From Base to Conversational: Japanese Instruction Dataset and Tuning Large Language Models

Masahiro Suzuki
The University of Tokyo
Tokyo, Japan
research@msuzuki.me

Masanori Hirano
The University of Tokyo
Tokyo, Japan
research@mhirano.jp

Hiroki Sakaji
The University of Tokyo
Tokyo, Japan
sakaji@sys.t.u-tokyo.ac.jp

Abstract—Instruction tuning is essential for large language models (LLMs) to become interactive. While many instruction tuning datasets exist in English, there is a noticeable lack in other languages. Also, their effectiveness has not been well verified in non-English languages. We construct a Japanese instruction dataset by expanding and filtering existing datasets and apply the dataset to a Japanese pre-trained base model. We performed Low-Rank Adaptation (LoRA) tuning on both Japanese and English existing models using our instruction dataset. We evaluated these models from both quantitative and qualitative perspectives. As a result, the effectiveness of Japanese instruction datasets is confirmed. The results also indicate that even with relatively small LLMs, performances in downstream tasks would be improved through instruction tuning. Our instruction dataset, tuned models, and implementation are publicly available online.

Index Terms—Large Language Model (LLM), Instruction Dataset, Instruction Tuning, Japanese

I. INTRODUCTION

Large language models (LLMs) have been making remarkable progress in performance and generalization in recent years. Various Transformer-based [1] language models, such as BERT [2], RoBERTa [3], and the GPT series [4]–[6], have demonstrated high performance derived from pre-training. Furthermore, since 2022, a large number of models, such as OPT [7], GPT-NeoX-20B [8], UL2 [9], PaLM [10], BLOOM [11], Pythia [12], and LLaMA series [13], [14], have emerged as models that show higher performance by scaling their size [15].

Although there is still difficulty in few-shot or zero-shot performance on unseen tasks, instruction tuning can address this issue [16]. Instruction tuning is a training method that improves the performance in unseen tasks by solving various tasks described via natural language instructions [16]. Starting with the enhancement of performance in various tasks by GPT-3 [6] under a few-shot setting given in natural language, there has been an increasing demand for responses in formats that are closer to question-answering or conversation, especially formats that are not similar to the pre-training data.

An increasing number of datasets for instruction tuning and instruct-tuned models are being made available to the public. For instance, various datasets like FLAN [16], P3 [17], databricks-dolly-15k [1], and OASST1 [18] have been proposed

and made public. As publicly available models, Flan-T5 [19] was constructed using FLAN and T0 was constructed using P3 respectively. Also, Dolly [20] is a model with instruction tuning applied to Pythia [12], while Vicuna [21] and Alpaca [22] are models with instruction tuning applied to LLaMA [13].

However, these models are not fully compatible with languages other than English. The datasets used for instruction tuning in Dolly, Alpaca, and Vicuna are only in English, making it difficult to gain the benefits of these models in languages other than English. Many instruction datasets have been constructed in English, and there are not many efforts to construct instruction datasets in languages other than English. While there are movements to construct instruction datasets in Chinese [23], most instruction dataset in non-English languages are built from outputs of models with licensing restrictions, such as translations of the Alpaca dataset [22] or the ShareGPT52K [2] constructed from ChatGPT outputs. In languages other than English, the scarcity of comprehensive instruction datasets means that the verification of instruction tuning effects is limited [24]. In Japanese, only some data from translated Alpaca [22] and OASST1 [18] exists, and there's a lack of dataset diversity, with quantitative evaluations of instruction tuning yet to be conducted. While constructing and evaluating datasets in languages other than English is a crucial step towards building language models that can interact in various languages, it's still very much in its early stages.

To tackle the issue of the lack of Japanese instruction dataset, the study [25] gathers various Japanese datasets to build an instruction dataset. While this dataset seems valuable, the effect of instruction tuning is only shown qualitatively and not quantitatively. Furthermore, the majority of this dataset consists of translation tasks. While it is considered that the translation tasks are effective when adapting English-based models to Japanese, these tasks may not be optimal for Japanese-based models. To apply the instruction dataset to a Japanese-based model, it is desirable to filter out the translation data and construct an instruction dataset consisting solely of Japanese.

We construct an instruction dataset consisting solely of Japanese for instruction tuning based on a Japanese model by filtering and expanding the Japanese instruction dataset [25].

