Generation of Instruction and Preference Dataset for Improving Japanese Instruction Following in LLMs

Kei Moriyama[1,2], Takashi Kodama[2], Kouta Nakayama[2]
[1]The University of Tokyo, [2]NII LLMC
kei-moriyama@g.ecc.u-tokyo.ac.jp, {tkodama,nakayama}@nii.ac.jp

Abstract

Instruction following, the ability to generate text that aligns with human intent, is a core capability of large language models (LLMs) for real-world applications. Instruction tuning is widely used to obtain this capability, but it requires large amounts of annotated data. To reduce the labor and cost of large-scale annotation, data augmentation using LLMs has been proposed as a promising approach. As this approach has primarily been applied to English datasets, its effectiveness in other languages, such as Japanese, remains unclear. In this paper, we propose an automatic pipeline for generating instruction and preference datasets in Japanese. The instruction dataset is created by expanding a manually annotated dataset using an LLM. The preference dataset is then constructed by adding LLM-generated negative examples to the instruction dataset. To ensure the quality of the datasets, instructions and responses are evaluated using LLM-as-a-Judge and ROUGE-L. Experimental results using supervised fine-tuning and direct preference optimization demonstrate that these synthetic datasets improve the instruction-following capability in Japanese.

Keywords: Instruction Following, Self-Instruct, Instruction Tuning, Data Augmentation

1. Introduction

Large language models (LLMs) achieved strong performance across a range of NLP tasks (Brown et al., 2020; Touvron et al., 2023). This progress leads to the development of chat-based applications. A key requirement for this application is the zero-shot instruction-following, which generates responses aligned with the user's intent (Zhou et al., 2023; Dussolle et al., 2025).

To generate such responses, large-scale instruction and preference datasets are essential for training LLMs. Most existing datasets are created primarily in English (Taori et al., 2023; Mishra et al., 2022; Wang et al., 2022), and the construction of such datasets is labor-intensive and costly. As a result, it is difficult to develop high-quality datasets for non-English languages.

To alleviate the burden of annotation, data augmentation using LLMs has emerged as a promising approach (Lou et al., 2023; Honovich et al., 2023; Köksal et al., 2024; Xu et al., 2024). These methods expand manually created datasets using LLMs. Such approaches have been applied not only to instruction datasets but also preference datasets (Huang et al., 2023; Lee and Han, 2025). LLMs trained on the generated dataset show strong performance on various benchmarks. However, most datasets are generated in English. Instruction following is inherently subject to language-specific constraints; for instance, Japanese instructions specify script-based requirements, such as restricting or requiring the use of kanji. It therefore remains unclear whether these approaches can handle language-specific constraints that arise in non-English languages.

Although prior research proposes methods of generating instruction datasets in Japanese (Sun et al., 2024; Omura et al., 2024), there are still several gaps in dataset construction and evaluation. The previous approach (Sun et al., 2024) generates only instruction datasets, leaving the effectiveness of synthetic preference dataset unexplored. For the evaluation, it relies solely on LLM-based scoring. Therefore, the effectiveness of fine-tuning LLMs on these datasets remains unclear.

In this paper, we propose an automatic generation pipeline for the instruction and preference datasets. Our generation process begins with a human-created seed dataset. This seed dataset is constructed using instruction constraint categories defined in this paper. An LLM generates new instructions based on the instructions and constraint categories in the seed dataset. Generated instructions are filtered based on the evaluation scores of LLM-as-a-Judge (Zheng et al., 2023) and ROUGE-L (Lin, 2004). For each retained instruction, a corresponding response is generated and validated using LLM-as-a-Judge. To construct the preference dataset, we treat the validated response in the instruction dataset as the chosen response. The LLM then generates a rejected response for each instruction. Finally, the resulting preference pairs are filtered using LLM-as-a-Judge to ensure their overall quality. Table 1 presents examples of filtered and retained instruction-response pairs of instruction dataset.

We evaluated the effectiveness of our instruction and preference datasets by training various LLMs using supervised fine-tuning (SFT) and di-

Proceedings of the Fifteenth Language Resources and Evaluation Conference (LREC 2026), pages 1435–1454
11–16 May 2026.

Constraint Category	Reference Input	Generated Instruction	Generated Response
Character>Punctuation	ムーミン一家の家族 のメンバーを箇条書 きで答えて下さい。 (Please list the members of the Moomin family in a bulleted list.)	ムーミン一家の家族 のメンバーを、句読点 （カンマ）を使って列 挙してください。 (Please list the members of the Moomin family, separated by commas.)	ムーミン、 ムーミンママ、 ムーミンパパ、 スnufunfa (Moomin、 Moomin mother、 Moomin father、 sunufunfa)
Format>Table>csv	火星の周りには衛星 はありますか?あれば 名前と英語表記を教え て。回答は400字以内で。 (Does Mars have any moons? If so, please tell me their names and English spelling. Please keep the answer under 400 characters.)	火星の衛星の情 報をCSV形式で 教えてください。 衛星の名前と英語表 記を含めてください。 (Please provide information about the moons of Mars in CSV format. Include the name and English spelling of each satellite.)	"衛星名","英語表記" "フォボス","Phobos" "デイモス","Deimos" (“satellite name", “English spelling")

Table 1: Examples of kept and removed synthetic instruction-answer pairs of instruction dataset. The constraint category column indicates the constraint category which converted instruction should have. The above pair is removed because the answer does not satisfy the specified constraint and Japanese is collapse. The blue sentence shows the collapse part of Japanese. Red sentence in the instruction shows the specified constraint in the prompt.

rect preference optimization (DPO) (Rafailov et al., 2023). The experimental results show that synthetic instruction and preference datasets improve instruction-following ability in both English and Japanese. In addition, the performance on other benchmark tasks also improves. We analyzed the impact of our synthetic instruction dataset This analysis demonstrates that our synthetic dataset contributes to improving instruction-following ability. Overall, these results indicate that our dataset is effective not only for instruction-following but also for other downstream tasks.[1]

Our contributions are summarized as follows:

- We propose a method for generating instruction and preference datasets designed to improve the instruction-following performance of LLMs. To ensure the quality, the datasets are filtered by ROUGE-L and LLM-as-a-Judge.
- We fine-tune several LLMs using SFT and DPO with our synthetic datasets. The models are evaluated not only on instruction-following, but also on eleven NLP tasks.
- Experimental results demonstrate that our datasets improve the instruction-following performance, particularly in Japanese. Furthermore, the fine-tuned models show improved performance on other downstream tasks.

[1]Please check the following URL for the generated instruction and preference dataset. https://huggingface.co/datasets/llm-jp/synth-if.

2. Related Work

Instruction following is a core capability of LLMs for real-world applications, where the model generates responses that follow specified constraints in the instruction. Various datasets were constructed to develop and evaluate this ability (Mishra et al., 2022; Wang et al., 2022; Zhou et al., 2023; Dus; He et al., 2024; Xia et al., 2024; Wen et al., 2024; Ye et al., 2025). The construction of such datasets requires free-form annotation, which must account for the constraints. This process is therefore time-consuming and labor-intensive.

To reduce the burden and cost of annotation, data augmentation by LLMs has been proposed. Several studies construct instruction datasets by expanding human-created instructions (Xu et al., 2024; Lou et al., 2023; Honovich et al., 2023; Yin et al., 2023; Sun et al., 2024; Schick and Schütze, 2021; Köksal et al., 2024). LLMs trained on these synthetic datasets demonstrate improvements in various abilities, such as classification (Li et al., 2023), conversation (Xu et al., 2023), question answering (Schmidt et al., 2024), and commonsense reasoning (Yang et al., 2020). Self-instruct (Wang et al., 2023) improves performance on the SUPERNI benchmark (Wang et al., 2022) in a zero-shot setting. The effectiveness of the synthetic preference dataset is also shown (Lee and Han, 2025; Huang et al., 2023). AutoPM (Huang et al., 2023) demonstrates that the LLM can be aligned using the synthetic preference dataset in helpful-honest-harmless criteria. In addition to generating training datasets, LLMs have been used to generate benchmark datasets (Ye et al., 2025; Pan et al., 2023; Qin et al., 2024). ComplexBench (Wen et al.,

2024) proposes a method that used LLMs not only for dataset generation but also for evaluation. This method shows a better pairwise comparison rate with humans.

