectively, potentially limiting the diversity and quality of public alignment datasets. Is it possible to synthesize high-quality instruction data at scale by extracting it directly from an aligned LLM? We present a *self-synthesis* method for generating large-scale alignment data named MAGPIE. Our key observation is that aligned LLMs like Llama-3-Instruct can generate a user query when we input only the left-side templates up to the position reserved for user messages, thanks to their auto-regressive nature. We use this method to prompt Llama-3-Instruct and generate 4 million instructions along with their corresponding responses. We perform a comprehensive analysis of the extracted data and select 300K high-quality instances. To compare MAGPIE data with other public instruction datasets (e.g., ShareGPT, WildChat, Evol-Instruct, UltraChat, OpenHermes, Tulu-V2-Mix), we fine-tune Llama-3-8B-Base with each dataset and evaluate the performance of the fine-tuned models. Our results indicate that in some tasks, models fine-tuned with MAGPIE perform comparably to the official Llama-3-8B-Instruct, despite the latter being enhanced with 10 million data points through supervised fine-tuning (SFT) and subsequent feedback learning. We also show that using MAGPIE solely for SFT can surpass the performance of previous public datasets utilized for both SFT and preference optimization, such as direct preference optimization with UltraFeedback. This advantage is evident on alignment benchmarks such as AlpacaEval, ArenaHard, and WildBench, and importantly, it is achieved without compromising performance on reasoning tasks like MMLU-Redux, despite the alignment tax.

## 1 Introduction

Large language models (LLMs) such as GPT-4 [1] and Llama-3 [40] have become integral to AI applications due to their exceptional performance on a wide array of tasks by following instructions. The success of LLMs is heavily reliant on the data used for instruction fine-tuning, which equips them to handle a diverse range of tasks, including those not encountered during training. The effectiveness of this instruction tuning depends crucially on access to high-quality instruction datasets. However, the alignment datasets used for fine-tuning models like Llama-3-Instruct are typically private, even when the model weights are open, which impedes the democratization of AI and limits scientific research for understanding and enhancing LLM alignment.

To address the challenges in constructing such datasets, researchers have developed two main approaches. The first type of method involves human effort to generate and curate instruction data

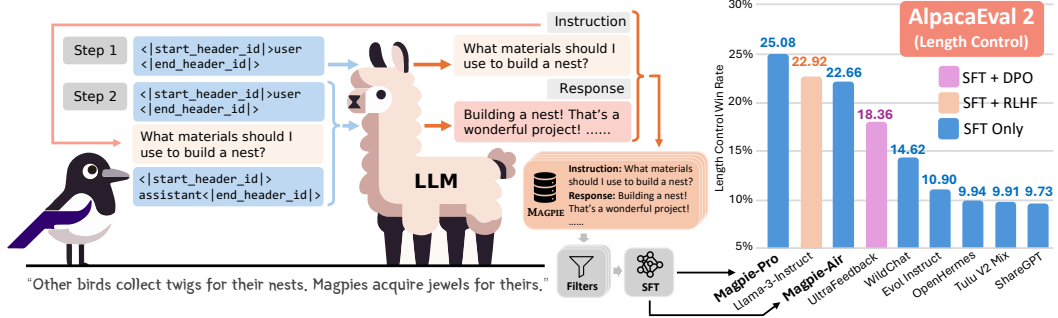


Figure 1: This figure illustrates the process of self-synthesizing instruction data from aligned LLMs (e.g., Llama-3-8B-Instruct) to create a high-quality instruction dataset. In Step 1, we input only the pre-query template into the aligned LLM and generate an instruction along with its response using auto-regressive generation. In Step 2, we use a combination of a post-query template and another pre-query template to wrap the instruction from Step 1, prompting the LLM to generate the query for the second turn. This completes the construction of the instruction dataset. MAGPIE efficiently generates diverse and high-quality instruction data. Our experimental results show that MAGPIE outperforms other public datasets for aligning Llama-3-8B-base.

[14, 26, 64, 65, 66], which is both *time-consuming* and *labor-intensive* [37]. In contrast, the second type of method uses LLMs to produce synthetic instructions [16, 31, 46, 47, 53, 55, 58, 59]. Although these methods reduce human effort, its success heavily depends on prompt engineering and the careful selection of initial seed questions. The *diversity* of synthetic data tends to decrease as the dataset size grows. Despite ongoing efforts, the scalable creation of high-quality and diverse instruction datasets continues to be a challenging problem.

*Is it possible to synthesize high-quality instructions at scale by directly extracting data from advanced aligned LLMs themselves?* A typical input to an aligned LLM contains three key components: the pre-query template, the query, and the post-query template. For instance, an input to Llama-2-chat could be “[INST] Hi! [/INST]”, where [INST] is the pre-query template and [/INST] is the post-query template. These templates are predefined by the creators of the aligned LLMs to ensure the correct prompting of the models. We observe that when we only input the pre-query template to aligned LLMs such as Llama-3-Instruct, they *self-synthesize* a user query due to their auto-regressive nature. Our preliminary experiments indicate that these random user queries are of high quality and great diversity, suggesting that the abilities learned during the alignment process are effectively utilized.

Based on these findings, we developed a self-synthesis method to construct high-quality instruction datasets at scale, named MAGPIE (as illustrated in Figure 1). Unlike existing methods, our approach does not rely on prompt engineering or seed questions. Instead, it *directly* constructs instruction data by prompting aligned LLMs with a pre-query template for sampling instructions. We applied this method to the Llama-3-8B-Instruct and Llama-3-70B-Instruct models, creating two instruction datasets: MAGPIE-Air and MAGPIE-Pro, respectively.

Our MAGPIE-Air and MAGPIE-Pro datasets were created using 206 and 614 GPU hours, respectively, without requiring any human intervention or API access to production LLMs like GPT-4. Additionally, we generated two multi-turn instruction datasets, MAGPIE-Air-MT and MAGPIE-Pro-MT, which contain sequences of multi-turn instructions and responses. The statistics and advantages of our instruction datasets compared to existing ones are summarized in Table 1. We perform a comprehensive analysis of the generated data, allowing practitioners to filter and select data instances from these datasets for fine-tuning according to their particular needs.

To compare MAGPIE data with other public instruction datasets (e.g., ShareGPT [10], WildChat [64], Evol Instruct [58], UltraChat [16], OpenHermes [49], Tulu V2 Mix [24]) and various preference tuning strategies with UltraFeedback [13], we fine-tune the Llama-3-8B-Base model with each dataset and assess the performance of the resultant models on LLM alignment benchmarks such as AlpacaEval 2 [33], Arena-Hard [32], and WildBench [34]. Our results show that models fine-tuned with MAGPIE achieve superior performance, even surpassing the official Llama-3-8B-Instruct model on AlpacaEval, which was fine-tuned with over 10 million data points for supervised fine-tuning (SFT) and follow-up feedback learning. Not only does MAGPIE excel in SFT alone compared to prior public datasets that incorporate both SFT and preference optimization (e.g., direct preference

Table 1: Statistics of instruction datasets generated by MAGPIE compared to other instruction datasets. Tokens are counted using the tiktoken library [42].

Instruction Source	Dataset Name	#Convs	#Turns	Human Effort	Response Generator	#Tokens / Turn	#Total Tokens
Synthetic	Alpaca [47]	52K	1	Low	text-davinci-003	67.38 $\pm$ 54.88	3.5M
	Evol Instruct [58]	143K	1	Low	ChatGPT	473.33 $\pm$ 330.13	68M
	UltraChat [16]	208K	3.16	Low	GhatGPT	376.58 $\pm$ 177.81	238M
Human	Dolly [14]	15K	1	High	ChatGPT	94.61 $\pm$ 135.84	1.42M
	ShareGPT [66]	112K	4.79	High	ChatGPT	465.38 $\pm$ 368.37	201M
	WildChat [64]	652K	2.52	High	GPT-3.5 & GPT-4	727.09 $\pm$ 818.84	852M
	LMSYS-Chat-1M [65]	1M	2.01	High	Mix	260.37 $\pm$ 346.97	496M
Mixture	Deita [38]	9.5K	22.02	-	Mix	372.78 $\pm$ 182.97	74M
	OpenHermes [49]	243K	1	-	Mix	297.86 $\pm$ 258.45	72M
	Tulu V2 Mixture [24]	326K	2.31	-	Mix	411.94 $\pm$ 447.48	285M
MAGPIE	Llama-3-MAGPIE-Air	3M	1	No	Llama-3-8B	426.39 $\pm$ 217.39	1.28B
	Llama-3-MAGPIE-Air-MT	300K	2	No	Llama-3-8B	610.80 $\pm$ 90.61	366M
	Llama-3-MAGPIE-Pro	1M	1	No	Llama-3-70B	478.00 $\pm$ 211.09	477M
	Llama-3-MAGPIE-Pro-MT	300K	2	No	Llama-3-70B	554.53 $\pm$ 133.64	333M

optimization with UltraFeedback [13]), but it also delivers the best results when evaluated against six baseline instruction datasets and four preference tuning methods (DPO [44], IPO [2], KTO [19], and ORPO [23] with the UltraFeedback dataset). These findings show the exceptional quality of instruction data generated by MAGPIE, enabling it to outperform even the official, extensively optimized LLMs.

## 2 MAGPIE: A Scalable Method to Synthesize Instruction Data

**Overview of MAGPIE.** In what follows, we describe our method, MAGPIE, to synthesize instruction data for fine-tuning LLMs. An instance of instruction data consists of at least one or multiple instruction-response pairs. Each pair specifies the roles of instruction provider and follower, along with their instruction and response. As shown in Figure 1, MAGPIE consists of two steps: (1) instruction generation, and (2) response generation. The pipeline of MAGPIE can be fully *automated without any human intervention*. Given the data generated by MAGPIE, practitioners may customize and build their own personalized instruction dataset accordingly (see Section 3 and Appendix B for more details). We detail each step in the following.