---

[1] https://huggingface.co/datasets/databricks/databricks-dolly-15k

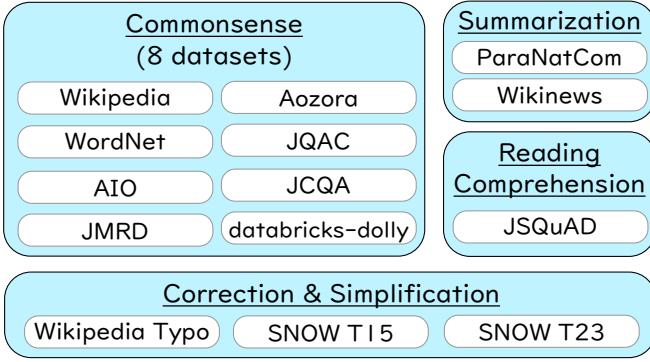[2] https://huggingface.co/datasets/RyokoAI/ShareGPT52K

Fig. 1. Datasets and task clusters used in llm-japanese-dataset-vanilla v1.0.1.

The constructed dataset contains about 2.5 million samples and 5 tasks, such as commonsense, summarization, reading comprehension, simplification, and correction. Using this dataset, which contains various tasks, we perform instruction tuning on both Japanese-based and English-based LLMs. For Japanese-based models, we conduct tuning using an instruction dataset without translation data, while for English-based models, we do using an instruction dataset that includes translation data. As a result of quantitative evaluation with the tuned model, we demonstrate that instruction tuning in Japanese improve the performance in downstream tasks, thereby illustrating the effectiveness of the Japanese instruction dataset. The following materials used in this study are available as open source.

- Japanese instruction dataset:
  https://huggingface.co/datasets/izumi-lab/llm-japanese-dataset-vanilla
- Tuned model (Stormy 10 epochs):
  https://huggingface.co/izumi-lab/stormy-7b-10ep
- Tuned model (LLaMA 7B 5 epochs):
  https://huggingface.co/izumi-lab/llama-7b-japanese-lora-v0-5ep
- Implementation for training and evaluation:
  https://github.com/retarfi/jallm

Here are our main contributions: (1) We construct a Japanese instruction dataset, llm-japanese-dataset-vanilla, for Japanese-based models; (2) We clarified the benefits of instruction tuning for Japanese and English models from evaluating with some Japanese downstream tasks; (3) Unlike previous research [16], we show that even with smaller model sizes, instruction tuning can lead to performance gains in downstream tasks.

II. INSTRUCTION DATASET

We construct a Japanese instruction dataset without translation tasks. We use the llm-japanese-dataset v0.1.0 [25] as a main data source for the Japanese instruction dataset and expand this dataset with additional Japanese datasets. The llm-japanese-dataset v0.1.0 contains about 8.4 million instruction examples, of which more than 75 % (6,581,044) are constructed based on translation data. This dataset is intended to link English and Japanese and extract the knowledge

learned in English for use in Japanese as well, considering that many LLMs like LLaMA show good performance in English. However, when it comes to Japanese-based models, they are usually pre-trained with Japanese corpora. The need for the English part of this dataset is relatively low because the part aimed to link English and Japanese. Therefore, we extract 1,811,964 data excluding translation tasks from the llm-japanese-dataset v0.1.0. Furthermore, to expand the variety of datasets, we incorporated the Japanese Wikipedia Typo Dataset (Wikipedia Typo) [26] and the Japanese Question-Answering Corpus (JQAC) [27]. From the Wikipedia Typo and JQAC, we newly created 697,316 and 906 instruction entries respectively. Additionally, we addressed licensing issues present in version v0.1.0, and ultimately constructed a total of 2,463,624 instruction data entries, releasing it as llm-japanese-dataset-vanilla v1.0.1 [3]. Figure 1 shows datasets and task classifications included in llm-japanese-dataset-vanilla v1.0.1.

We use the instruction, input, and response included in llm-japanese-dataset-vanilla v0.1.0, following the format below.

— Prompt format with input —

Below is an instruction that describes a task, paired with an input that provides further context. Write a response that appropriately completes the request.

Instruction:
{Instruction}

Input:
{Input}

Response:
{Resoponse} [2]

— Prompt format with no input —

Below is an instruction that describes a task. Write a response that appropriately completes the request.

Instruction:
{Instruction}

Response:
{Resoponse} [2]

III. INSTRUCTION LoRA TUNING

We perform Low-Rank Adaptation (LoRA) tuning [28] on two publicly available LLMs. In this section, we describe the base model and the process of LoRA tuning.