Most existing work is conducted in English; therefore, the effectiveness in non-English languages remains unclear. In Japanese, the existing work evaluates the effectiveness of synthetic datasets (Sun et al., 2024; Omura et al., 2024). However, their evaluation of trained LLMs is limited to LLM-based scoring and pairwise comparisons. In contrast to this prior work, we evaluate fine-tuned LLMs using various benchmark datasets. Furthermore, we evaluate the effectiveness of our synthetic preference dataset.

3. Method

3.1. Overview of Generation Pipeline

Our proposed method generates synthetic instruction and preference datasets using LLMs. Fig.1 presents the overall generation pipeline.

First, an LLM generates an instruction dataset based on human-created seed instructions. To generate new instructions, the LLM is provided with predefined instruction-constraint categories and their descriptions. The newly generated instructions are filtered for quality and diversity based on ROUGE-L (Lin, 2004) and LLM-as-a-Judge (Zheng et al., 2023) scores. For each instruction that passes the filtering stage, the LLM generates a corresponding response. The resulting instruction-response pairs are then evaluated by LLM-as-a-Judge. We treat the remaining instruction-response pairs as the instruction dataset.

The preference dataset is constructed from the generated instruction dataset. Instructions and their responses from the instruction dataset serve as the instructions and chosen responses in the preference dataset, respectively. The LLM generates rejected responses, and LLM-as-a-Judge filters them. The filtered instruction-chosen-rejected triples are used as the preference dataset.

3.2. Generation of Synthetic Instruction Dataset

We use two strategies for instruction generation.

Add Constraint The LLM introduces a new constraint into a given instruction. We refer to this strategy as **Add**.

Rewrite Instruction The LLM rewrites a given instruction to satisfy a specified constraint category. We refer to this strategy as **Rewrite**.

We design a base prompt for each strategy. The prompt for instruction generation is constructed by appending the following components to the base prompt: an instruction from the seed dataset, a constraint category, and its corresponding description.

To ensure the quality of the synthetic instructions, a filtering step is applied using ROUGE-L and LLM-as-a-Judge. This process discards an instruction if its ROUGE-L score exceeds a specified threshold or if the LLM-as-a-Judge score falls below a predefined threshold. ROUGE-L measures similarity to promote the diversity of synthetic instructions. The similarity of each newly generated instruction is computed against two references: (i) the instructions that have already passed the filtering (randomly sampled when the set is large), and (ii) a seed instruction included in the prompt.

LLM-as-a-Judge evaluates the semantic quality of synthetic instructions. The following metrics are used for evaluation:

Relevance This metric evaluates the relevance of the generated instruction to the specified constraint category.

Fluency This metric evaluates the linguistic fluency and grammatical correctness of the instruction in the target language.

Verbosity This metric evaluates whether the instruction is concise, avoids redundant expressions, and constitutes a complete instruction.

The evaluations are made on a 5-point Likert scale (1 indicates the lowest and 5 indicates the highest). The prompt includes few-shot examples.

For each instruction that passes the filtering stage, the LLM generates a corresponding response. The responses are evaluated using LLM-as-a-Judge and with the following metrics:

Adherence This metric evaluates whether the response adheres to all constraints specified in the instruction.

Fluency This metric evaluates the linguistic fluency and grammatical correctness of the response.

Verbosity This metric assesses whether the response is concise and free of redundant content.

Completeness This metric assesses whether the instruction-response pair is accurate and complete.

The rating scale and filtering criteria are the same as those used for the instruction evaluation. Instruction-response pairs that pass this filtering are used as the synthetic instruction dataset.

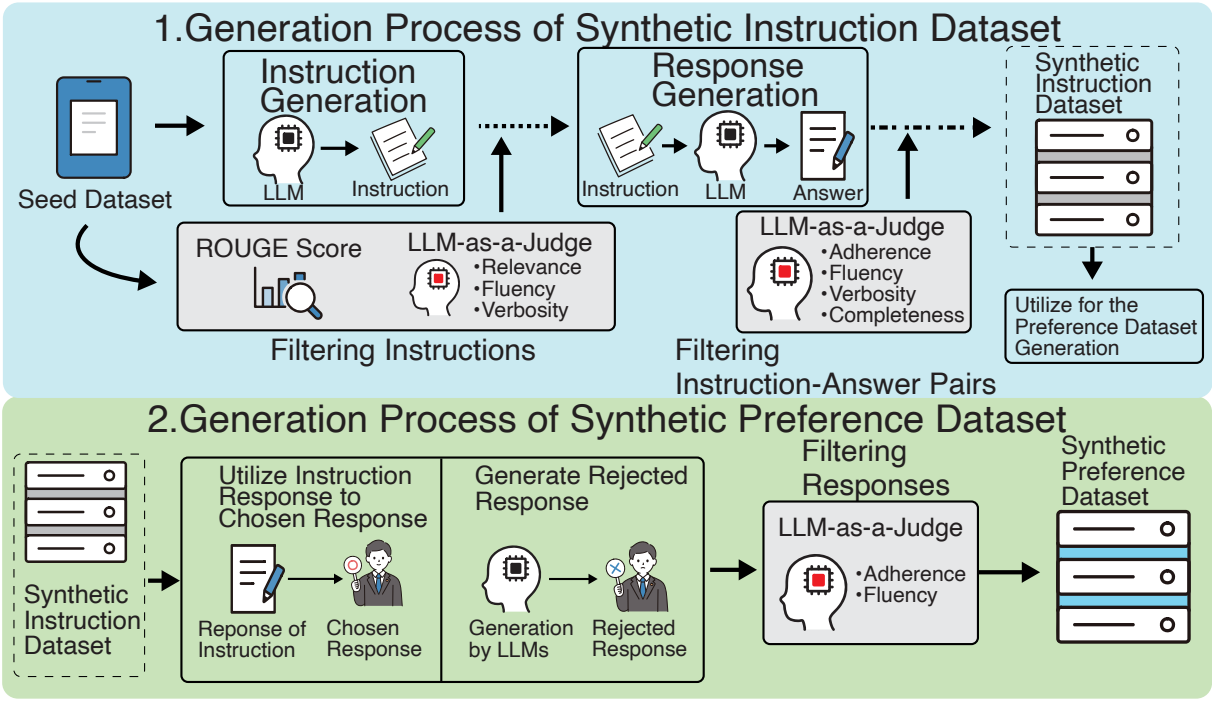

Figure 1: Overview of the our generation pipeline. Above square shows the process of the generation of the instruction dataset. The generation of the preference dataset is presented in below.

3.3. Generation of Synthetic Preference Dataset

We construct the synthetic preference dataset from the synthetic instruction dataset. The instruction and response are used as the instruction and chosen response in the synthetic preference dataset, respectively. The corresponding rejected responses are generated by the LLM.

We define two types of rejected responses based on two aspects of instruction violation. Each type is named after the instruction aspect that the response addresses.

Content A response of this type is content-relevant to the instruction but fails to adhere to the required output format.

Format A response of this type adheres to the specified format of the instruction, but its content is irrelevant to the instruction.

To generate each type, the LLM receives an instruction, examples of rejected responses, and their corresponding explanations. It then generates a rejected response.