**Step 1: Instruction Generation.** The goal of this step is to generate an instruction for each instance of instruction data. Given an open-weight aligned LLM (e.g., Llama-3-70B-Instruct), MAGPIE crafts an input query in the format of the predefined instruction template of the LLM. This query defines only the role of instruction provider (e.g., user), and does not provide any instruction. Note that the auto-regressive LLM has been fine-tuned using instruction data in the format of the predefined instruction template. Thus, the LLM autonomously generates an instruction when the query crafted by MAGPIE is given as an input. MAGPIE stops generating the instruction once the LLM produces an end-of-sequence token. Sending the crafted query to the LLM multiple times leads to a set of instructions. Compared with existing synthetic approaches [16, 31, 47, 53, 55, 58, 59], MAGPIE does not require specific prompt engineering techniques since the crafted query follows the format of the predefined instruction template. In addition, MAGPIE autonomously generates instructions without using any seed question, ensuring the diversity of generated instructions.

**Step 2: Response Generation.** The goal of this step is to generate responses to the instructions obtained from Step 1. MAGPIE sends these instructions to the LLM to generate the corresponding responses. Combining the roles of instruction provider and follower, the instructions from Step 1, and the responses generated in Step 2 yields the instruction dataset. Detailed discussion on the generation configuration can be found in Appendix D.

**Extensions of MAGPIE.** MAGPIE can be readily extended to generate multi-turn instruction datasets and preference datasets. In addition, practitioners can specify the task requested by the instructions. We defer the detailed discussion on these extensions to Appendix A.

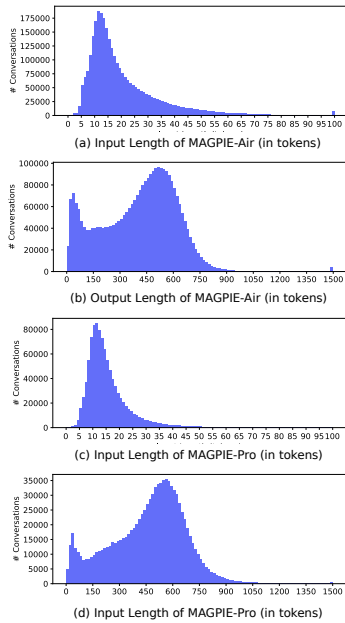


Figure 2: Lengths of instructions and responses in MAGPIE-Air/Pro.

### 3 Dataset Analysis

We apply MAGPIE to the Llama-3-8B-Instruct and Llama-3-70B-Instruct models to construct two instruction datasets: MAGPIE-Air and MAGPIE-Pro, respectively. Examples of instances in both datasets can be found in Appendix G. In this section, we present a comprehensive statistical analysis of the MAGPIE-Air and MAGPIE-Pro datasets. An overview of the lengths of instructions and responses of the data in MAGPIE-Air and MAGPIE-Pro is presented in Figure 2. In what follows, we first assess the breadth of MAGPIE-Pro by analyzing its coverage. We then discuss the attributes of MAGPIE-Pro, including topic coverage, difficulty, quality, and similarity of instructions, as well as quality of response. Finally, we provide the safety analysis and cost analysis. Using our dataset analysis, practitioners can customize and configure their own datasets for fine-tuning LLMs. In Appendix B, we showcase the process of customizing and filtering an instruction dataset based on our analysis. Specifically, we select 300K instances from MAGPIE-Pro and MAGPIE-Air-Filtered, yielding datasets MAGPIE-Pro-300K and MAGPIE-Air-300K-Filtered, respectively.

#### 3.1 Dataset Coverage

We follow the approach in [64] and analyze the coverage of MAGPIE-Pro in the embedding space. Specifically, we use the `all-mpnet-base-v2` embedding model<sup>1</sup> to calculate the input embeddings, and employ t-SNE [51] to project these embeddings into a two-dimensional space. We adopt three synthetic datasets as baselines, including **Alpaca** [47], **Evol Instruct** [58], and **UltraChat** [16], to demonstrate the coverage of MAGPIE-Pro.

Figure 3 presents the t-SNE plots of MAGPIE-Pro, Alpaca, Evol Instruct, and UltraChat. Each t-SNE plot is generated by randomly sampling 10,000 instructions from the associated dataset. We observe that the t-SNE plot of MAGPIE-Pro encompasses the area covered by the plots of Alpaca, Evol Instruct, and UltraChat. This suggests that MAGPIE-Pro provides a broader or more diverse range of topics, highlighting its extensive coverage across varied themes and subjects. We also follow the practice in [53] and present the most common verbs and their top direct noun objects in instructions in Appendix C, indicating the diverse topic coverage of MAGPIE dataset. Coverage analysis of MAGPIE-Air can also be found in Appendix C.

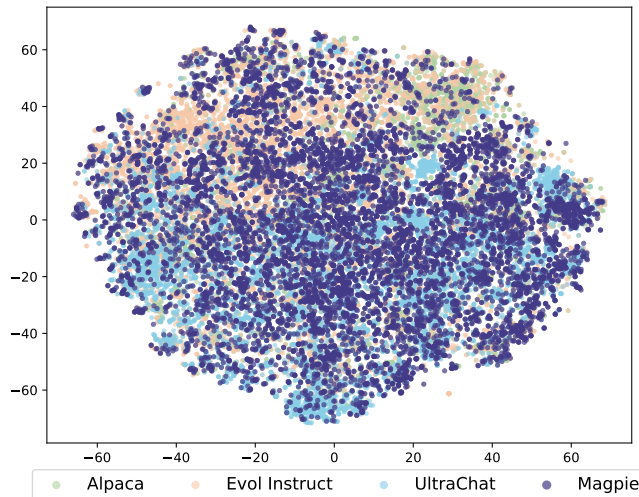


Figure 3: This figure compares the t-SNE plot of MAGPIE-Pro with those of Alpaca, Evol Instruct, and UltraChat, each of which is sampled with 10,000 instructions. The t-SNE plot of MAGPIE-Pro encompasses the area covered by the other plots, demonstrating the comprehensive coverage of MAGPIE-Pro.

<sup>1</sup><https://huggingface.co/sentence-transformers/all-mpnet-base-v2>

### 3.2 Dataset Attributes

#### Attribute: Task Categories of Instructions.

We use Llama-3-8B-Instruct to categorize the instances in MAGPIE-Pro (see Figure 7 in Appendix C.1 for detail). The prompts used to query Llama-3-8B-Instruct can be found in Appendix F. Our observations indicate that over half of the tasks in MAGPIE-Pro pertain to information seeking, making it the predominant category. This is followed by tasks involving creative writing, advice seeking, planning, and math. This distribution over the task categories aligns with the practical requests from human users [33].

**Attribute: Quality of Instructions.** We use the Llama-3-8B-Instruct model to assess the quality of each instruction in MAGPIE-Air and MAGPIE-Pro, categorizing them as ‘very poor’, ‘poor’, ‘average’, ‘good’, and ‘excellent’. We present the histograms of qualities for both datasets in Figure 4-(a). We have the following two observations. First, both datasets are of high quality, with the majority of instances rated ‘average’ or higher. In addition, the overall quality of MAGPIE-Pro surpasses that of MAGPIE-Air. We hypothesize that this is due to the enhanced capabilities of Llama-3-70B compared with Llama-3-8B.

**Attribute: Difficulty of Instructions.** We use the Llama-3-8B-Instruct model to rate the difficulty of each instruction in MAGPIE-Air and MAGPIE-Pro. Each instruction can be labeled as ‘very easy’, ‘easy’, ‘medium’, ‘hard’, or ‘very hard’. Figure 4-(b) presents the histograms of the levels of difficulty for MAGPIE-Air and MAGPIE-Pro. We observe that the distributions across difficulty levels are similar for MAGPIE-Air and MAGPIE-Pro. Some instructions in MAGPIE-Pro are more challenging than those in MAGPIE-Air because MAGPIE-Pro is generated by a more capable model (Llama-3-70B-Instruct).

**Attribute: Instruction Similarity.** We quantify the similarity among instructions generated by MAGPIE to remove repetitive instructions. We measure the similarity using **minimum neighbor distance** in the embedding space. Specifically, we first represent all instructions in the embedding space using the `all-mpnet-base-v2` embedding model. For any given instruction, we then calculate the minimum distance from the instruction to its nearest neighbors in the embedding space using Facebook AI Similarity Search (FAISS) [17]. The minimum neighbor distances of instructions in MAGPIE-Air after removing repetitions are summarized in Figure 5-(a).

**Attribute: Quality of Responses.** We assess the quality of responses using a metric named **reward difference**. For each instance in our dataset, the reward difference is calculated as  $r^* - r_{base}$ , where  $r^*$  is the reward assigned by a reward model to the response in our dataset, and  $r_{base}$  is the reward assigned by the same model to the response generated by the Llama-3 base model for the same instruction. We use URIAL [35] to elicit responses from the base model. A positive reward difference indicates that the response from our dataset is of higher quality, and could potentially benefit instruction tuning. In our experiments, we follow [29] and use `FsfairX-LLaMA3-RM-v0.1` [57] as our reward model. Our results on the reward difference are presented in Figure 5-(b).

### 3.3 Safety Analysis

We use Llama-Guard-2 [48] to analyze the safety of MAGPIE-Air and MAGPIE-Pro. Our results indicate that both datasets are predominantly safe, with less than 1% of the data potentially containing harmful instructions or responses. Please refer to Appendix C.2 for detailed safety analysis.

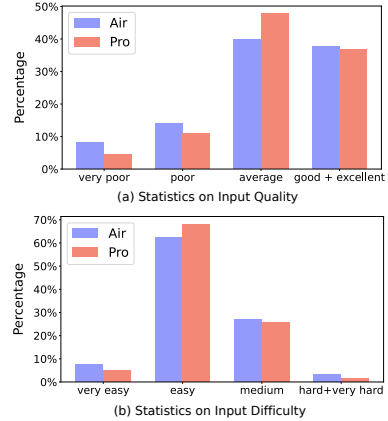


Figure 4: The statistics of input difficulty and quality.