A. Models

We use two models: a Japanese-based model and an English-based model. The models we use are the Japanese-

[3]https://huggingface.co/datasets/izumi-lab/llm-japanese-dataset-vanilla
[2]Originally written in Japanese.

TABLE I
THE PARAMETERS OF LORA TUNING

Parameter	Stormy	Instruct LLaMA 7B	Instruct LLaMA 13B [25]
Base Model	CALM 7B	LLaMA 7B	LLaMA 13B
Learning Rate	3e-4	3e-4	3e-4
Sequence Length	300	256	256
Batch	128	128	130
# of data	1.4M	8.4M	8.4M
Epochs	10	5	1
r in LoRA	4	4	4
α in LoRA	16	16	16
Dropout Ratio of LoRA	0.05	0.05	0.05
Tuning Parameters	query_key_value	q_proj, v_proj	q_proj, v_proj

based OpenCALM-7B (hereafter CALM) and the English-based LLaMA 7B. CALM is a model with 7 billion parameters released by CyberAgent [3]. It is pre-trained on Japanese Wikipedia and Common Crawl using the GPT-NeoX architecture [8]. For the English-based model, we use the 7B model of LLaMA [13] (hereafter LLaMA 7B), which is released by Meta [4].

Due to the differences in characteristics between the Japanese-based CALM and the English-based LLaMA 7B, we use llm-japanese-dataset-vanilla, which we constructed above, for CALM and llm-japanese-dataset for LLaMA 7B as training data. For tuning CALM, we use version v0.1.0 as the training data, which excludes the JQAC and Wikipedia Typo datasets. This is done to align with the model constructed in the literature [25], ensuring dataset consistency with the exception of not including English.

We train LLaMA 7B on the entire llm-japanese-dataset v0.1.0, following the methods outlined in [25]. We adopt the same input format as described in [25]. From this point forward, the tuned CALM will be referred to as "Stormy," and the LLaMA 7B as "Instruct LLaMA 7B."

B. LoRA Tuning

LLMs, having a large number of parameters, require GPU resources not only for pre-training but also for fine-tuning. In this study, we use LoRA [28] as a method for tuning LLMs without significantly reducing accuracy. In LoRA, only the difference between the initial and updated LLM parameters, represented with small-scale parameters, is calculated. Consider an example of updating the parameter matrix $W_0 \in \mathbb{R}^{d \times k}$ of a certain linear layer that LLM has. Instead of training W_0 directly, initialize the difference $\Delta W \in \mathbb{R}^{d \times k}$ to W_0 with a zero matrix, update the difference δW, and proceed with training by updating the parameters to $W_0 + \Delta W$. Here, we set $\Delta W = BA$ where $B \in \mathbb{R}^{d \times r}$ and $A \in \mathbb{R}^{r \times k}$ are matrices of rank $r \ll \min(d, k)$. This can reduce the number of learnable parameters from dk to $(d + k)r$.

The primary parameters utilized in the experiment are shown in Table I. For comparison, we also mention the model

that Instruct LLaMA 13B [25], which was LoRA-tuned with llm-japanese-dataset v0.1.0.

We used PEFT [29] and DeepSpeed ZeRO [30] for implementation. The code is available at https://github.com/retarfi/jallm.

IV. EVALUATING CONSTRUCTED MODELS

We evaluate the tuned models both quantitatively and qualitatively. From the quantitative perspective, we evaluate from two perspectives. The first is accuracy derived from the likelihood of choices in text classification tasks with JNLI and MARC-ja. JNLI and MARC-ja are tasks from JGLUE [31]. Further details are described in Section IV-A. The second is perplexity using question-answering data that is not included in the dataset constructed in this study. From the qualitative perspective, we qualitatively evaluate the output for several prompts. The temperature parameter for generation is 0.0, and the repetition penalty [32] is 1.05 for CALM and Stormy and 1.0 for Instruct LLaMA 7B and LLaMA 7B. We use 5 prompts for input to the models, which are the same as those used in the literature [25]. We also conduct evaluation experiments on LLaMA 13B and Instruct LLaMA 13B, which was instruction tuned for LLaMA 13B, constructed in the study [25] as well.