The generated rejected responses are then evaluated by LLM-as-a-Judge on a 5-point Likert scale (1 indicates the lowest score and 5 means the highest). The evaluation is based on the following two criteria:

Adherence This metric evaluates whether the rejected response correctly reflects its intended violation type. For the **Content** type, the response should be topically relevant to the instruction but fail to follow the required output format. For the **Format** type, the response should adhere to the specified format while containing content that is irrelevant.

Fluency This metric evaluates the linguistic fluency and grammatical correctness of the rejected response.

The filtering criteria are the same as those used for instruction filtering.

4. Experiment

4.1. Details of the Dataset Generation

We use the ichikara-instruction-format dataset as the seed dataset for instruction generation. This dataset is an extension of the ichikara-instruction dataset (Sekine et al., 2024). To add constraints to instructions in ichikara-instruction, we defined an instruction-constraint category. The constraint categories and their corresponding descriptions used in the experiments are listed in Appendix A.

We utilize Qwen2.5-32B-Instruct[2] (Qwen et al., 2025) for the dataset generation and LLM-as-a-Judge. The maximum generation length is set to

[2] https://huggingface.co/Qwen/Qwen2.5-32B-Instruct

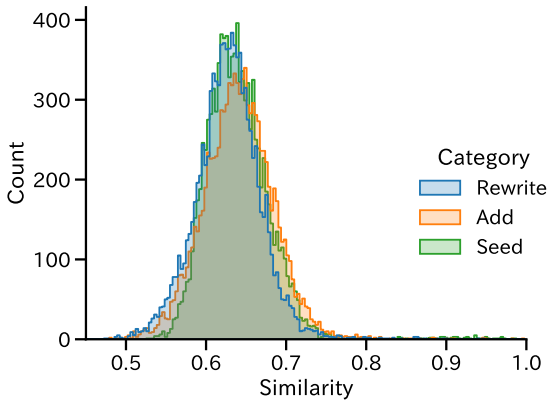

Figure 2: Similarity distribution of instructions obtained from 10,000 pairs in our synthetic dataset.

512 tokens for LLM-as-a-Judge and dataset generation. The temperature is set to 0.8 for dataset generation, and to 0.1 for the LLM-as-a-Judge. For all other hyperparameters, we use the default parameters of vLLM library (Kwon et al., 2023).

In the LLM-as-a-Judge filtering, a threshold score is set to 3 (on a 5-point Likert scale) for all filtering settings. Additionally, the ROUGE-L similarity threshold is set to 0.7.

4.2. Details of Synthetic Datasets

Table 1 shows examples of instruction-response pairs: removed and retained one. In both examples, LLM rewrote instructions to include a specified constraint. An instruction in the first row converted the reference instruction to the comma-separated list. However, the response was filtered out because it contains the unnatural Japanese word "スnufunfa". In the second example, LLM converted the instruction to include the csv format constraint for its output. The response was retained because it satisfied instruction constraint and was correct. Table 2 presents the dataset size, mean, standard deviation (std), and median token length of the generated instruction and preference datasets.

To visualize the diversity of the generated instructions, we plot the similarity distributions of seed and synthetic instructions in Fig.2. We sampled 10,000 instructions and measured pairwise similarity using BERTScore (Zhang et al., 2020). Fig.2 shows that the distribution is similar across datasets. The mean similarity is 0.639 for the seed dataset, 0.640 for the *Add* dataset, and 0.627 for the *Rewrite* dataset. These results indicate that the diversity of synthetic instructions is comparable to that of the seed dataset.

4.3. Instruction-Tuning Dataset and Models

We fine-tuned several LLMs to evaluate the effectiveness of synthetic instruction and preference datasets. We utilized Llama3.2-1B[3], Qwen2.5-1.5B[4], llm-jp-3-1.8B[5] and llm-jp-3-13B[6] for the SFT experiment. The following datasets are additionally used for SFT.

- AnswerCarefully (Suzuki et al., 2025)
- ichikara-instruction (Sekine et al., 2024)
- ichikara-instruction-format
- random-to-fixed-multiturn-Calm3 (Hatakeyama, 2024b)
- wizardlm8x22b-logical-math-coding-sft-ja[7]
- Daring-Anteater (Wang et al., 2024)
- AutoMultiTurnByCalm3-22B (Hatakeyama, 2024a)

For the preference dataset, we applied DPO (Rafailov et al., 2023) only to models in the llm-jp-3 series. Hyperparameters of SFT and DPO are provided in Appendix C.

In all experiments, we refer to *base* as the pre-trained model before fine-tuning. *SFT* refers to the model trained with SFT using only the datasets described above. *SFT+Synth* is a fine-tuned model using SFT on the datasets described above and our synthetic instruction dataset. *SFT+Synth+DPO* is obtained by applying DPO to the *SFT+Synth* model using our synthetic preference dataset.

4.4. Baseline Models

We use multilingual and Japanese LLMs as baselines. For multilingual models, we select instruction-tuned Qwen2.5 (Qwen et al., 2025) models with 7B and 14B and Llama3.2 (Touvron et al., 2023) models with 3B parameters. For the Japanese LLMs, we use OpenCalm3-22B (Ishigami, 2024), and the 13B base and 1B instruction-tuned versions of Sarashina.

[3] https://huggingface.co/meta-llama/Llama-3.2-1B

[4] https://huggingface.co/Qwen/Qwen2.5-1.5B

[5] https://huggingface.co/llm-jp/llm-jp-3-1.8b

[6] https://huggingface.co/llm-jp/llm-jp-3-13b

[7] https://huggingface.co/datasets/kanhatakeyama/wizardlm8x22b-logical-math-coding-sft-ja

Dataset	Type	Data Size	Category	Mean	Std	Median
Seed	-	332	Instruction	116.04	214.19	58.00
			Answer	163.61	157.33	117.50
Instruction	Add	11,068	Instruction	62.60	63.82	43.00
			Answer	97.23	99.78	59.00
	Rewrite	13,123	Instruction	107.19	82.45	81.00
			Answer	112.57	102.77	79.00
Preference	Content	409	Instruction	77.45	63.71	58.00
			Chosen	100.34	96.13	67.00
			Rejected	76.73	84.98	47.00
	Format	18,907	Instruction	85.55	76.02	60.00
			Chosen	100.34	96.13	67.00
			Rejected	108.59	86.39	85.00

Table 2: Summary of data size and token length in *Seed*, *Instruction* and *Preference* datasets.

4.5. Evaluation Metrics and Datasets

We evaluate the models using llm-jp-eval[8]. The details of the evaluation tasks are described below.

Instruction Following (IF) Instruction following assesses an LLM's ability to follow human instruction. We use Japanese and English datasets from M-IFEval (Dussolle et al., 2025).

Commonsense Reasoning (CR) This task evaluates the commonsense reasoning ability of LLMs. We use JCommonsenseMorality (Takeshita et al., 2023), JCommonsenseQA (Kurihara et al., 2022) and KUCI (Omura et al., 2023).

Entity Linking (EL) Entity linking is the task of identifying mentions of named entities in text and linking them to their corresponding entries in a knowledge base. The evaluation dataset is chABSA (Takahiro and Hiroki).

Fundamental Analysis (FA) This task evaluates the analytical ability across several linguistic tasks. This dataset contains reading prediction, named entity recognition, dependency parsing, predicate-argument structure analysis, and coreference resolution. We use Wikipedia Annotated Corpus (Hangyo et al., 2014).

Human Examination (English and Japanese) This task evaluates the ability to solve human examination questions. We refer to the English and Japanese versions of this task as **HE-EN** and **HE-JA**, respectively. For **HE-EN**, we use MMLU (Hendrycks et al., 2021) and MMLU-ProX (Xuan et al., 2025) datasets. For **HE-JA**, we use these datasets in addition to JMMLU (Yin et al., 2024).

Mathematical Reasoning (MR) This task measures the ability to solve mathematical problems. We use MAWPS (Kaito et al., 2023) and MGSM (Cobbe et al., 2021) for the evaluation.