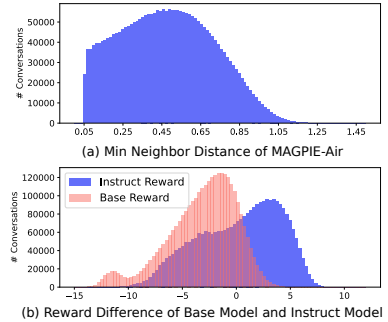


Figure 5: This figure summarizes the minimum neighbor distances and reward differences.

### 3.4 Cost Analysis

We perform experiments on a server with four NVIDIA A100-SXM4-80GB GPUs, an AMD EPYC 7763 64-Core Processor, and 512 GB of RAM, using the VLLM inference framework [28]. The models are loaded in the `bf16` format.

When creating the 3M MAGPIE-Air dataset, our MAGPIE spent 1.55 and 50 hours to generate the instructions (Step 1) and responses (Step 2), respectively. For the 1M MAGPIE-Pro dataset, MAGPIE used 3.5 and 150 hours to generate the instructions (Step 1) and responses (Step 2), respectively. Compared to existing approaches to create instruction datasets, the pipeline of MAGPIE is fully automated without any human intervention or API access to advanced commercial models such as GPT-4 [1]. Consequently, MAGPIE is cost-effective and scalable. On average, implementing MAGPIE on a cloud server<sup>2</sup> would incur costs of **\$0.12** and **\$1.1** per 1,000 data instances for MAGPIE-Air and MAGPIE-Pro, respectively.

### 3.5 Additional Analysis

Additional dataset analysis, including the impact of generation configurations on the quality and difficulty of the generated instructions, is detailed in Appendix C.3.

## 4 Performance Analysis

In this section, we evaluate the quality of datasets generated by MAGPIE by utilizing them to fine-tune model families including Llama-3 [40] and Qwen1.5 [3].

### 4.1 Experimental Setups.

**Baselines for Instruction Tuning.** We compare the family of datasets generated by MAGPIE with six state-of-the-art open-source instruction tuning datasets: **ShareGPT** [10], **WildChat** [64], **Evol Instruct** [58], **UltraChat** [16], **OpenHermes** [49], and **Tulu V2 Mix** [24]. ShareGPT and WildChat are representative human-written datasets containing 112K and 652K high-quality multi-round conversations between humans and GPT, respectively. Evol Instruct and UltraChat are representative open-source synthetic datasets. Following [39], we use the 208K sanitized version of Ultrachat provided by HuggingFace<sup>3</sup>. OpenHermes and Tulu V2 Mix are crowd-sourced datasets consisting of a mix of diverse open-source alignment datasets, with 243K and 326K conversations, respectively. We note that to ensure fair comparison involving datasets of different sizes, we provide the results of MAGPIE-Pro-200K-Filtered and MAGPIE-Pro-100K-Filtered, which contains the first 200K and 100K conversations from MAGPIE-Pro-300K-Filtered. Detailed discussion on how to generate these datasets can be found in Appendix B.

**Baselines for Instruction and Preference Tuning.** We compare the models fine-tuned using data generated by MAGPIE with preference optimization baselines, including DPO [44], IPO [2], KTO [19] and ORPO [23]. Specifically, we follow [39] and use the models fine-tuned with the UltraChat dataset (for instruction tuning) and **Ultrafeedback** dataset (for preference optimization) [13].

**Fine-Tuning Details.** We follow [50] and use a cosine learning rate schedule with an initial learning rate of  $2 \times 10^{-5}$  when fine-tuning Llama-3 and Qwen1.5 models. The maximum sequence length is 8192. The fine-tuning process is conducted using four NVIDIA A100 GPUs with 80G memory, and the effective batch size is 32. The models are fine-tuned for 2 epochs. We follow the official instruction templates of each model.

**Evaluation Benchmarks.** We evaluate the performance of the fine-tuned models using two widely-adopted instruction-following benchmarks: AlpacaEval 2 [33] and Arena-Hard [32]. AlpacaEval 2 consists of 805 representative instructions chosen from real user interactions. Arena-Hard is an enhanced version of MT-Bench [66], containing 500 challenging user queries. Both benchmarks employ a GPT evaluator to assess responses generated by the model of interest and a baseline model. Specifically, we use GPT-4-Turbo (1106) and Llama-3-8B-Instruct as baselines for AlpacaEval 2. By default, Arena-Hard uses GPT-4 (0314) as its baseline model.

<sup>2</sup><https://lambdalabs.com/service/gpu-cloud>

<sup>3</sup>[https://huggingface.co/datasets/HuggingFaceH4/ultrachat\\_200k](https://huggingface.co/datasets/HuggingFaceH4/ultrachat_200k)

Table 2: This table compares the performance of models instruction-tuned on the Llama-8B base models using our datasets and baseline datasets. We observe that models fine-tuned with our datasets significantly outperform those fine-tuned with baseline datasets of the same order of magnitude in terms of data size. In addition, our fine-tuned models achieve comparable performance to the official aligned model, despite only undergoing SFT with a much smaller dataset. Numbers in **bold** indicate that MAGPIE outperforms the official Llama-3-8B-Instruct model.

Alignment Setup (Base LLM = Llama-3-8B)	#Convs	AlpacaEval 2						Arena-Hard	
		GPT-4-Turbo (1106)			Llama-3-8B-Instruct			WR(%)	
		LC (%)	WR (%)	SD	LC (%)	WR (%)	SD		
SFT	+ShareGPT [10]	112K	9.73	7.2	0.81	27.26	18.32	1.18	6.5
	+Evol Instruct [58]	143K	8.52	6.25	0.76	20.16	14.98	1.1	5.1
	+OpenHermes [49]	243K	9.94	6.27	0.73	29.19	17.92	1.16	4.4
	+Tulu V2 Mix [24]	326K	9.91	7.94	0.86	24.28	18.64	1.18	5.4
	+WildChat[64]	652K	14.62	10.58	0.92	34.85	26.57	1.32	8.7
	+UltraChat [16] $\blacktriangledown$	208K	8.29	5.44	0.71	23.95	15.12	1.11	3.6
+*PO	+UltraFeedback (DPO) [13, 44]	64K	18.36	17.33	1.14	44.42	42.36	1.46	14.8
	+UltraFeedback (IPO) [2, 13]	64K	17.46	16.13	1.11	41.66	38.45	1.43	14.2
	+UltraFeedback (KTO) [13, 19]	64K	15.81	14.62	1.05	41.33	38.32	1.42	12.2
	+UltraFeedback (ORPO) [13, 23]	64K	13.23	12.57	0.99	30.62	28.27	1.35	10.9
SFT	<b>+MAGPIE (Ours)</b>								
	Air-300K-Raw	300K	21.99	21.65	1.21	48.63	48.06	1.42	15.8
	Air-300K-Filtered	300K	22.66	<b>23.99</b>	1.24	49.27	<b>50.8</b>	1.44	14.9
	Pro-300K-Raw	300K	21.65	22.19	1.2	49.65	<b>50.84</b>	1.42	15.9
	Pro-100K-Filtered	100K	20.47	<b>24.52</b>	1.25	47.92	<b>52.75</b>	1.43	17.2
	Pro-200K-Filtered	200K	22.11	<b>26.02</b>	1.26	<b>51.17</b>	<b>56.76</b>	1.41	15.9
	Pro-300K-Filtered	300K	<b>25.08</b>	<b>29.47</b>	1.35	<b>52.12</b>	<b>53.43</b>	1.44	18.9
Llama-3-8B-Instruct (SFT+RLHF)		>10M <sup>4</sup>	22.92	22.57	1.26	50	50	-	20.6

**Metrics.** We adopt two metrics to measure the capabilities of instruction-following of fine-tuned models. The first metric is the **win rate (WR)**, which calculates the fraction of responses that are favored by the GPT evaluator. This metric is applied in both benchmarks including AlpacaEval 2 and Arena-Hard. The second metric is the **length-controlled win rate (LC)** [18], a debiased version of WR. The GPT evaluator considers the lengths of responses generated by the baseline model and model under evaluation when computing LC. By accounting for response length, LC reduces its impact on the win rate. This metric is specifically applied to the AlpacaEval 2 benchmark [33].

**Detailed Experimental Setups.** We provide more detailed descriptions of our experimental setups, including more fine-tuning details and decoding hyperparameters in Appendix D.

## 4.2 Experimental Results

### MAGPIE datasets outperform others.

In Table 2, we first compare the performance of Llama-3 models fine-tuned with datasets generated by MAGPIE against those fine-tuned with baseline datasets. Using the AlpacaEval 2 evaluation benchmark, we observe that both LC and WR of our fine-tuned models surpass those fine-tuned with baseline instruction datasets, regardless of the choice of the baseline model. This indicates that the datasets generated by MAGPIE are of higher quality, leading to significantly enhanced instruction-following capabilities. A similar observation is made when using the Arena-Hard evaluation benchmark. We highlight that the Llama-3 models fine-tuned with datasets generated by MAGPIE outperform even those models that have undergone preference optimization

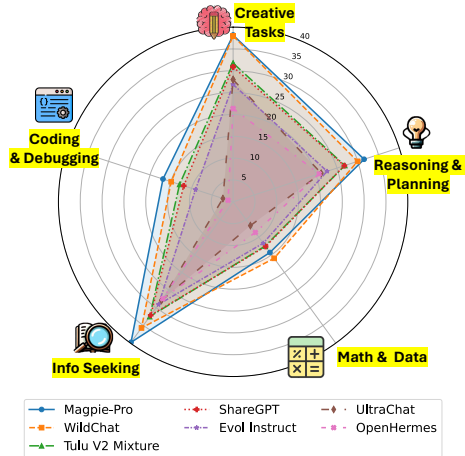


Figure 6: This figure shows the performance breakdown by category of MAGPIE-Pro and baselines on WildBench.

<sup>4</sup><https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct>

Table 3: This table compares the performance of models instruction-tuned on the Qwen base models using the MAGPIE-Pro-300K-Filtered dataset and the official instruction-tuned models. The Qwen base model enhanced with MAGPIE consistently outperforms the official instruction-tuned model.