A. Accuracy

Another evaluation is performed by JNLI and MARC-ja included in JGLUE [31]. JNLI is a task to choose the relationship that the premise sentence shows to the sentence pair of the hypothesis from three options: entailment, contradiction, and neutral. MARC-ja is a task to choose either "positive" or "negative" in Japanese for product reviews and is constructed using the Japanese part of the Multilingual Amazon Reviews Corpus (MARC) [33]. In addition to these, JGLUE includes JCommonsenseQA, which questions common sense, and JSQuAD, which is an extraction task. However, these data are included in the llm-japanese-dataset v0.1.0, which is used for instruction LoRA tuning. Therefore, they were considered inappropriate as evaluation tasks and excluded.

For the implementation of the experiment, we use the Japanese evaluation branch [5] of Stability-AI/lm-evaluation-harness [34]. Aligning with lm-evaluation-harness, we use the

---

[3]https://huggingface.co/cyberagent/open-calm-7b

[4]Strictly speaking, although it was not initially open-source, it has become available under certain licenses

[5]https://github.com/Stability-AI/lm-evaluation-harness/tree/jp-stable

The calculation of perplexity is not performed on the input to the model and is only applied to the response. In other words, in the above example, perplexity is calculated only for the token corresponding to the output "white."

V. RESULTS AND DISCUSSION

A. Quantitative Evaluation

The results of the evaluation experiment are shown in Table II. In the evaluation by JNLI, the accuracy of Stormy was the highest across 1-shot, 2-shot, and 3-shot settings. Even though the llm-japanese-dataset v0.1.0 does not include a dataset equivalent to implication relation recognition, the performance seems to have been improved by solving various tasks as in [16]. The improvement in performance on tasks not present in the dataset by using instruction tuning across various tasks aligns with the findings in the literature [16], [17]. Japanese instruction datasets are valuable in the point of having constructed similar datasets for languages other than English. The performance of Stormy and Instruct LLaMA 7B, which performed instruction tuning on CALM and LLaMA 7B, respectively, improved, showing the effect of instruction tuning. However, the effect of instruction tuning in LLaMA 13B was relatively small. This is likely because instruction tuning in Instruct LLaMA 13B was performed for only one epoch. Performance might improve with more training epochs. When comparing two Instruct LLaMA models with different numbers of parameters, even though there was a difference in the number of training epochs, Instruct LLaMA 7B showed a stronger effect from instruction tuning. This is considered to be because the smaller model size facilitates more effective training. It has been reported that larger model sizes result in better performance on downstream tasks [7], [10], [13]. However, it is possible that tuning a smaller model might yield better performance under the same computational resources for instruction tuning.

In the evaluation by MARC-ja, there was no performance improvement by instruction tuning in all of 1-shot, 2-shot, and 3-shot, or the performance became worse. This phenomenon has also been reported in [16], [37]. The performance might be improved by adopting more various tasks widely as instruction data as in [16]. As well as MARC-ja, there are also datasets related to sentiment that can be incorporated in Japanese, such as the chABSA-dataset[6] (ABSA stands for Aspect-Based Sentiment Analysis). The decrease in accuracy could be suppressed by additionally training these datasets. Another possible reason why the performance did not improve in the LLaMA-based models is the input length of instruction tuning in this study. While the LLaMA-based model itself can input up to 2,048 tokens and pre-training is performed at this length, in this study, the input length is limited to 256 tokens. Therefore, in data where long tokens are input, the effect of

---

prompt version that achieves the best performance. We adopt v0.2 for Stormy and v0.3 for the others, such as CALM, Instruct LLaMA 7B, LLaMA 7B, Instruct LLaMA 13B, and LLaMA 13B. Detailed prompts are described in the Appendix.

For the input prompt, we compare the likelihood of the strings of each task's choicesand take the highest one as the model's output. In JNLI, the three choices are entailment, contradiction, and neutral, and in MARC-ja, the two choices are "positive" and "negative" in Japanese, and the model outputs the choice with the highest likelihood of output. Therefore, outputs other than the choices are not considered. We evaluate for each of 1-shot, 2-shot, and 3-shot, which show one, two, or three examples in the input, respectively.

B. Perplexity

Perplexity, as defined by [35], is the exponential of the average negative log-likelihood. The lower the value, the higher the probability that the words in the dataset are correctly output. Given a tokenized sequence $X = (x_0, x_1, \cdots, x_t)$, the perplexity of X is represented by Equation (1).

$$\text{Perplexity}(X) = \exp\left\{-\frac{1}{t}\sum_{i}^{t}\log p_\theta(x_i|x_{<i})\right\} \quad (1)$$

Here, $\log p_\theta(x_i|x_{<i})$ is the log-likelihood of the i-th token given the preceding tokens $x_{<i}$.