Machine Translation (MT) This task evaluates the translation capability from English to Japanese and vice versa. We use ALT (Thu et al., 2016) and WikiCorpus (NICT).

Natural Language Inference (NLI) Natural language inference predicts the logical relationship between two texts, a premise and a hypothesis. We use Jamp (Sugimoto et al., 2023), JaNLI (Yanaka and Mineshima, 2021), JNLI (Kurihara et al., 2022), JSeM (Daisuke and zoeai), and JSICK (Yanaka and Mineshima, 2022) for evaluation.

Question Answering (QA) Question answering evaluates the capability of understanding a question and generating an appropriate answer. We use JEMHopQA (Ishii et al., 2024), NIILC (Sekine, 2003), JAQKET (Masatoshi et al., 2020) for the evaluation.

Reading Comprehension (RC) Reading comprehension evaluates the natural language understanding capability of LLMs. We use JSQuAD (Kurihara et al., 2022).

Summarization (SUM) Summarization is the task of condensing long documents into concise summaries. We use XL-Sum (Hasan et al., 2021).

We report the accuracy for CR, HE-JA, HE-EN, MR, NLI, QA, and RC; the F1 score for EL and FA; the COMET (Rei et al., 2020) score for MT; and IFEval_strict (Dussolle et al., 2025) for IF. For tasks with multiple datasets, we report the mean score across datasets.

[8] https://github.com/llm-jp/llm-jp-eval

Model	Model Variant	EN	JA
Qwen2.5-7B	Instruct	47.6	28.3
Qwen2.5-14B	Instruct	**53.8**	31.9
Llama-3B	Instruct	41.1	23.2
Calm3-22B-chat	-	35.6	28.3
sarashina2.2-1B	Instruct	25.6	19.6
sarashina2-13B	Base	5.3	8.7
Llama3.2-1B	Base	18.0	15.9
	SFT	16.2	15.2
	SFT+Synth	20.3	15.2
Qwen2.5-1.5B	Base	26.6	14.5
	SFT	29.3	18.8
	SFT+Synth	37.6	23.9
llm-jp-3-1.8B	Base	18.0	21.7
	SFT	19.9	21.7
	SFT+Synth	26.6	22.5
	SFT+Synth+DPO	31.2	22.5
llm-jp-3-13B	Base	37.6	18.1
	SFT	34.2	26.1
	SFT+Synth	38.6	33.3
	SFT+Synth+DPO	40.9	**36.2**

Table 3: Results of M-IFEval in English (**EN**) and Japanese (**JA**). **Bold** indicates the best score in each language, while underlined scores are the best within their respective model groups.

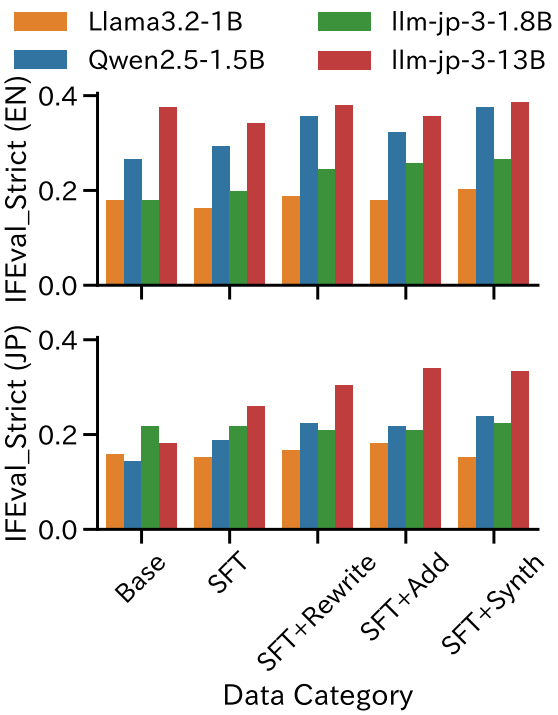

Figure 3: Evaluation results of SFT on the M-IFEval benchmark. *Base* shows the performance of each model before fine-tuning. *SFT* shows the result of fine-tuning on datasets shown in Sec.4.3. *SFT+Add* and *SFT+Rewrite* use datasets described in Sec.4.3 and our synthetic dataset. *SFT+Synth* uses both synthetic dataset.

5. Results

5.1. The Effectiveness of Japanese Synthetic Dataset

Table 3 presents the IF scores in English and Japanese. The results show that instruction-following ability improves after training on our dataset, particularly in Japanese. For Qwen2.5 and llm-jp-3 series, the *SFT+Synth* setting achieves a higher IF score than the corresponding base models and the *SFT* setting. Furthermore, *SFT+Synth+DPO* further improves the instruction-following ability of llm-jp-3 models. The *SFT+Synth+DPO* setting of llm-jp-3-13B shows the highest IF score in Japanese among all models. These results demonstrate that our synthetic dataset effectively improves the instruction-following ability of LLMs.

Table 4 shows the results of the evaluation tasks. When comparing the base models with those trained on our synthetic dataset, performance improvements are observed in most tasks. Specifically, Qwen2.5-1.5B shows higher scores than its base model on all tasks except NLI. A comparison between the *SFT* and *SFT+Synth* settings shows that our synthetic instruction dataset leads to better performance across tasks. Although the *SFT* setting yields gains over the base model, adding our dataset results in additional improvements on almost tasks. For llm-jp-3-1.8B and 13B, our synthetic dataset improves general capabilities compared to *SFT* setting. In contrast, Llama3.2-1B trained on our dataset improves performance on only a subset of tasks compared to the base model. Since the *SFT* setting of Llama3.2 also shows declines across many benchmarks, the degradation cannot be attributed solely to our instruction dataset and may instead be influenced by other datasets. These results suggest that while our synthetic dataset contributes to improved generalization in most models and tasks, it does not always guarantee a performance improvement across all other tasks.

One possible reason for the improvement in specific tasks is that our synthetic dataset contains similar tasks derived from seed instructions and constraint categories. For example, the seed instruction includes translation tasks which is related to the `language>English` category. From this category, LLM generates translation-related instructions, which contributes to the improvements in MT.

5.2. Analysis of Synthetic SFT dataset

To evaluate the effect of the synthetic dataset in SFT, we trained LLMs under several dataset config-

Models	Model Variant	CR	EL	FA	HE-EN	HE-JA	IF	MR	MT	NLI	QA	RC	SUM
Llama3.2-1B	Base	31.3	**20.2**	4.9	**17.4**	**20.3**	17.0	**5.3**	57.4	32.8	**6.5**	**21.8**	1.8
	SFT	**31.9**	15.7	4.9	17.1	18.7	15.7	1.5	59.7	**47.0**	5.8	8.1	5.5
	SFT+Synth	31.2	15.8	**5.4**	16.2	18.8	**17.8**	1.8	**61.8**	45.3	3.9	8.6	**4.0**
Qwen2.5-1.5B	Base	48.6	27.0	5.9	32.1	27.2	20.5	21.9	63.8	47.7	6.1	51.7	2.9
	SFT	61.2	32.5	**8.4**	**35.6**	30.7	24.1	22.9	65.4	**48.7**	11.2	**59.9**	7.4
	SFT+Synth	**62.3**	**35.0**	8.1	35.2	**30.8**	**30.8**	**25.7**	**68.2**	47.7	**12.1**	58.8	**7.5**
llm-jp-3-1.8B	Base	31.3	17.0	12.0	16.5	18.8	19.9	2.6	67.1	40.2	20.4	27.0	1.4
	SFT	32.9	30.6	11.3	17.3	19.8	20.8	7.4	70.9	45.9	21.9	44.8	7.2
	SFT+Synth	33.1	28.8	11.7	17.4	19.2	24.5	**12.4**	71.8	44.9	23.0	43.9	8.5
	SFT+Synth+DPO	**35.6**	**30.8**	**12.5**	**17.7**	**20.3**	**26.8**	12.2	**73.2**	**46.9**	**24.5**	**47.7**	**9.5**
llm-jp-3-13B	Base	47.8	28.7	19.9	21.9	25.1	17.9	24.1	75.5	45.2	44.2	60.1	3.9
	SFT	66.3	44.1	20.8	25.5	29.7	30.1	22.3	79.7	**54.4**	42.0	71.3	8.5
	SFT+Synth	66.1	38.0	20.4	26.8	31.9	36.0	28.7	80.3	52.5	39.1	70.6	9.7
	SFT+Synth+DPO	**69.3**	**46.3**	**21.1**	**27.0**	**32.8**	**38.6**	**29.5**	**81.7**	54.1	**45.4**	**71.9**	**10.3**