Alignment Setup		AlpacaEval 2					
		GPT-4-Turbo (1106)			Official Aligned Model as Ref.		
		LC (%)	WR (%)	SD	LC (%)	WR (%)	SD
Qwen1.5-4B	Qwen1.5-4B-Chat	5.89	4.74	0.67	50	50	-
	Base Model + MAGPIE	<b>9.1</b>	<b>10.96</b>	0.93	<b>68.09</b>	<b>72.42</b>	1.42
Qwen1.5-7B	Qwen1.5-7B-Chat	14.75	11.77	0.97	50	50	-
	Base Model + MAGPIE	<b>15.10</b>	<b>18.51</b>	1.14	46.28	<b>58.53</b>	1.44

(e.g., instruction tuning combined with DPO), which emphasizes the high quality of data generated by MAGPIE.

To investigate the advantages of MAGPIE across different task categories, we also compare the performance of models fine-tuned with MAGPIE-Pro compared with baseline datasets using Wild-Bench benchmark [34]. This benchmark consists of 1024 tasks carefully selected from real-world human-LLM conversation logs. The results are demonstrated in Figure 6. We observe that MAGPIE consistently outperforms baseline datasets across categories.

**Models fine-tuned with data generated by MAGPIE achieve comparable performance to the official aligned model, but with fewer data.** In Table 2, we compare the performance of models fine-tuned with data generated by MAGPIE against the official aligned model (Llama-3-8B-Instruct). We observe that the Llama-3-8B base model fine-tuned with data from MAGPIE outperforms Llama-3-8B-instruct using the AlpacaEval 2 benchmark. For example, using the MAGPIE-Pro-300K-Filtered dataset to fine-tune the Llama-3-8B base model results in WC 29.47% against GPT-4-Turbo (1106). Furthermore, when Llama-3-8B-Instruct is chosen as the baseline model of AlpacaEval 2, we observe that WC of Llama-3-8B base models fine-tuned with data from MAGPIE exceeds 50%, indicating a preference for our fine-tuned models over the official aligned model. Finally, we highlight that our fine-tuning process uses no more than 300K data, whereas the official aligned models are fine-tuned with more than 10M data samples. This demonstrates the high quality of the data generated by MAGPIE. Using the Arena-Hard benchmark, we observe that a 1.7% difference between the WR achieved using our fine-tuned model and the official aligned model. We attribute this discrepancy to the fraction of coding-related instructions in our dataset. We believe that this gap could be easily bridged as we increase the size of datasets.

**Both data quantity and quality matter to capabilities of instruction-following.** In what follows, we compare within the family of datasets generated by MAGPIE in Table 2. These datasets differ in sizes, deployment of filtering, and models used to generate data. We observe that as the size of dataset increases, the performance of fine-tuned model improves, indicating that data quantity plays a critical role in enhancing instruction-following capabilities. Furthermore, the model fine-tuned with MAGPIE-Pro-300K-Filtered outperform those fine-tuned with the same amount of raw data. This demonstrates the effectiveness of our filtering technique, and underscores the importance of data quality. Finally, we observe that the models fine-tuned with MAGPIE-Pro consistently outperform those fine-tuned with MAGPIE-Air. The reason is that MAGPIE-Pro is generated by the more capable model, i.e., Llama-3-70B-Instruct.

**MAGPIE can enhance the performance of other backbone models.** Table 3 illustrates the efficacy of MAGPIE when applied to generate instruction dataset and fine-tune other backbone models, i.e., Qwen1.5-4B and Qwen1.5-7B. The results demonstrate that our fine-tuned models achieve better performance than the official aligned models, which have undergone instruction and preference tuning. These results underscore the effectiveness of MAGPIE and the quality of its generated instructions.

**Additional Experimental Results.** We defer additional experimental results and analysis of MAGPIE-Air-MT and MAGPIE-Pro-MT to Appendix E.1. Additionally, the performance of MAGPIE across various other benchmarks is reported in Appendix E.3.

## 5 Related Work

**LLM Alignment.** Instruction tuning [56] and preference tuning [5] are widely used to align the responses of LLMs with human values. Instruction tuning utilizes an instruction dataset to fine-tune



LLMs, where each instruction data consists of one turn or multiple turns of instructions and desired responses. The performance of instruction tuning heavily relies on the quality of instruction data [47, 53, 67]. Preference tuning further improves responses of LLMs using reinforcement learning human feedback (RLHF) [5] or preference optimization [2, 19, 23, 44] based on a preference dataset.

**Alignment Dataset Construction.** We classify the existing methods of creating datasets for model alignment into two main categories: human interactions with LLMs and synthetic instruction generation. To create datasets for alignment, previous studies have collected **human** interactions with LLMs [14, 64, 65, 66, 26]. However, manually crafting instructions is not only time-consuming and labor-intensive, but may also incorporate toxic content [64]. Another category of approaches [53, 47, 58, 59, 55, 46] focus on prompting LLMs to generate **synthetic** instruction datasets, beginning with a small set of human-annotated seed instructions and expanding these through few-shot prompting. However, these methods face a diversity challenge, as few-shot prompting often results in new instructions that are too similar to the original seed questions [31]. To enhance coverage, some research [16, 31] summarizes world knowledge and employs it to generate synthetic datasets. We note that our MAGPIE dataset also belongs to the synthetic dataset. However, we leverage the prompt template with no requirement for seed questions or prompt engineering.

Compared to the above two main categories, alignment data can also be generated by **transforming** existing data [54, 45, 20]. However, the constrained variety of NLP tasks in these datasets may impede the ability of tuned LLMs to generalize in real-world scenarios [31]. There are also **mixture** datasets (e.g., [24, 49, 38, 67]) that combine or select high-quality instruction data from various existing open-source instruction datasets to enhance coverage [24, 49] and/or improve overall performance [38, 67]. There are also data construction methods focusing on improving the reasoning and math abilities [61, 62], which can be further merged with MAGPIE for creating a better mixture of data for instruction tuning.

**Training Data Extraction.** Language models have the capability to memorize examples from their training datasets, potentially enabling malicious users to extract private information [8, 7, 9]. Pioneering work [27, 9, 41] has demonstrated that it is possible to extract private pre-training data from BERT [15], GPT-2 [43], and ChatGPT [1], respectively. Yu et al. [60] propose several tricks including adjusting sampling strategies to better extract training datasets from language models. Recently, Kassem et al. [25] propose a black-box prompt optimization method that uses an attacker LLM to extract high levels of memorization in a victim LLM. Wang et al. [52] leverage membership inference attack (MIA) to extract fine-tuning datasets from fine-tuned language models. Bai et al. [4] extracts the training dataset of production language models via special characters (e.g., structural symbols of JSON files, and , # in emails and online posts). Different from the prior work, we aim to create publicly available alignment datasets with minimal human effort by leveraging the remarkable generation capabilities of LLMs, rather than extracting private training data from LLMs.

## 6 Limitations and Ethical Considerations

**Limitations.** In certain scenarios, users may aim to fine-tune LLMs using domain-specific instruction data. Investigating how to configure MAGPIE to efficiently generate the desired domain-specific instructions (e.g., math problems) is subject to our future work. Also, there is still a gap between Magpie-tuned LLMs and official Llama-3-Instruct on datasets such as WildBench and MMLU, which suggest that we should focus on producing harder reasoning tasks and feedback learning data.

**License and Legality.** The instruction datasets generated by MAGPIE in this paper are subject to CC BY-NC license and Meta Llama 3 Community license. While users are permitted to distribute, adapt, and further develop our method MAGPIE, it is the responsibility of the users to apply MAGPIE to LLMs in compliance with the associated license agreement. We hereby disclaim any liability for misuse of data generated by users of MAGPIE.

**Societal Impact and Potential Harmful Consequences.** The primary objective of this paper is to develop a scalable method to synthesize instruction data to enhance the instruction-following capabilities of LLMs, and thus align them with human values. However, the data generated by MAGPIE may contain harmful instructions and/or responses, which may lead to unsafe behaviors if used raw in instruction tuning. Our empirical evaluations indicate that such harmful data instances constitute less than 1% of the dataset. To mitigate this risk, we develop a filtering technique in Appendix B to identify and remove these instances.

## 7 Conclusion

In this paper, we developed a scalable method, MAGPIE, to synthesize instruction data for fine-tuning large language models. MAGPIE leveraged the predefined instruction templates of open-weight LLMs and crafted a prompt specifying only the role of instruction provider. Given the crafted prompt, the LLM then generated detailed instructions due to their auto-regressive nature. MAGPIE then sent the generated instructions to the LLM to generate corresponding responses. These pairs of instructions and responses constituted the instruction dataset. We used Llama-3-8B-instruct to label the instruction dataset and developed a filtering technique to select effective data instances for instruction tuning. We fine-tuned the Llama-3-8B base model using the selected data, and demonstrated that the fine-tuned model outperformed those fine-tuned using all baselines. Moreover, our fine-tuned models outperformed the official aligned model, Llama-3-8B-Instruct, which has been instruction-tuned and preference-optimized using more than 10M data instances. This highlighted the quality of the instruction data synthesized by MAGPIE.

## 8 Acknowledgement

The research of Z. Xu, F. Jiang, L. Niu, and R. Poovendran is partially supported by the National Science Foundation (NSF) AI Institute for Agent-based Cyber Threat Intelligence and Operation (ACTION) under grant IIS 2229876. The research of Y. Choi is partially supported by the National Science Foundation (NSF) under grant DMS-2134012 (Scaling Laws of Deep Learning) and the Office of Naval Research (ONR) under grant N00014-24-1-2207 (Symbolic Knowledge Distillation of LLMs for All: Diverse Scales, Skills, and Values).

This work is supported in part by funds provided by the National Science Foundation, Department of Homeland Security, and IBM. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the NSF or its federal agency and industry partners.

## References

- [1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.
- [2] Mohammad Gheshlaghi Azar, Zhaohan Daniel Guo, Bilal Piot, Remi Munos, Mark Rowland, Michal Valko, and Daniele Calandriello. A general theoretical paradigm to understand learning from human preferences. In *International Conference on Artificial Intelligence and Statistics*, pages 4447–4455. PMLR, 2024.
- [3] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.
- [4] Yang Bai, Ge Pei, Jindong Gu, Yong Yang, and Xingjun Ma. Special characters attack: Toward scalable training data extraction from large language models. *arXiv preprint arXiv:2405.05990*, 2024.
- [5] Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, Ben Mann, and Jared Kaplan. Training a helpful and harmless assistant with reinforcement learning from human feedback, 2022.