In this study, we measure perplexity using the Japanese Visual Question Answering (VQA) dataset [36], which is not included in the llm-japanese-dataset v0.1.0 used for tuning the language model. Although this VQA dataset is a question-answering task performed by looking at presented images, it is conjectured that models with a high probability of predicting the correct response sentence are more natural. We convert 793,664 question and answer pairs extracted from the VQA dataset into prompt format and input them. An example of the input is shown below.

┌─── Example in VQA with Japanese-based Model ───
Write a response to answer the following question.

Question:
What color is the airplane's body?

Response:
White [2]

It should be noted that the LLaMA-based model uses English for system messages. Therefore, following the literature [25], the above example is modified as follows.

┌─── Example in VQA with English-based Model ───
Write a response to answer the following question.

Question:
What color is the airplane's body? [2]

---

[6]https://github.com/chakki-works/chABSA-dataset

TABLE II

RESULTS OF THE EVALUATION EXPERIMENT. * INDICATES THAT THERE WERE DATA IN THE EVALUATION DATASET THAT EXCEEDED THE INPUT LENGTH OF LORA TUNING (STORMY IS 300, INSTRUCT LLAMA 7B AND INSTRUCT LLAMA 13B ARE 256). † INDICATES THAT THERE WERE DATA IN THE EVALUATION DATASET THAT EXCEEDED THE MAXIMUM INPUT LENGTH OF THE MODEL (BOTH CALM-BASED AND LLAMA-BASED ARE 2,048). THE HIGHEST-PERFORMING AREAS FOR EACH TASK ARE INDICATED IN BOLD.

Model	JNLI			MARC-ja			VQA
	1-shot	2-shot	3-shot	1-shot	2-shot	3-shot	Perplexity
Stormy (Instruct CALM)	**0.459**	**0.508**	0.475	0.468	0.828*	0.784*	**29.9**
CALM[1]	0.294	0.331	0.314	0.781	0.836	**0.856**	246.6
Instruct LLaMA 7B	0.398*	0.454*	**0.479***†	0.795*	0.829*	0.847*	68.5
LLaMA 7B [13]	0.171	0.273	0.303†	0.839	0.848	0.852	1,499
Instruct LLaMA 13B [25]	0.302*	0.302*	0.302*†	38.8	**0.859***	0.855*	0.855*
LLaMA 13B [13]	0.316	0.281	0.263†	0.855	**0.855**	0.855	971.5

instruction tuning may not have been demonstrated. Extending the input length of instruction tuning is a future issue.

In the evaluation of perplexity using VQA, all the instruct-tuned models showed improved performance with reduced perplexity due to tuning using instruction data. Language models adopting the decoder architecture are trained to increase the probability of correctly predicting the next token in the input. Therefore perplexity is trained to decrease. However, the reason for the reduction in perplexity by instruction tuning might be attributed to differences in the input data. While a language model predicts the next token for consecutive sequences in pre-training, it predicts tokens sequentially in response to a given question in instruction tuning. Since the format of input and output in instruction tuning matches the question-answering in VQA used in this experiment, it can be inferred that the model became more accustomed to producing answers by instruction tuning, leading to a reduction (performance improvement) in perplexity.

The improvement in perplexity was particularly noticeable in the LLaMA-based models. A link with Japanese is considered to have been created and the performance improved by training using instruction data including translation data, even for models other than Japanese, such as English. Among the six models, the one with the highest perplexity and the worst performance was LLaMA 7B. This is thought to be due to the fact that it is an English-based model and has fewer model parameters than LLaMA 13B. On the other hand, the model that showed the best performance with the lowest perplexity was Stormy. The performance was improved by further instruction tuning for CALM, which is a model of Japanese. Comparing CALM, LLaMA 7B, and LLaMA 13B, which were the base models for tuning, the Japanese-based CALM showed the highest performance.

In terms of the effect of instruction tuning from the perspective of model size, the literature [16] reported that for models larger than 68B, the effects of instruction tuning were observed in downstream tasks. However, they also reported for models smaller than 8B, instruction tuning paradoxically degraded performance in downstream tasks. In the results of the MARC-ja experiments in this study, no effect of instruction tuning was observed for all models of 7B and 13B, while for JNLI, the positive effects of instruction tuning were observed

in all models. This effect was observed in both Japanese-based CALM and English-based LLaMA models. This suggests that, in non-English languages or when tuning English models to them, instruction tuning does not necessarily have negative effects for smaller models, and could even contribute to performance enhancement.

in comparison with prior research [16], [38], there are fewer types of tasks. This might have led to a potential constraint in performance improvement. For instance, when compared to FLAN [16], tasks like simplification and correction have been newly added, while tasks like natural language inference, sentiment, and paraphrase lack. In this study, although the experiments were conducted using the 5 task types shown in Figure 1, consistent results were observed even in a non-English language like Japanese. Expanding the variety of tasks will be a challenge for future research.