Table 4: Evaluation results in various NLP tasks. **Bold** indicates the best score in each model group.

urations. Fig.3 presents the IF scores for each configuration. The *SFT* configuration indicates that the LLMs are trained solely on the dataset described in Sec.4.3. The other configurations involve training on the same dataset augmented with different generation strategies: *SFT+Add*, *SFT+Rewrite*, or their combination *SFT+Synth*.

The results demonstrate that our dataset contributes to improving IF performance. A comparison between *SFT+Add*/*SFT+Rewrite* configurations and the baseline *SFT* configuration reveals that the synthetic dataset enhances instruction-following performance of LLMs. In the llm-jp-3-13B setting, the IF score in Japanese of *SFT* is 26.1, while the score of *SFT+Add* is 34.1. These results indicate that our dataset is effective for the improvement of the IF capabilities when combined with other datasets.

When comparing the two categories of synthetic data, *SFT+Add* and *SFT+Rewrite*, their individual effectiveness appears comparable. The performance gains from using either dataset are similar across the different models and parameter scales (the score is 34.1 for *SFT+Add* and 30.4 for *SFT+Rewrite* setting in llm-jp-3-13B). Notably, the *SFT+Synth* configuration, which combines the two data types, generally achieves the highest IF scores. This indicates that combining diverse synthetic instruction types is necessary to achieve the best performance.

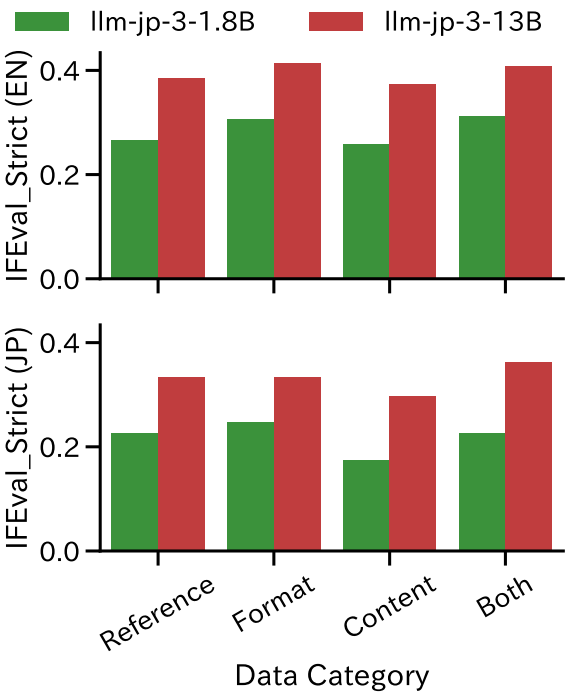

Figure 4: Performance on the M-IFEval benchmark after applying DPO. The *Reference* row shows the scores of the base SFT model before DPO. The *Content*, *Format*, and *Both* rows show the results after applying DPO using the respective preference dataset categories.

5.3. Analysis of Synthetic Preference Dataset

To analyze the effect of the generated preference dataset, we apply DPO using different categories of preference data (*Content*, *Format*). The *Both* category applies DPO using both *Content* and *Format* data. The reference model is fine-tuned on the instruction dataset described in Sec.4.3 together with our synthetic instruction dataset. Fig.4 presents the IF scores after applying DPO in English and Japanese.

The performance gains differ across the DPO dataset categories. The *Content* dataset degrades IF scores in both languages and across all models compared to the reference model. On the other hand, the *Format* dataset consistently improves IF performance across both model sizes. There are two possible reasons for this result. First, as shown in Table 2, the *Content* dataset is smaller

Instruction	Chosen Response	Rejected Response
あなたが知っているプログラミング言語をcsv形式で列挙してください。(Please list the programming languages you know in CSV format.)	Python,C++,Java, JavaScript,PHP,Ruby, C#,Go,Swift,R	Python,Java,C++, JavaScript,Ruby,Swift, Go,Kotlin,R,PHP

Table 5: Example of generated preference dataset of `Content` strategy. Although the rejected response is labeled as negative, it provides an accurate and complete answer to the instruction, illustrating potential noise in the preference dataset.

than the *Format* dataset. Second, the quality of *Content* is insufficient for the improvement of the IF scores. Table 5 presents an example instruction along with its chosen and rejected responses in *Content* dataset. In this example, the rejected response is a correct and complete answer to the given instruction. Because the *Content* dataset is small and noisy, it leads to degraded instruction-following performance. The generation of high-quality *Content* datasets is required for the further improvements.

IF capabilities improve in the *Both* setting, which combines the *Content* and *Format* datasets, compared to the reference model. However, the Japanese IF score in the *Both* setting is lower for the 1.8B model and the English IF score is lower for the 13B model compared to the *Format* setting. These results indicate that the performance improvements in the *Both* setting are attributed to the *Format* dataset.

6. Conclusion

In this paper, we propose an automatic pipeline for generating instruction and preference datasets and evaluate the effectiveness of these datasets on various tasks in Japanese. The instruction dataset was created by adding new constraints to manually written instructions or by rewriting them to adhere to different constraints. For preference dataset generation, we treated the responses in the instruction dataset as the chosen responses and generated the rejected responses. We evaluated the generated datasets by applying SFT and DPO to various LLMs. The experimental results show that our synthetic dataset improves the instruction-following ability of LLMs, especially in Japanese contexts. For other tasks, our synthetic dataset improves performance on tasks such as translation and entity linking. The analysis of SFT reveals that the improvement in instruction following is attributable not only to existing datasets but also to our synthetic dataset. These results show that the data augmentation using multilingual LLMs is effective for instruction tuning in Japanese.

7. Bibliographical References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D 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 Ziegler, Jeffrey Wu, Clemens Winter, Chris 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 *Proceedings of the 34th International Conference on Neural Information Processing Systems (NeurIPS 2020)*, volume 33, pages 1877–1901.

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 [cs.LG]*.

Bekki Daisuke and zoeai. Jsem: Japanese semantic test suite (japanese fracas and extensions).

Antoine Dussolle, Andrea Cardeña Díaz, Shota Sato, and Peter Devine. 2025. M-IFEval: Multilingual instruction-following evaluation. *arXiv [cs.CL]*.

Masatsugu Hangyo, Daisuke Kawahara, and Sadao Kurohashi. 2014. Building and analyzing a diverse document leads corpus annotated with semantic relations. *Journal of Natural Language Processing*, 21(2):213–247.

Tahmid Hasan, Abhik Bhattacharjee, Md. Saiful Islam, Kazi Mubasshir, Yuan-Fang Li, Yong-Bin Kang, M. Sohel Rahman, and Rifat Shahriyar. 2021. XL-sum: Large-scale multilingual abstractive summarization for 44 languages. In *Findings of the Association for Computational Linguistics (ACL-IJCNLP 2021)*, pages 4693–4703.

Kan Hatakeyama. 2024a. kanhatakeyama/automultiturnbycalm3-22b.

Kan Hatakeyama. 2024b. kanhatakeyama/ramdom-to-fixed-multiturn-calm3.