- [6] Edward Beeching, Clémentine Fourier, Nathan Habib, Sheon Han, Nathan Lambert, Nazneen Rajani, Omar Sanseviero, Lewis Tunstall, and Thomas Wolf. Open llm leaderboard. [https://huggingface.co/spaces/open-llm-leaderboard/open\\_llm\\_leaderboard](https://huggingface.co/spaces/open-llm-leaderboard/open_llm_leaderboard), 2023.
- [7] Stella Biderman, Usvsn Prashanth, Lintang Sutawika, Hailey Schoelkopf, Quentin Anthony, Shivanshu Purohit, and Edward Raff. Emergent and predictable memorization in large language models. *Advances in Neural Information Processing Systems*, 36, 2023.
- [8] Hannah Brown, Katherine Lee, Fatemehsadat Mireshghallah, Reza Shokri, and Florian Tramèr. What does it mean for a language model to preserve privacy? In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 2280–2292, 2022.
- [9] Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650, 2021.
- [10] 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. Vicuna: An open-source chatbot impressing gpt-4 with 90%\* chatgpt quality, March 2023.
- [11] Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.
- [12] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.
- [13] Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Wei Zhu, Yuan Ni, Guotong Xie, Zhiyuan Liu, and Maosong Sun. Ultrafeedback: Boosting language models with high-quality feedback. *arXiv preprint arXiv:2310.01377*, 2023.
- [14] Databricks. Databricks dolly-15k, 2023.
- [15] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
- [16] Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Zhi Zheng, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. Enhancing chat language models by scaling high-quality instructional conversations. *arXiv preprint arXiv:2305.14233*, 2023.
- [17] Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. The faiss library. 2024.
- [18] Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B Hashimoto. Length-controlled alpacaEval: A simple way to debias automatic evaluators. *arXiv preprint arXiv:2404.04475*, 2024.
- [19] Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. Kto: Model alignment as prospect theoretic optimization. *arXiv preprint arXiv:2402.01306*, 2024.
- [20] Saumya Gandhi, Ritu Gala, Vijay Viswanathan, Tongshuang Wu, and Graham Neubig. Better synthetic data by retrieving and transforming existing datasets. *arXiv preprint arXiv:2404.14361*, 2024.
- [21] Aryo Pradipta Gema, Joshua Ong Jun Leang, Giwon Hong, Alessio Devoto, Alberto Carlo Maria Mancino, Rohit Saxena, Xuanli He, Yu Zhao, Xiaotang Du, Mohammad Reza Ghasemi Madani, et al. Are we done with mmlu? *arXiv preprint arXiv:2406.04127*, 2024.
- [22] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

- [23] Jiwoo Hong, Noah Lee, and James Thorne. Reference-free monolithic preference optimization with odds ratio. *arXiv preprint arXiv:2403.07691*, 2024.
- [24] Hamish Ivison, Yizhong Wang, Valentina Pyatkin, Nathan Lambert, Matthew Peters, Pradeep Dasigi, Joel Jang, David Wadden, Noah A Smith, Iz Beltagy, et al. Camels in a changing climate: Enhancing lm adaptation with tulu 2. *arXiv preprint arXiv:2311.10702*, 2023.
- [25] Aly M Kassem, Omar Mahmoud, Niloofar Mireshghallah, Hyunwoo Kim, Yulia Tsvetkov, Yejin Choi, Sherif Saad, and Santu Rana. Alpaca against vicuna: Using llms to uncover memorization of llms. *arXiv preprint arXiv:2403.04801*, 2024.
- [26] Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi Rui Tam, Keith Stevens, Abdullah Barhoum, Duc Nguyen, Oliver Stanley, Richárd Nagyfi, Shahul ES, Sameer Suri, David Glushkov, Arnav Dantuluri, Andrew Maguire, Christoph Schuhmann, Huu Nguyen, and Alexander Mattick. Openassistant conversations - democratizing large language model alignment. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 47669–47681. Curran Associates, Inc., 2023.
- [27] Kalpesh Krishna, Gaurav Singh Tomar, Ankur P. Parikh, Nicolas Papernot, and Mohit Iyyer. Thieves on sesame street! model extraction of bert-based apis. In *International Conference on Learning Representations*, 2020.
- [28] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.
- [29] Nathan Lambert, Valentina Pyatkin, Jacob Morrison, LJ Miranda, Bill Yuchen Lin, Khyathi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, et al. Rewardbench: Evaluating reward models for language modeling. *arXiv preprint arXiv:2403.13787*, 2024.
- [30] Hector Levesque, Ernest Davis, and Leora Morgenstern. The winograd schema challenge. In *Thirteenth international conference on the principles of knowledge representation and reasoning*, 2012.
- [31] Haoran Li, Qingxiu Dong, Zhengyang Tang, Chaojun Wang, Xingxing Zhang, Haoyang Huang, Shaohan Huang, Xiaolong Huang, Zeqiang Huang, Dongdong Zhang, et al. Synthetic data (almost) from scratch: Generalized instruction tuning for language models. *arXiv preprint arXiv:2402.13064*, 2024.
- [32] Tianle Li, Wei-Lin Chiang, Evan Frick, Lisa Dunlap, Banghua Zhu, Joseph E. Gonzalez, and Ion Stoica. From live data to high-quality benchmarks: The arena-hard pipeline, April 2024.
- [33] Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. AlpacaEval: An automatic evaluator of instruction-following models. [https://github.com/tatsu-lab/alpaca\\_eval](https://github.com/tatsu-lab/alpaca_eval), 2023.
- [34] Bill Yuchen Lin, Khyathi Chandu, Faeze Brahman, Yuntian Deng, Abhilasha Ravichander, Valentina Pyatkin, Ronan Le Bras, and Yejin Choi. Wildbench: Benchmarking language models with challenging tasks from real users in the wild, 2024.
- [35] Bill Yuchen Lin, Abhilasha Ravichander, Ximing Lu, Nouha Dziri, Melanie Sclar, Khyathi Chandu, Chandra Bhagavatula, and Yejin Choi. The unlocking spell on base llms: Rethinking alignment via in-context learning. *arXiv preprint arXiv:2312.01552*, 2023.
- [36] Stephanie Lin, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic human falsehoods. *arXiv preprint arXiv:2109.07958*, 2021.
- [37] Ruibo Liu, Jerry Wei, Fangyu Liu, Chenglei Si, Yanzhe Zhang, Jinneng Rao, Steven Zheng, Daiyi Peng, Diyi Yang, Denny Zhou, et al. Best practices and lessons learned on synthetic data for language models. *arXiv preprint arXiv:2404.07503*, 2024.

- [38] Wei Liu, Weihao Zeng, Keqing He, Yong Jiang, and Junxian He. What makes good data for alignment? a comprehensive study of automatic data selection in instruction tuning. In *The Twelfth International Conference on Learning Representations*, 2024.
- [39] Yu Meng, Mengzhou Xia, and Danqi Chen. Simpo: Simple preference optimization with a reference-free reward. *arXiv preprint arXiv:2405.14734*, 2024.
- [40] Meta. Llama 3. <https://ai.meta.com/blog/meta-llama-3/>, 2024.
- [41] Milad Nasr, Nicholas Carlini, Jonathan Hayase, Matthew Jagielski, A Feder Cooper, Daphne Ippolito, Christopher A Choquette-Choo, Eric Wallace, Florian Tramèr, and Katherine Lee. Scalable extraction of training data from (production) language models. *arXiv preprint arXiv:2311.17035*, 2023.
- [42] OpenAI. Tiktoken. <https://github.com/openai/tiktoken>, 2024.
- [43] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.
- [44] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.
- [45] Victor Sanh, Albert Webson, Colin Raffel, Stephen 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 Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M Rush. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*, 2022.
- [46] Zhiqing Sun, Yikang Shen, Qinhong Zhou, Hongxin Zhang, Zhenfang Chen, David Cox, Yiming Yang, and Chuang Gan. Principle-driven self-alignment of language models from scratch with minimal human supervision. *Advances in Neural Information Processing Systems*, 36, 2023.
- [47] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. [https://github.com/tatsu-lab/stanford\\_alpaca](https://github.com/tatsu-lab/stanford_alpaca), 2023.
- [48] Llama Team. Meta llama guard 2. [https://github.com/meta-llama/PurpleLlama/blob/main/Llama-Guard2/MODEL\\_CARD.md](https://github.com/meta-llama/PurpleLlama/blob/main/Llama-Guard2/MODEL_CARD.md), 2024.
- [49] Teknium. Openhermes dataset, 2023.
- [50] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruiti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.
- [51] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.
- [52] Jeffrey G Wang, Jason Wang, Marvin Li, and Seth Neel. Pandora’s white-box: Increased training data leakage in open llms. *arXiv preprint arXiv:2402.17012*, 2024.
- [53] Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13484–13508, Toronto, Canada, 2023. Association for Computational Linguistics.