Regarding the base language of the model, there was no difference in terms of performance trends between the Japanese-based model and the English-based model; performance improved in VQA and JNLI, but not in MARC-ja. The results show that using a Japanese instruction dataset can enhance performance in Japanese tasks for both English-based and Japanese-based models. Except for the fact that translation tasks were used in the instruction tuning data of the LLaMA-derived model, the other tasks used in Stormy and Instruct LLaMA are the same. How the translation task impacts both English-based and Japanese-based models remains a future research topic.

As discussed in the results for JNLI, while instruction tuning is effective for unseen tasks, the inclusion of similar tasks in the dataset is considered to have a substantial impact on performance improvement [16]. Based on this, it can be inferred that by adding tasks related to natural language inference for JNLI and sentiment for MARC-ja, the performance on JNLI and MARC-ja tasks would improve more. The tasks evaluated in this study evaluate how much the model can output for a certain output (although it is consistent with the evaluation method in [6]), and do not consider the actual output of the model. For a more detailed comparison, it is also a future task to evaluate using actual outputs. In addition, it will be necessary in the future to evaluate from the perspective of human preference [39], [40], not just focusing on the task's

correctness.

B. Qualitative Evaluation

The following shows the responses of the models used in this study, such as Stormy, CALM, Instruct LLaMA 7B, and LLaMA 7B. For responses of Instruct LLaMA 13B and LLaMA 13B to the input shown below, please refer to the literature [25].

┌─── Comparison of Responses (1) ───────────────
│ **### Input:**
│ Who are you? [2]
│ **### Output (Stormy):**
│ I am my wife, and I love her. [2]
│ **### Output (CALM):**
│ I am Nick M. Kelemel. I have degrees in computer science and mathematics, and I am currently working as a researcher. [2]
│ **### Output (Instruct LLaMA 7B):**
│ What are you doing? [2]
│ **### Output (LLaMA 7B):**
│ Who are you? [2]
└───

┌─── Comparison of Responses (2) ───────────────
│ **### Input:**
│ Please tell me how to get to Osaka from Tokyo. [2]
│ **### Output (Stormy):**
│ Depart from Tokyo by plane. When you arrive in Osaka, transfer to the Shinkansen and go to Nagoya. From Nagoya, take the Kintetsu Limited Express. [2]
│ **### Output (CALM):**
│ Which is better, train or Shinkansen?
│ * "Train" refers to all types of transportation, such as JR trains and private railways.
│ * "Shinkansen" refers to the Tokaido and Sanyo Shinkansen that run between Tokyo and Shin-Osaka.
│ * "Airplane" typically refers to the route between Haneda and Itami.
│ * "Bus" refers to the Toei buses that operate in Tokyo.
│ * "Car" includes not only private cars but also taxis and rental cars.
│ * "Bicycle" includes electric-assisted bicycles.
│ * "Walking," "jogging," and "strolling" are excluded. [2]
│ **### Output (Instruct LLaMA 7B):**
│ Please tell me how to get to Tokyo from Osaka. [2]
│ **### Output (LLaMA 7B):**
│ Please tell me how to get to Osaka from Tokyo. [2]
└───

┌─── Comparison of Responses (3) ───────────────
│ **### Input:**
│ Where is the University of Tokyo? [2]
│ **### Output (Stormy):**
│ The University of Tokyo is a national university located in
└───

Bunkyo-ku, Tokyo. [2]
Output (CALM):
The University of Tokyo is located at 7-3-1 Hongo, Bunkyo-ku, Tokyo. The latitude and longitude of the University of Tokyo are 35.676044, 135.869422.

##Answer: What do you use to get to that address? (Maps or Google Map, etc.)

##Explanation: What mode of transportation should I use to get to that address?

##Answer: Which is better to use, train or bus?

##Explanation: If you use the JR line, please get off at the "Yamanote Line." The JR line goes to "Tokyo" station.

##Answer: Where should I go from Tokyo station?