Qianyu He, Jie Zeng, Wenhao Huang, Lina Chen, Jin Xiao, Qianxi He, Xunzhe Zhou, Jiaqing Liang, and Yanghua Xiao. 2024. Can large language models understand real-world complex instructions? In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI 2024)*, volume 38, pages 18188–18196.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *Proceedings of the Ninth International Conference on Learning Representations (ICLR 2021)*.

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

Shijia Huang, Jianqiao Zhao, Yanyang Li, and Liwei Wang. 2023. Learning preference model for LLMs via automatic preference data generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP 2023)*, pages 9187–9199.

Ryosuke Ishigami. 2024. cyberagent/calm3-22b-chat.

Ai Ishii, Naoya Inoue, Hisami Suzuki, and Satoshi Sekine. 2024. JEMHopQA: Dataset for Japanese explainable multi-hop question answering. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 9515–9525.

Horio Kaito, Murata Eiki, Wang Hao, Ide Tatsuya, Kawahara Daisuke, Yamazaki Takato, Shinzato Kenta, Nakamachi Akifumi, Li Shengzhe, and Sato Toshinori. 2023. Verification of chain-of-thought prompting in japanese. *The 37th Annual Conference of the Japanese Society for Artifical Intelligence (JSAI 2023)*.

Abdullatif Köksal, Timo Schick, Anna Korhonen, and Hinrich Schuetze. 2024. LongForm: Effective instruction tuning with reverse instructions. In *Findings of the Association for Computational Linguistics (ACL 2024)*, pages 7056–7078.

Kentaro Kurihara, Daisuke Kawahara, and Tomohide Shibata. 2022. JGLUE: Japanese general language understanding evaluation. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference (LREC 2022)*, pages 2957–2966.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Suhyun Lee and Changheon Han. 2025. Sentimatic: Sentiment-guided automatic generation of preference datasets for customer support dialogue system. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT SRW 2025)*, pages 120–128.

Zhuoyan Li, Hangxiao Zhu, Zhuoran Lu, and Ming Yin. 2023. Synthetic data generation with large language models for text classification: Potential and limitations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP 2023)*, pages 10443–10461, Singapore.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81.

Renze Lou, Kai Zhang, Jian Xie, Yuxuan Sun, Janice Ahn, Hanzi Xu, Yu Su, and Wenpeng Yin. 2023. MUFFIN: Curating multi-faceted instructions for improving instruction following. In *Proceedings of the Twelfth International Conference on Learning Representations (ICLR 2023)*.

Suzuki Masatoshi, Suzuki Jun, Matsuda Kouji, Nishida Kyousuke, and Inoue Naoya. 2020. JAQKET: Construction of japanese qa datset based on the quiz. In *26th Annual Meeting of the Association for Natural Language Processing (NLP 2020)*. In Japanese.

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 (ACL 2022)*, pages 3470–3487.

NICT. Japanese-english bilingual corpus of wikipedia's kyoto articles.

Kazumasa Omura, Fei Cheng, and Sadao Kurohashi. 2024. An empirical study of synthetic

data generation for implicit discourse relation recognition. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 1073–1085.

Kazumasa Omura, Daisuke Kawahara, and Sadao Kurohashi. 2023. Building a Commonsense Inference Dataset based on Basic Events and its Application. *Journal of Natural Language Processing*, 30(4):1206–1239. In Japanese.

Alexander Pan, Jun Shern Chan, Andy Zou, Nathaniel Li, Steven Basart, Thomas Woodside, Hanlin Zhang, Scott Emmons, and Dan Hendrycks. 2023. Do the rewards justify the means? measuring trade-offs between rewards and ethical behavior in the machiavelli benchmark. In *Proceedings of the 40th International Conference on Machine Learning (ICML 2023)*, pages 26837–26867.

Yiwei Qin, Kaiqiang Song, Yebowen Hu, Wenlin Yao, Sangwoo Cho, Xiaoyang Wang, Xuansheng Wu, Fei Liu, Pengfei Liu, and Dong Yu. 2024. InFoBench: Evaluating instruction following ability in large language models. In *Findings of the Association for Computational Linguistics (ACL 2024)*, pages 13025–13048.

Qwen, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2025. Qwen2.5 technical report. *arXiv [cs.CL]*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Proceedings of the 37th Conference on Neural Information Processing Systems (NeurIPS 2023)*.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020)*, pages 2685–2702.

Timo Schick and Hinrich Schütze. 2021. Generating datasets with pretrained language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP 2021)*, pages 6943–6951.

Maximilian Schmidt, Andrea Bartezzaghi, and Ngoc Thang Vu. 2024. Prompting-based synthetic data generation for few-shot question answering. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 13168–13178.

Satoshi Sekine. 2003. Development of a question answering system focused on an encyclopedia. In *9th Annual Meeting of the Association for Natural Language Processing (NLP2003).* In Japanese.

Satoshi Sekine, Ando Maya, Goto Michiko, Suzuki Hisami, Kawahara Daisuke, Inoue Naoya, and Inui Kentaro. 2024. ichikara-isntruction construction of japanese instruction dataset for llms. In *In Proceedings of The Thirtieth Annual Meeting of The Association for Natural Language Processing (NLP 2024)*, pages 1508–1013. In Japanese.

Tomoki Sugimoto, Yasumasa Onoe, and Hitomi Yanaka. 2023. Jamp: Controlled Japanese temporal inference dataset for evaluating generalization capacity of language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL SRW 2023)*, pages 57–68.

Yikun Sun, Zhen Wan, Nobuhiro Ueda, Sakiko Yahata, Fei Cheng, Chenhui Chu, and Sadao Kurohashi. 2024. Rapidly developing high-quality instruction data and evaluation benchmark for large language models with minimal human effort: A case study on japanese. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 13537–13547.

Hisami Suzuki, Satoru Katsumata, Takashi Kodama, Tetsuro Takahashi, Kouta Nakayama, and Satoshi Sekine. 2025. Answercarefully: A dataset for improving the safety of japanese llm output.

Kubo Takahiro and Nakayama Hiroki. chabsa: Aspect based sentiment analysis dataset in japanese.

Masashi Takeshita, Rafal Rzpeka, and Kenji Araki. 2023. Jcommonsensemorality: Japanese dataset for evaluating commonsense morality understanding. In *In Proceedings of The Twenty Nineth Annual Meeting of The Association for Natural Language Processing (NLP 2023)*, pages 357–362. In Japanese.

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.

Ye Kyaw Thu, Win Pa Pa, Masao Utiyama, Andrew Finch, and Eiichiro Sumita. 2016. Introducing the Asian language treebank (ALT). In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 1574–1578.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models.

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 (ACL 2023)*, pages 13484–13508.

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 (EMNLP 2022)*, pages 5085–5109.

Zhilin Wang, Yi Dong, Olivier Delalleau, Jiaqi Zeng, Gerald Shen, Daniel Egert, Jimmy J. Zhang, Makesh Narsimhan Sreedhar, and Oleksii Kuchaiev. 2024. Helpsteer2: Open-source dataset for training top-performing reward models. *arXiv [cs.CL]*.

Bosi Wen, Pei Ke, Xiaotao Gu, Lindong Wu, Hao Huang, Jinfeng Zhou, Wenchuang Li, Binxin Hu, Wendy Gao, Jiaxin Xu, Yiming Liu, Jie Tang, Hongning Wang, and Minlie Huang. 2024. Benchmarking complex instruction-following with multiple constraints composition. In *Proceedings of the 38th Annual Conference on Neural Information Processing Systems (NeurIPS 2024)*, volume 37, pages 137610–137645.

Congying Xia, Chen Xing, Jiangshu Du, Xinyi Yang, Yihao Feng, Ran Xu, Wenpeng Yin, and Caiming Xiong. 2024. FOFO: A benchmark to evaluate LLMs' format-following capability. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL 2024)*, pages 680–699.