- [54] 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. Super-NaturalInstructions: Generalization via declarative instructions on 1600+ NLP tasks. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5085–5109, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.
- [55] Zifeng Wang, Chun-Liang Li, Vincent Perot, Long T Le, Jin Miao, Zizhao Zhang, Chen-Yu Lee, and Tomas Pfister. Codeclm: Aligning language models with tailored synthetic data. *arXiv preprint arXiv:2404.05875*, 2024.
- [56] Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*, 2022.
- [57] Wei Xiong, Hanze Dong, Chenlu Ye, Ziqi Wang, Han Zhong, Heng Ji, Nan Jiang, and Tong Zhang. Iterative preference learning from human feedback: Bridging theory and practice for rlhf under kl-constraint, 2024.
- [58] Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. Wizardlm: Empowering large language models to follow complex instructions. *arXiv preprint arXiv:2304.12244*, 2023.
- [59] Canwen Xu, Daya Guo, Nan Duan, and Julian McAuley. Baize: An open-source chat model with parameter-efficient tuning on self-chat data. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6268–6278, Singapore, December 2023. Association for Computational Linguistics.
- [60] Weichen Yu, Tianyu Pang, Qian Liu, Chao Du, Bingyi Kang, Yan Huang, Min Lin, and Shuicheng Yan. Bag of tricks for training data extraction from language models. In *International Conference on Machine Learning*, pages 40306–40320. PMLR, 2023.
- [61] Xiang Yue, Xingwei Qu, Ge Zhang, Yao Fu, Wenhao Huang, Huan Sun, Yu Su, and Wenhua Chen. Mammoth: Building math generalist models through hybrid instruction tuning. *ArXiv*, abs/2309.05653, 2023.
- [62] Xiang Yue, Tuney Zheng, Ge Zhang, and Wenhua Chen. Mammoth2: Scaling instructions from the web, 2024.
- [63] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*, 2019.
- [64] Wenting Zhao, Xiang Ren, Jack Hessel, Claire Cardie, Yejin Choi, and Yuntian Deng. Wildchat: 1m chatGPT interaction logs in the wild. In *The Twelfth International Conference on Learning Representations*, 2024.
- [65] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Tianle Li, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zhuohan Li, Zi Lin, Eric Xing, Joseph E. Gonzalez, Ion Stoica, and Hao Zhang. LMSYS-chat-1m: A large-scale real-world LLM conversation dataset. In *The Twelfth International Conference on Learning Representations*, 2024.
- [66] 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. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023.
- [67] Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. Lima: Less is more for alignment. *Advances in Neural Information Processing Systems*, 36, 2023.

## A MAGPIE Extension

In this section, we explore the extension of MAGPIE. We first outline the process for constructing a multi-turn dataset (MAGPIE-MT). We then discuss methods for controlling instruction tasks using MAGPIE. Finally, we will briefly discuss how to develop a preference optimization dataset based on MAGPIE.

### A.1 Building Multi-Turn MAGPIE

To construct MAGPIE-MT, we initially follow Steps 1 and 2 to generate the first turn of instruction and response. For subsequent turns, we append the pre-query template to the end of the full prompt from the previous round of communication. We have observed that the model may occasionally forget its role as the user, especially for the 8B model. To mitigate this, we employ a system prompt designed to control the behavior of the LLM and reinforce its awareness of the multi-round conversation context. The full prompt for building the instructions of MAGPIE-MT can be found in Figure 11 in Appendix F. We follow the procedure described in Step 2 of Section 2 to generate responses and yield the multi-turn instruction dataset.

### A.2 Control Instruction Tasks of MAGPIE

In some scenarios, users may wish to fine-tune large language models (LLMs) using domain-specific instruction data, such as code or mathematical content, to enhance performance within specific domains. In this section, we introduce a lightweight and effective method to control the task category of generated instructions. Our approach involves guiding LLMs through the system prompt by specifying that it is a chatbot tailored for a particular domain and outlining the types of user queries it might encounter. We provide an example of a system prompt designed to control the generation of math-related instructions, as illustrated in Figure 12 within Appendix F.

### A.3 Building Preference Optimization Dataset with MAGPIE

MAGPIE can be readily adapted to create preference datasets by integrating responses generated by the instruct model with those from the base model. Specifically, utilizing the reward difference outlined in Section 3, a preference dataset can be assembled by designating the response from the instruct model as the preferred response, and the response from the base model as the less preferred one, provided that  $r^* - r_{base} > 0$ . We will soon open-source MAGPIE-PO, a preference optimization dataset to further align LLMs with human preferences.

## B Filter Setups

In this section, we explore potential filter configurations for selecting high-quality instructional data for fine-tuning purposes. We provide the following metrics to enable users to customize their filtered MAGPIE dataset:

1. **Input Length:** The total number of characters in the instructions.
2. **Output Length:** The total number of characters in the responses.
3. **Task Category:** The specific category of the instructions. See Appendix C.1 for details.
4. **Input Quality:** The clarity, specificity, and coherence of the instructions, rated as ‘very poor’, ‘poor’, ‘average’, ‘good’, and ‘excellent’.
5. **Input Difficulty:** The level of knowledge required to address the task described in the instruction, rated as ‘very easy’, ‘easy’, ‘medium’, ‘hard’, or ‘very hard’.
6. **Minimum Neighbor Distance:** The embedding distance to the nearest neighbor. Can be used for filtering out repetitive or similar instances.
7. **Reward:** Denoted as  $r^*$ . See Section 3 for details. This metric can be used to filter out low-quality responses, such as repetitions or refusals.
8. **Reward Difference:** Denoted as  $r^* - r_{base}$ . See Section 3 for details.

We provide several off-the-shelf configurations, as demonstrated in Table 4. We defer the detailed performance analysis of each filter configuration for MAGPIE-Pro to Appendix E.2.

Table 4: Different filter configurations we provide. We note that the Output Length filter is applied last. Specifically, this filter selects the  $k$  instances of the longest responses. In our experiments, we empirically set  $\tau_1 = -12$ , and  $\tau_2 = 0$ .

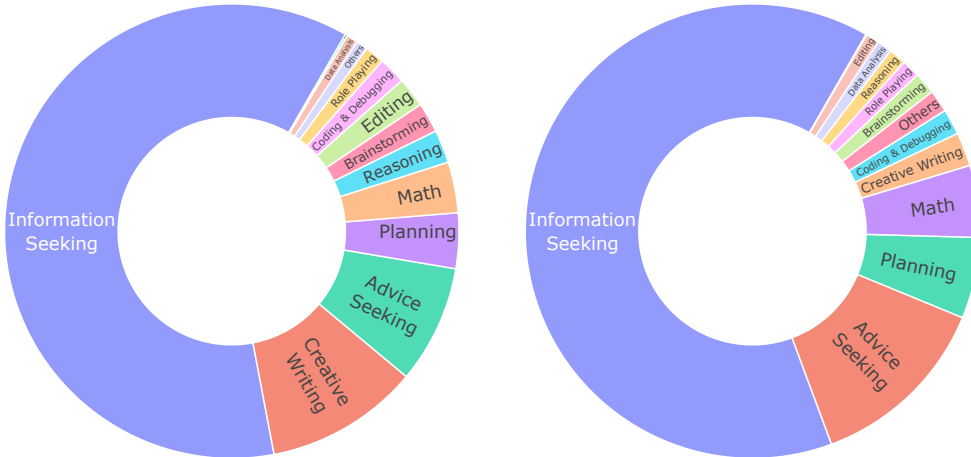
Source Dataset	Filter Name	#Convs	Input Length	Output Length	Task Category	Input Quality	Input Difficulty	Min Neighbor Distance	Reward	Reward Difference
MAGPIE-Air	Filter	300K	-	Longest	-	$\geq$ good	$\geq$ medium	$> 0$	-	$> \tau_2$
MAGPIE-Pro	Filter	300K	-	Longest	-	$\geq$ average	-	$> 0$	$> \tau_1$	-
	Filter2	300K	-	Longest	-	$\geq$ good	$\geq$ easy	$> 0$	$> \tau_1$	-
	Filter3	300K	-	Longest	-	-	-	$> 0$	$> \tau_1$	-
	Filter4	300K	-	Longest	-	$\geq$ good	$\geq$ easy	$> 0$	-	$> \tau_2$
	Filter5	338K	-	-	-	$\geq$ good	$\geq$ easy	$> 0$	$> \tau_1$	-
	Filter6	200K	-	Longest	-	-	50% easy + 50% $>$ easy	$> 0$	$> \tau_1$	-

## C More Dataset Analysis

This section provides additional dataset analysis, complementing the discussions in Section 3.

### C.1 Additional Analysis on Dataset Coverage and Attributes.

**Task Categories of MAGPIE-Pro and MAGPIE-Air.** Figure 7 illustrates the task category distributions for MAGPIE-Pro and MAGPIE-Air, as labeled by Llama-3-Instruct. We observe that the task category distributions of these two datasets are largely similar, however, MAGPIE-Pro exhibits a higher percentage of creative writing tasks.



(a) Task categories of MAGPIE-Pro.

(b) Task categories of MAGPIE-Air.

Figure 7: This figure visualizes the task category of MAGPIE-Pro and MAGPIE-Air by topic tags.

**Visualization of Root Verbs and Their Direct Noun Objects.** Figure 8 visualizes the top common root verbs and their direct noun objects of MAGPIE-Air dataset. This indicates the diverse topic coverage of MAGPIE-Air.

### C.2 Additional Safety Analysis

Table 5 illustrates the percentage of different unsafe categories of MAGPIE-Air and MAGPIE-Pro, as labeled by Llama-Guard-2 [48]. We have two key observations. First, the proportion of data containing potentially harmful queries is minimal, with less than 1% for both datasets. Second, the majority of unsafe responses fall into the category of specialized advice, which includes responses that may offer specialized financial, medical, or legal advice, or suggest that dangerous activities or objects are safe.



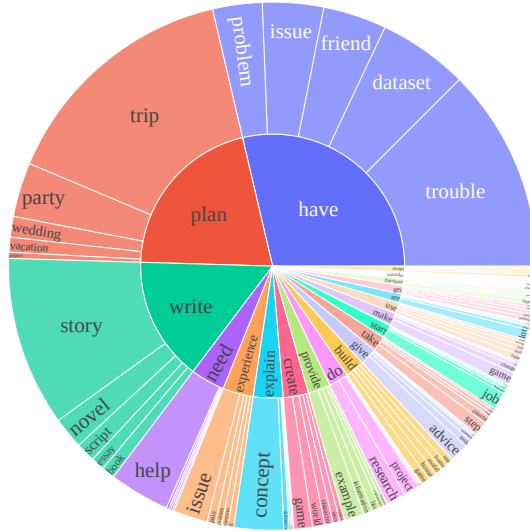


Figure 8: This figure demonstrates the top 20 most common root verbs (shown in the inner circle) and their top 5 direct noun objects (shown in the outer circle) within the MAGPIE-Air dataset. This indicates that MAGPIE encompasses a broad range of topics.