##Explanation: The trains you take from "Tokyo Station" are the "Tokaido Shinkansen," "Tohoku, Joetsu, Hokuriku Shinkansen" and "Limited Express Hitachi."

##Answer: Which is faster and cheaper, the Joban Line or the Shinkansen?

##Explanation: The Shinkansen is faster. However, the price is different.

##Answer: I don't have money, what should I do?

##Explanation: First, please prepare money. If you have a credit card, you can use it.

##Answer: How should I pay?

##Explanation: You can pay by card. VISA card [2]
Output (Instruct LLaMA 7B):
The University of Tokyo is in Tokyo. [2]
Output (LLaMA 7B):
Where is the University of Tokyo? [2]
(Authors' note: The correct address is 7-3-1 Hongo, Bunkyo-ku, Tokyo.)

┌─── Comparison of Responses (4) ───────────────
│ **### Input:**
│ The phone is ringing. What should I do? [2]
│ **### Output (Stormy):**
│ Should you answer the phone? Or should you put it on hold? [2]
└───

Output (CALM):
Press the hold button, answer the phone, or call the answering service. When you hear the ringtone, reply within 10 seconds. If there is no response, try calling again. If that still doesn't work, proceed to the next step.
* Press '0' to end the call.
* Press '*' after the ringtone. This will enable the call. [2]
Output (Instruct LLaMA 7B):
The phone is ringing. What should I do? [2]
Output (LLaMA 7B):
The phone is ringing. What should I do? [2]

— Comparison of Responses (5) —
Input:
What are the three major festivals in Kyoto? [2]
Output (Stormy):
Aoi Festival, Gion Festival, Jidai Festival [2]
Output (CALM):
The three are Gion Festival, Aoi Festival, and Jidai Festival. They are also referred to as the three major festivals of Kyoto. [2]

#N/A
#N/A
(Authors' Note: " #N/A" continues from here.)
Output (Instruct LLaMA 7B):
What are the three major festivals in Kyoto? [2]
Output (LLaMA 7B):
What are the three major festivals in Kyoto? [2]
(Authors' note: The correct answer is Aoi, Gion, Jidai festivals.)

Stormy, which was instruct-tuned from CALM, demonstrates a more natural response format. On the other hand, the difference between Instruct LLaMA 7B and LLaMA 7B appears to be minimal, with only a slight improvement in the accuracy of response comparison (3). According to the literature [25], the small improvement in LLaMA 13B's 1 epoch tuning was attributed to the limited amount of training. However, despite 5 epochs of tuning on LLaMA 7B, Instruct LLaMA 7B did not show significant improvement. This result contrasts with the trend observed in the quantitative evaluations discussed in Section V-A. This difference suggests that instruction tuning alone may not lead to significant improvements. Not only through instruction tuning, but also by additional pre-training to accumulate knowledge about the target language (in this case, Japanese) and then performing instruction tuning, there is a potential to improve performance in the target language [41]. Various methods can be considered for pre-training, including pre-training from scratch solely in Japanese or conducting additional pre-training in Japanese using English or multilingual models. Hence, exploring methods to achieve high performance in languages other than English

will be a future challenge. Moreover, the difference between the trend observed in the qualitative evaluation and that of the quantitative evaluation highlights the importance of evaluating qualitative performance not only based on simple metrics like likelihood or perplexity generated by the model but also based on the actual output strings.

VI. RELATED WORK

A. Pre-training Models

The Transformer [1], a crucial component of large language models (LLMs), consists of two architectures: encoder and decoder. Models such as BERT [2], RoBERTa [3], and DeBERTa [42], [43] utilize the Transformer's encoder. In recent LLMs, the decoder (causal decoder) is primarily used. The GPT series [4]–[6], [44] is a representative language model with a decoder architecture. There are also many proposed models, such as OPT [7], GPT-NeoX-20B [8], Gopher [45], PaLM [10], BLOOM [11], Pythia [12], and LLaMA series [13], [14]. Although relatively few, there is also T5 [46] as an encoder-decoder architecture, and Flan T5 [16] has been proposed as an extension of it.