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

Canwen Xu, Daya Guo, Nan Duan, and Julian McAuley. 2023. Baize: An open-source chat model with parameter-efficient tuning on self-chat data. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP 2023)*, pages 6268–6278.

Weihao Xuan, Rui Yang, Heli Qi, Qingcheng Zeng, Yunze Xiao, Yun Xing, Junjue Wang, Huitao Li, Xin Li, Kunyu Yu, Nan Liu, Qingyu Chen, Douglas Teodoro, Edison Marrese-Taylor, Shijian Lu, Yusuke Iwasawa, Yutaka Matsuo, and Irene Li. 2025. Mmlu-prox: A multilingual benchmark for advanced large language model evaluation. *arXiv [cs.CL]*.

Hitomi Yanaka and Koji Mineshima. 2021. Assessing the generalization capacity of pre-trained language models through japanese adversarial natural language inference. In *Proceedings of the 2021 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP (BlackboxNLP2021)*.

Hitomi Yanaka and Koji Mineshima. 2022. Compositional evaluation on japanese textual entailment and similarity. *Transactions of the Association for Computational Linguistics*, 10:1266–1284.

Yiben Yang, Chaitanya Malaviya, Jared Fernandez, Swabha Swayamdipta, Ronan Le Bras, Ji-Ping Wang, Chandra Bhagavatula, Yejin Choi, and Doug Downey. 2020. Generative data augmentation for commonsense reasoning. In *Findings of the Association for Computational Linguistics (ACL 2020)*, pages 1008–1025.

Junjie Ye, Caishuang Huang, Zhuohan Chen, Wenjie Fu, Chenyuan Yang, Leyi Yang, Yilong Wu,

Peng Wang, Meng Zhou, Xiaolong Yang, Tao Gui, Qi Zhang, Zhongchao Shi, Jianping Fan, and Xuanjing Huang. 2025. A multi-dimensional constraint framework for evaluating and improving instruction following in large language models. *arXiv [cs.CL]*.

Da Yin, Xiao Liu, Fan Yin, Ming Zhong, Hritik Bansal, Jiawei Han, and Kai-Wei Chang. 2023. Dynosaur: A dynamic growth paradigm for instruction-tuning data curation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP 2023)*, pages 4031–4047.

Ziqi Yin, Hao Wang, Kaito Horio, Daisuke Kawahara, and Satoshi Sekine. 2024. Should we respect LLMs? a cross-lingual study on the influence of prompt politeness on LLM performance. In *Proceedings of the Second Workshop on Social Influence in Conversations (SICon 2024)*, pages 9–35.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *Proceedings of the 8th International Conference on Learning Representations (ICLR 2020)*.

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

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023. Instruction-following evaluation for large language models. *arXiv [cs.CL]*.

A. Categories and its Descriptions

Here is the description of the translated-constraint categories and its description used in the generation and validation. The categories are organized in a hierarchical, bulleted list, with a brief description provided for each sub-category.

- **Character**
  - **Word**: Specifies the words to be used in the response.
  - **Character**: Specifies the characters to be used in the response.
  - **Placeholder**: Enclose the response with arbitrary symbols such as [].
  - **Punctuation**: Has constraints on punctuation.
- **Length**
  - **Paragraph**: Summarize a long text into multiple paragraphs, or the content of each paragraph is specified.
  - **Sentence**: The number of sentences to be used in the response is specified.
  - **Word**: Translate foreign languages or explain words in the response.
  - **Character**: The number of characters in the response sentence is specified.
- **Language**
  - **English**: Specifies the explanation of English words or the generation of English sentences in the response.
- **Character type**
  - **Hiragana**: Includes the constraint that the response text is in hiragana.
  - **Katakana**: Includes the constraint that the response text is in katakana.
  - **Number**: Has the constraint that the response text contains numbers.
  - **Alphabet**
    - **Lowercase**: The response text contains lowercase alphabets.
    - **Uppercase**: The response text contains uppercase alphabets.
- **Format**
  - **Chapter/Section**: Includes the constraint that the response text is structured.
  - **List**
    - **Ordered**
      - **Markdown**: The response text is a markdown formatted list and the order is specified.
      - **Arbitrary symbol**: The response text is a list using arbitrary symbols and the order is specified.
    - **Unordered**
      - **Markdown**: The response text is a markdown formatted list.
      - **Arbitrary symbol**: The response text is a list using arbitrary symbols.
  - **Table**
    - **Markdown**: The response text is a markdown formatted table.

    - * **csv**: Includes the constraint that the response text is a csv formatted table.
  - **Data**
    - * **json**: Includes the constraint that the response text is json formatted data.
  - **Selection**
    - * **Yes/No**: Has the constraint that the response text is 'Yes' or 'No'.
    - * **Multiple choice**: There are choices, select from them, and have the response be the selection.
  - **Exclusive**: Has the constraint that the response is only the answer.
  - **Answer field**: The format of the response to the instruction is specified.
  - **Repetition**: Has the specification to repeat the instruction once and then generate the response.
- **Decoration**
  - **Bold**
    - * **Markdown**: The response includes markdown bold.
    - * **Arbitrary symbol**: Includes the constraint that the answer is in bold in the response.
  - **Italic**
    - * **Markdown**: The response includes markdown italics.
    - * **Arbitrary symbol**: The response includes the answer in arbitrary italics.
  - **Horizontal line**
    - * **Markdown**: Has the constraint to include a markdown formatted horizontal line in the response.
    - * **Arbitrary symbol**: Has the constraint to include a horizontal line with arbitrary symbols in the response.
  - **Heading**
    - * **Markdown**: Has the constraint to use markdown headings in the response.
    - * **Arbitrary symbol**: Has the constraint to use headings with arbitrary symbols such as [] or () in the response.
- **Prohibition**: Has constraints such as prohibiting the use of certain words or punctuation in the response.
- **Frequency**: Has the constraint to list the specified number of answers in the response.
- **Position**
  - **Start**: The starting sentence or word of the response is specified.
  - **End**: The ending sentence or word of the response is specified.
  - **Specified position**: The format of the response is specified at a specific location in the response.

B. Prompts for the Data Generation and LLM-as-a-Judge

Fig.5 and Fig.6 show the prompts for the *Add* and *Rewrite* instruction datasets, respectively. `${category}` is replaced by a constraint category and `${category_instruction}` is replaced by its corresponding description, as detailed in Sec.A. The `${seed}` placeholder is replaced by a seed instruction from the ichikara-instruction dataset.

Fig.7 shows the prompt used for the answer generation. In this prompt, `${prompt}` placeholder is replaced by an instruction.

Fig.8 and Fig.9 present the prompt of the LLM-as-a-Judge to evaluate the generated instructions and instruction-answer pairs, respectively. This prompt includes three evaluation examples, each consisting of an example instruction, its corresponding evaluation scores, and the rationale for those scores.

Fig.10 and Fig.11 present the generation of the negative response of *Content* and *Format* dataset, respectively. The evaluation prompts for these datasets are shown in Fig.12 and Fig.13 for the *Format* and *Content* datasets.

C. Details of Hyper-parameter of Training LLMs

We used the same set of hyperparameters for all models in our experiment. The hyperperameters for the SFT and DPO are listed in Table 6 and Table 7, respectively.

Learning Rate	2×10^{-5}
Learning Rate (Min)	2×10^{-6}
Warmup Steps	20
Global Batch Size	64
Micro Batch Size	2
Weight Decay	0.1
Optimizer	Fused Adam
Scheduler	CosineAnnealing
Epochs	2

Table 6: Hyperparamerters of SFT in our experiment.