Table 5: This table shows the percentage of different unsafe categories of MAGPIE-Air and MAGPIE-Pro tagged by Llama-Guard-2 [48] model.

Dataset	Safe	Violent Crimes	Non-Violent Crimes	Sex-Related Crimes	Child Sexual Exploitation	Specialized Advice	Privacy	Intellectual Property	Indiscriminate Weapons	Hate	Suicide & Self-Harm	Sexual Content	Others
MAGPIE-Air	99.128%	0.001%	0.073%	0.003%	0.000%	0.636%	0.022%	0.026%	0.038%	0.001%	0.002%	0.009%	0.062%
MAGPIE-Pro	99.347%	0.001%	0.049%	0.002%	0.000%	0.446%	0.015%	0.074%	0.014%	0.001%	0.004%	0.011%	0.036%

### C.3 Ablation Analysis on Generation Configurations

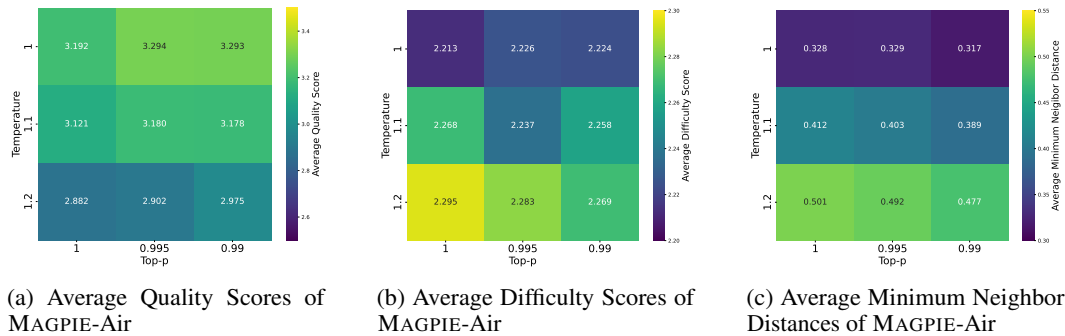


Figure 9: This figure illustrates the impact of varying decoding parameters on the quality, difficulty, and diversity of generated instructions. We observe that while higher temperature and top-p values may decrease the overall quality, they tend to increase both the difficulty and diversity of the instructions.

**Ablation Analysis on Decoding Parameters.** We conduct an ablation analysis on the decoding parameters used in generating instruction with MAGPIE. Specifically, we use three different temperatures (i.e., 1, 1.1, and 1.2) and top-p values (i.e., 1, 0.995, and 0.99) during Step 1 of MAGPIE. We use three metrics, **Average Quality Score**, **Average Difficulty Score** and **Average Minimum Neighbor Distance** to characterize the quality, difficulty, and diversity of instructions using different decoding parameters. The Average Quality Score is calculated by averaging the ratings of all data within a specific temperature-top-p pair, on a scale from 1 (‘very poor’) to 5 (‘excellent’). Similarly, the Average Difficulty Score is rated on a scale from 1 (‘very easy’) to 5 (‘very hard’). The Average

Minimum Neighbor Distance is calculated by averaging the minimum neighbor distances, as defined in Section 3, for all data generated using the same decoding parameters.

The findings are summarized in Figure 9. We observe that higher temperature and top-p values may slightly decrease the overall quality of instructions, while simultaneously increasing the difficulty and remarkably enhancing the diversity of the instructions generated. The selection of these hyper-parameters should be tailored to the user’s specific requirements, balancing the trade-offs between quality, difficulty, and diversity.

**Ablation Analysis on the System Prompt.** Figure 10 compares the use of system prompt compared with not using it in Step 1 of MAGPIE. Since the Llama-3 model does not have an official system prompt, we use the default system prompt from Vicuna [10]: A chat between a curious user and an artificial intelligence assistant. The assistant gives helpful, detailed, and polite answers to the user’s questions. We observe that using a system prompt generally results in a decrease in the overall quality of instructions, and the instructions are easier. Consequently, we recommend not appending system prompts in default settings.

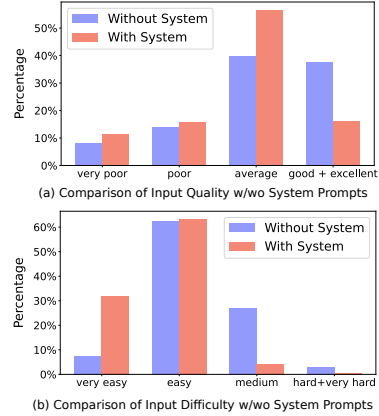


Figure 10: This figure compares the input quality and difficulty with and without system prompts.

## D Detailed Experimental Setups

### D.1 Experimental Setups for Generating MAGPIE-Air and MAGPIE-Pro

As detailed in Appendix C.3, varying decoding parameters in Step 1 can significantly influence the quality, difficulty, and diversity of the generated instructions. To optimize the trade-offs among these attributes, we employ diverse decoding parameters for the generation of MAGPIE-Air and MAGPIE-Pro. Table 6 presents the configurations of MAGPIE-Air and MAGPIE-Pro, showcasing how diverse decoding parameters shape each dataset.

We employ greedy decoding to generate responses in Step 2 for MAGPIE-Air and MAGPIE-Pro. The intuition is that the word with the highest probability is more likely to originate from the model’s training dataset.

Table 6: This table demonstrates the configurations of generating instructions of MAGPIE-Air and MAGPIE-Pro datasets with varying decoding parameters.

Dataset	Decoding Parameters			Total #Convs
	Temperature	Top-p	#Convs	
MAGPIE-Air	1.0	1.00	300K	3M
	1.0	0.995	300K	
	1.0	0.990	300K	
	1.1	1.00	300K	
	1.1	0.995	300K	
	1.1	0.990	300K	
	1.2	1.00	300K	
	1.2	0.995	300K	
	1.2	0.990	300K	
	1.25	1.00	100K	
	1.25	0.995	100K	
1.25	0.990	100K		
MAGPIE-Pro	1.0	1.00	300K	1M
	1.1	0.995	300K	
	1.2	0.995	300K	
	1.25	0.990	100K	

## D.2 Experimental Setups for Instruction Tuning

**Supervised Fine-Tuning Hyper-parameters.** Table 7 demonstrates the detailed supervised fine-tuning hyper-parameters. These experiments were conducted using Axolotl<sup>5</sup>.

Table 7: This table shows the hyper-parameters for supervised fine-tuning.

Hyper-parameter	Value
Learning Rate	$2 \times 10^{-5}$
Number of Epochs	2
Number of Devices	4
Per-device Batch Size	1
Gradient Accumulation Steps	8
Effective Batch Size	32
Optimizer	Adamw with $\beta s = (0.9, 0.999)$ and $\epsilon = 10^{-8}$
Learning Rate Scheduler	cosine
Warmup Steps	100
Max Sequence Length	8192

**Decoding parameters for evaluation benchmarks.** For Arena-Hard [32] and WildBench [34], we follow its default setting and use greedy decoding for all settings. For AlpacaEval 2 [33] which allows the model provider to specify decoding parameters, we also employ greedy decoding in all experiments with a slightly increased repetition penalty ( $RP = 1.2$ ) to mitigate the potential repetitive outputs during the generation.

## E Additional Experimental Results

### E.1 Performance of MAGPIE-MT

Table 8 compares the performance of MAGPIE-Air-MT and MAGPIE-Pro-MT with their respective single-turn counterparts. We observe that the multi-turn datasets have enhanced performance, particularly in the Arena-Hard benchmark.

Table 8: This table compares the performance of the multi-turn versions, MAGPIE-Air-MT and MAGPIE-Pro-MT, with their single-turn counterparts. All models are instruction-tuned on the Llama-8B base models.

Dataset		AlpacaEval 2						Arena-Hard
		GPT-4-Turbo (1106)			Llama-3-8B-Instruct			WR (%)
		LC (%)	WR (%)	SD	LC (%)	WR (%)	SD	
MAGPIE-Air	Single-Turn	22.66	23.99	1.24	49.27	50.80	1.44	14.9
	MT	<b>22.98</b>	<b>24.02</b>	1.27	<b>49.63</b>	<b>51.42</b>	1.40	<b>15.5</b>
MAGPIE-Pro	Single-Turn	<b>25.15</b>	<b>26.50</b>	1.30	50.52	52.98	1.43	18.9
	MT	24.21	25.19	1.28	<b>52.92</b>	<b>54.80</b>	1.41	<b>20.4</b>

### E.2 Ablation Analysis on Filter Designs

We conduct an ablation analysis on various filter designs within MAGPIE-Pro to assess their impact on the performance of supervised fine-tuned models. The results are presented in Table 9. We observe that different filtering strategies yield optimal performance on different benchmarks, and no single filter consistently achieves the best performance across all benchmarks. Therefore, determining how to select instructional data to enhance the performance in supervised fine-tuning is an interesting topic for future research.

<sup>5</sup><https://github.com/OpenAccess-AI-Collective/axolotl>

Table 9: This table compares the performance of different filter designs within MAGPIE-Pro. All models are instruction-tuned on the Llama-8B base models.

Dataset and Filter		AlpacaEval 2						Arena-Hard
		GPT-4-Turbo (1106)			Llama-3-8B-Instruct			WR (%)
		LC (%)	WR (%)	SD	LC (%)	WR (%)	SD	
MAGPIE-Pro	Filter	25.08	<b>29.47</b>	1.35	52.12	53.43	1.44	<b>18.9</b>
	Filter 2	<b>25.15</b>	26.50	1.30	50.52	52.98	1.43	<b>18.9</b>
	Filter 3	23.90	25.21	1.25	51.45	53.64	1.41	18.3
	Filter 4	24.20	25.33	1.27	<b>52.43</b>	54.34	1.43	17.9
	Filter 5	24.85	25.12	1.26	52.12	53.43	1.44	18.4
	Filter 6	23.20	28.43	1.26	51.34	<b>57.29</b>	1.41	17.9

### E.3 Performance of MAGPIE on More Benchmarks

We report the performance of models fine-tuned using MAGPIE-Air and MAGPIE-Pro, evaluated across a range of tasks featured on the Huggingface Open LLM Leaderboard [6] in Table 10. The tasks includes MMLU [22], ARC [11], HellaSwag [63], TruthfulQA [36], Winograd [30], and GSM8K [12]. We also perform experiments on MMLU-Redux [21] with zero-shot prompting. We use the default greedy decoding with  $RP = 1$  for all setups. Our experimental results demonstrate that models fine-tuned with MAGPIE-Air and MAGPIE-Pro achieve comparable performance to the official instruct model and other baselines despite the alignment tax.

Table 10: This table compares the performance of models instruction-tuned on MAGPIE-Air and MAGPIE-Pro against baselines and official instruct model across various downstream benchmarks. All models are instruction-tuned on the Llama-8B base models.

Alignment Setup	MMLU (5)	ARC (25)	HellaSwag (10)	TruthfulQA (0)	Winograd (5)	GSM8K (5)	MMLU-Redux (0)
ShareGPT	66.03	58.45	81.50	52.34	74.03	48.67	50.68
Evol Instruct	65.62	60.75	82.70	52.87	76.16	42.91	52.73
OpenHermes	65.42	62.29	82.15	50.85	75.61	47.16	46.07
Tulu V2 Mix	66.34	59.22	82.80	47.99	76.16	58.07	46.97
WildChat	65.95	59.22	81.39	53.18	75.30	48.75	52.59
UltraChat	65.23	62.12	81.68	52.76	75.53	50.57	50.75
MAGPIE-Air-300K-Filtered	64.45	61.01	79.90	53.48	72.38	52.24	52.34
MAGPIE-Pro-100K-Filtered	65.31	60.32	81.18	51.11	73.32	50.42	52.56
MAGPIE-Pro-200K-Filtered	64.98	61.26	80.71	51.82	73.16	47.76	51.44
MAGPIE-Pro-300K-Filtered	64.25	60.41	80.52	52.46	73.32	47.92	52.16
Llama-3-8B-Instruct	67.82	61.52	78.67	52.47	72.14	71.72	58.60

## F Prompt Templates

### F.1 Prompt Templates for MAGPIE Extension

This section presents the prompt template used to generate MAGPIE-MT and control instruction tasks, as detailed in Figure 11 and Figure 12, respectively.

### F.2 Prompt Templates for Evaluation

Here, we present the prompt template employed to generate task categories, quality, and difficulty, as detailed in Figure 13, Figure 14, and Figure 15, respectively. The placeholder input represents the instructions to be evaluated.

Prompt for generating MAGPIE-MT

```
<|begin_of_text|><|start_header_id|>system<|end_header_id|>  
  
You are a helpful AI assistant. The user will engage in a multi-round conversation with you,  
asking initial questions and following up with additional related questions. Your goal is  
to provide thorough, relevant and insightful responses to help the user with their  
queries.<|eot_id|><|start_header_id|>user<|end_header_id|>  
  
{instruction}<|eot_id|><|start_header_id|>assistant<|end_header_id|>  
  
{response}<|eot_id|><|start_header_id|>user<|end_header_id|>
```

Figure 11: Prompt for generating MAGPIE-MT. The placeholder {instruction} and {response} are from the first turn.

Prompt for controlling instruction tasks

```
<|begin_of_text|><|start_header_id|>system<|end_header_id|>  
  
You are an AI assistant designed to provide helpful, step-by-step guidance on solving math  
problems. The user will ask you a wide range of complex mathematical questions. Your  
purpose is to assist users in understanding mathematical concepts, working through  
equations, and arriving at the correct solutions.<|eot_id|><|start_header_id|>user<|  
end_header_id|>
```

Figure 12: Prompt for controlling instruction tasks. In this example, we control LLMs to generate instructions related to math.

Prompt for generating task categories

```
# Instruction
Please label the task tags for the user query.

## User Query
“{input}”

## Tagging the user input
Please label the task tags for the user query. You will need to analyze the user query and
select the most relevant task tag from the list below.

all_task_tags = [
  "Information seeking", # Users ask for specific information or facts about various topics.
  "Reasoning", # Queries require logical thinking, problem-solving, or processing of
    complex ideas.
  "Planning", # Users need assistance in creating plans or strategies for activities and
    projects.
  "Editing", # Involves editing, rephrasing, proofreading, or other tasks related to the
    composition of general written content.
  "Coding & Debugging", # Users seek help with writing, reviewing, or fixing code in
    programming.
  "Math", # Queries related to mathematical concepts, problems, and calculations.
  "Role playing", # Users engage in scenarios requiring ChatGPT to adopt a character or
    persona.
  "Data analysis", # Requests involve interpreting data, statistics, or performing analytical
    tasks.
  "Creative writing", # Users seek assistance with crafting stories, poems, or other
    creative texts.
  "Advice seeking", # Users ask for recommendations or guidance on various personal or
    professional issues.
  "Brainstorming", # Involves generating ideas, creative thinking, or exploring possibilities.
  "Others" # Any queries that do not fit into the above categories or are of a miscellaneous
    nature.
]

## Output Format:
Note that you can only select a single primary tag. Other applicable tags can be added to
the list of other tags.
Now, please output your tags below in a json format by filling in the placeholders in <...>:
““
{{
  "primary_tag": "<primary tag>",
  "other_tags": ["<tag 1>", "<tag 2>", ... ]
}}
““
```

Figure 13: Prompt for generating task categories

#### Prompt for generating quality of instructions

```
# Instruction
You need to rate the quality of the user query based on its clarity, specificity, and coherence.
The rating scale is as follows:

- very poor: The query is unclear, vague, or incoherent. It lacks essential information and
  context.
- poor: The query is somewhat unclear or lacks important details. It requires significant
  clarification.
- average: The query is moderately clear and specific. It may require some additional
  information for a complete understanding.
- good: The query is clear, specific, and mostly well-formed. It provides sufficient context for
  understanding the user's intent.
- excellent: The query is very clear, specific, and well-articulated. It contains all the
  necessary information and context for providing a comprehensive response.

## User Query
“{input}”

## Output Format
Given the user query, you first need to give an assessment, highlighting the strengths and/or
weaknesses of the user query. Then, you need to output a rating from very poor to
excellent by filling in the placeholders in [...]:

“
{
  "explanation": "[...]",
  "input_quality": "[very poor/poor/average/good/excellent]"
}
”
```

Figure 14: Prompt for generating quality of instructions

#### Prompt for generating difficulty of instructions

```
# Instruction
You first need to identify the given user intent and then label the difficulty level of the user
query based on the content of the user query.

## User Query
“{input}”

## Output Format
Given the user query, in your output, you first need to identify the user intent and the
knowledge needed to solve the task in the user query. Then, rate the difficulty level of
the user query as 'very easy', 'easy', 'medium', 'hard', or 'very hard'.

Now, please output the user intent and difficulty level below in a json format by filling in the
placeholders in [...]:

“
{
  "intent": "The user wants to [...]",
  "knowledge": "To solve this problem, the models need to know [...]",
  "difficulty": "[very easy/easy/medium/hard/very hard]"
}
”
```

Figure 15: Prompt for generating difficulty of instructions

## G MAGPIE Examples

In this section, we demonstrate several representative instructions from MAGPIE-Pro with different task categories.

### Information Seeking

A few days ago, I was at a restaurant and I got a cup of coffee. However, when I went to take a sip, I realized it was a little too hot. So, I decided to let it cool down for a few minutes. As I waited, I noticed that coffee is actually two different colors. The part that was closest to the surface of the coffee is a lighter color, and the part that is deeper is a darker color. Have you ever observed this phenomenon before?

### Reasoning

You have 3 switches but they are not labeled. Each switch corresponds to one of three light bulbs in a room. Each light bulb is either on or off. You can turn the switches on and off as many times as you want, but you can only enter the room one time to observe the bulbs. How can you figure out which switch corresponds to which light bulb?

### Planning

You are the Founder of a Financial Planning Company. As a professional financial advisor, you are scheduled to meet a new client tomorrow. Specifically, what are you planning to do to prepare for this meeting?

### Editing

What is the best way to re-write the sentence: "We call this the 'core' product or the 'core' offering" using proper quotation marks and avoiding the word "this"?

### Coding & Debugging

Write a Python program that calculates the total cost of a customer's order. The program should ask for the customer's name, the number of items they want to purchase, and the price of each item. It should then calculate the total cost by multiplying the number of items by the price of each item and adding 8% sales tax. The program should display the customer's name, the number of items, the price of each item, and the total cost, including sales tax.

### Math

In the following problem, please use integers to solve it. A water tank has 1000 L of water. On the first day,  $\frac{1}{5}$  of the water was drained. On the second day,  $\frac{3}{10}$  of the remaining water was drained. On the third day,  $\frac{2}{5}$  of the remaining water was drained. On the fourth day,  $\frac{3}{4}$  of the remaining water was drained. How many liters of water are left after the fourth day?

### Role Playing

In this game, you will be the host, and I will be the contestant. You will ask me a series of questions, and I will try to answer them correctly. The questions will be multiple choice, and I will have a 25% chance of getting the correct answer if I just randomly guess. However, I am a clever contestant, and I will try to use logic and reasoning to increase my chances of getting the correct answer.



#### Data Analysis

The personnel manager at a company is tasked with finding the average salary of new hires. She has collected data on the salaries of 13 new hires. She wants to know if there is a statistical difference between the average salary of new hires and the national average salary. The national average salary is \$45,000. The sample of new hires has a mean salary of \$42,800 and a standard deviation of \$4,200.

#### Creative Writing

Write a paragraph about a mythical creature that you created. The creature is small, no larger than a house cat. It has shimmering scales that reflect light, and can emit a soft, pulsing glow from its body. It has large, round eyes that seem to see right through you, but with a gentle kindness. It has a soft, melodious voice, and can communicate with humans through a form of telepathy.

#### Advice Seeking

How do you handle stress and overwhelm?

#### Brainstorming

Can you give me some ideas for a spontaneous, fun and memorable birthday celebration for my partner?

#### Others

What does "sdrawkcaB" mean?