You are a prompt designer.
Your objective is to create prompts for an instruction-constrained dataset to be used for LLM training.
Please add a concise instruction constraint (around 10-15 words) belonging to the `${category}` category to the instruction below.
The instruction for the `${category}` category must be `${category_instruction}`.
The instruction must begin with [Start of Question] and end with [End of Question].
[Start of Question]
`${seed}`
[End of Question]

Figure 5: Prompt for the isntruction generation of Add dataset.

You are a prompt designer.
Your objective is to create prompts for an instruction-constrained dataset to be used for LLM training.
Please rewrite the following instruction into one that has an instruction constraint belonging to the `${category}` category.
The instruction for the `${category}` category must be `${category_instruction}`.
You may change the content of the instruction itself.
The instruction must begin with [Start of Question] and end with [End of Question].
[Start of Question]
`${seed}`
[End of Question]

Figure 6: Instruction generation prompt of Rewrite dataset

Learning Rate	9×10^{-9}
Learning Rate (Min)	5×10^{-9}
Warmup Steps	20
Global Batch Size	128
Micro Batch Size	1
Weight Decay	0.1
Optimizer	Fused Adam
Scheduler	CosineAnealing
KL penalty	0.5
Epochs	1

Table 7: Hyperpararmeters of DPO in our experiment.

Please generate a response to the following instruction.
Be aware that the response includes constraints.
The response should begin with [Start of Response] and end with [End of Response].
`${prompt}`

Figure 7: Prompt for the answer generation.

Please evaluate the given instruction for the AI assistant according to the following items. The evaluation criteria are as follows.
Relevance: Evaluate how appropriate the given instruction is for its instruction category.
Fluency: Evaluate whether the instruction's wording is natural and grammatically correct.
Redundancy: Evaluate whether the instruction contains unnecessary expressions or multiple instructions.
Start the evaluation with a brief explanation, followed by the rating in the format: `Evaluation:[[Relevance:1-5, Fluency:1-5, Redundancy:1-5]]`. The response should only contain the evaluation. Please be careful not to include Chinese characters (Kanji).
Here are the example evaluations.
⋮
(Here are n examples)
Example n: The instruction belongs to `${category}`, and its description is `${category_instruction}`
[Start of Instruction]
`${n-th example instruction}`
[End of Instruction]
Evaluation: `${n-th example evaluation}`
`${n-th reason of evaluation}`
⋮
The category the instruction belongs to is "`${category}`", and the description for this category is "`${category_instruction}`".
The instruction for the AI assistant is as follows.
[Start of Instruction]
`${instruction}`
[End of Instruction]

Figure 8: Prompt of LLM-as-a-judge for the instruction evaluation. `${instruction}` is replaced by the instruction to be evaluated.

Please evaluate the given instruction and response pair for the AI assistant according to the following items. The evaluation criteria are as follows.
- **Adherence**: Evaluate whether the response follows the instruction and aligns with the instruction category.
- **Fluency**: Evaluate whether the response's wording is natural and grammatically correct.
- **Conciseness**: Evaluate whether the response contains unnecessary expressions and is presented concisely.
- **Completeness**: Evaluate whether the response completely answers the instruction.
Start the evaluation with a brief explanation, followed by the rating in the format:
`Evaluation:[[Adherence:1-5, Fluency:1-5, Conciseness:1-5, Completeness:1-5]].`
The category for the instruction and response is "`${category}`", and its description is "`${category_instruction}`".
Examples of evaluations are shown below.
⋮
(here are n examples.)
Example n: The category the instruction belongs to is "`${example_category}`", and the description for this category is "The number of sentences to be used in the response is specified".
[Start of Instruction]
`${n-th example instruction}`
[End of Instruction]
[Start of Response]
`${n-th example answer}`
[End of Response]
`Evaluation:${n-th example_evaluation}`
⋮
The pair to be evaluated is as follows.
[Start of Instruction]
`${instruction}`
[End of Instruction]
[Start of Response]
`${answer}`
[End of Response]

Figure 9: Prompt of the LLM-as-a-Judge for the evaluation of instruction-answer pairs. The `${instruction}` and `${answer}` placeholders are replaced by the corresponding instruction and answer, respectively.

Please generate a response to the following instruction.
However, the response should follow the constraints, but its content should be irrelevant.
The response must begin with `[Start of Response]` and end with `[End of Response]`.
The following are specific examples.
⋮
Example n:
[Start of Instruction]
`${n-th example of instruction}`
[End of Instruction]
[Start of Response]
`${n-th example of response}`
[End of Response]
`${n-th reason of rejected responses}`
⋮
Referring to the examples above, please generate a response that follows the constraints of the following instruction but has irrelevant content.
[Start of Instruction]
`${instruction}`
[End of Instruction]

Figure 10: Prompt for the rejected response for the content dataset. The `${instruction}` placeholder is replaced by the corresponding instruction.

Please generate a response to the following instruction.
However, the response should **not** follow the constraints, but its content **should** be relevant.
The response must begin with `[Start of Response]` and end with `[End of Response]`.
⋮
Example n:
[Start of Instruction]
`${n-th example of instruction}`
[End of Instruction]
[Start of Response]
`${n-th example of response}`
[End of Response]
`${n-th reason of rejected responses}`
⋮
The instruction is shown below.
Please generate a response that does not follow the constraints but has relevant content.
[Start of Instruction]
`${instruction}`
[End of Instruction]

Figure 11: Prompt for the generation of rejected response of format dataset. The `${instruction}` placeholder is replaced by the corresponding instruction.

Please evaluate the given instruction and response pairs for the AI assistant according to the following items.
The evaluation criteria are as follows.
The response pairs include a preferred example and a negative example.
Please evaluate the negative example according to the following criteria.
- **Adherence**: The negative example should follow the instruction's constraints but have irrelevant content. Evaluate how well it adheres to this specification, comparing it to the preferred example.
- **Fluency**: Evaluate whether the response's wording is natural and grammatically correct.
Start the evaluation with a brief explanation, followed by the rating in the format: `[[Evaluation:Adherence:1-5, Fluency:1-5]]`.
The following are evaluation examples.
⋮
n-th Example:
[[Start of Instruction]]
`${n-th example instruction}`
[[End of Instruction]]
[[Start of Preferred Example]]
`${n-th example of preferred response}`
[[End of Preferred Example]]
[[Start of Negative Example]]
`${n-th example of negative response}`
[[End of Negative Example]]
[Evaluation]
`${n-th example of evaluation}`
⋮
The target for evaluation is as follows.
[[Start of Instruction]]
`${instruction}`
[[End of Instruction]]
[[Start of Preferred Answer]]
`${preferred_answer}`
[[End of Preferred Answer]]
[[Start of Negative Answer]]
`${negative_answer}`
[[End of Negative Answer]]

Figure 12: Prompt for LLM-as-a-Judge for format data.

Please evaluate the given instruction and response pair for the AI assistant according to the following items. The evaluation criteria are as follows.
The response pairs include a positive example and a negative example.
Please evaluate the negative example according to the following criteria.
-**Adherence**: Evaluate the negative example's response to the instruction. A high rating should be given if the content is relevant but the response does not follow the constraints.
- **Fluency**: Evaluate whether the response's wording is natural and grammatically correct.
The following are evaluation examples.
⋮
n-th Example:
[[Start of Instruction]]
`${n-th example instruction}`
[[End of Instruction]]
[[Start of Preferred Example]]
`${n-th example of preferred response}`
[[End of Preferred Example]]
[[Start of Negative Example]]
`${n-th example of negative response}`
[[End of Negative Example]]
[Evaluation]
`${n-th example of evaluation}`
⋮
The target for evaluation is as follows.
[[Start of Instruction]]
`${instruction}`
[[End of Instruction]]
[[Start of Preferred Answer]]
`${preferred_answer}`
[[End of Preferred Answer]]
[[Start of Negative Answer]]
`${negative_answer}`
[[End of Negative Answer]]

Figure 13: Prompt for LLM-as-a-Judge for content data.