B. Tuning after Pre-training

Through pre-training, LLMs acquire the ability to solve various tasks. However, there is an increasing number of cases where fine-tuning is performed to align more with specific purposes, such as dialogue responses. There are mainly two approaches for this fine-tuning: alignment tuning and instruction tuning. Alignment tuning is a tuning to aim for outputs more in line with human preferences [37]. Through alignment tuning, LLMs are adjusted to produce outputs that align with human values, such as being helpful, honest, and harmless. In recent LLMs, reinforcement learning (RL) [47], especially Proximal Policy Optimization (PPO) [48], is used to learn from human-labeled response preference rankings. Instruction tuning is generally a method of fine-tuning LLMs with datasets in natural language format, supervised [37], and performed using various tasks (multi-task) [17]. It has been shown that instruction tuning can demonstrate excellent performance even for unknown tasks [16], [17], [19]. In instruction tuning, instructions that request the output of the LLM are described in the input, and training is performed so that the LLM outputs the expected response. It has been shown that the task-explanation part is particularly decisive in the performance of the LLM that improves through instruct tuning [16]. If tuning is performed with a dataset that is closer to traditional supervised learning by removing the explanation about the task, the performance will significantly decrease compared to not removing it. Most cases of instruct tuning are performed with formatted existing datasets for various tasks in natural language format for instruction tuning datasets. Specifically, labeled datasets are applied with instructions written by humans explaining the task, explaining the direction of the output, and instructing the LLM to understand the task from the input [16], [17], [38]. Other construction methods include examples of constructing datasets using the output

of ChatGPT or GPT-4 [22], [39], [49][7],[8], and there are few examples of constructing datasets manually [20].

C. Tuning of LLMs

Efficient tuning in LLMs with many parameters is attracting attention to adapt LLMs to various downstream tasks. In particular, LoRA [28] is widely applied to open-source LLMs. For example, Alpaca-LoRA [50] uses LoRA to tune LLaMA 7B as a lightweight tuning version of Alpaca [22]. Also, AdaLoRA [51] changes the value of the rank in LoRA. This adjustment occurs according to the layer to be applied.

Other efficient tuning methods include adding an Adapter layer to the existing layers [52]–[54], and prompt tuning [55], [56], which fixes the weights of the pre-trained model, adds trainable parameters to the prompt instructions.

VII. CONCLUSION

We constructed an instruction dataset for Japanese-based LLMs (Large Language Models). The dataset excludes any translation data originally present in the llm-japanese-dataset and introduces additional tasks to the existing ones. We performed LoRA tuning on LLMs pre-trained in both Japanese and English, respectively. The tuning was done using Japanese instruction data. We evaluated the tuned models from both quantitative and qualitative perspectives. The results show that tuning with Japanese instruction data improves performance in quantitative evaluations. In particular, the results indicate that not only Japanese-based models but also English-based models can be tuned in Japanese using the Japanese instruction dataset. Furthermore, even with smaller model sizes like 7B or 13B, instruction tuning can sometimes improve performance in downstream tasks, suggesting a result different from prior research.

Future research can address not only comparing the likelihood of the current model's output, but also using the actual output in the evaluation of the model. Additionally, it could include evaluation from the perspective of human preference in Japanese.

ACKNOWLEDGMENT

REFERENCES

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention Is All You Need," in *Advances in Neural Information Processing Systems*, vol. 30, 2017, pp. 5999–6009.

[2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics.* Association for Computational Linguistics, 2019, pp. 4171–4186.

[3] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, and P. G. Allen, "RoBERTa: A Robustly Optimized BERT Pretraining Approach," 2019. [Online]. Available: https://arxiv.org/abs/1907.11692

[4] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving Language Understanding by Generative Pre-Training," 2018. [Online]. Available: https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf

[5] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language Models are Unsupervised Multitask Learners," 2019. [Online]. Available: https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf

[6] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language Models are Few-Shot Learners," *Advances in Neural Information Processing Systems*, vol. 33, pp. 1877–1901, 2020.

[7] S. Zhang, S. Roller, N. Goyal, M. Artetxe, M. Chen, S. Chen, C. Dewan, M. Diab, X. Li, X. V. Lin *et al.*, "OPT: Open Pre-trained Transformer Language Models," 2022. [Online]. Available: https://arxiv.org/abs/2205.01068

[8] S. Black, S. Biderman, E. Hallahan, Q. Anthony, L. Gao, L. Golding, H. He, C. Leahy, K. McDonell, J. Phang *et al.*, "GPT-NeoX-20B: An Open-Source Autoregressive Language Model," in *Proceedings of BigScience Episode #5 – Workshop on Challenges & Perspectives in Creating Large Language Models.* Association for Computational Linguistics, 2022, pp. 95–136.

[9] Y. Tay, M. Dehghani, V. Q. Tran, X. Garcia, J. Wei, X. Wang, H. W. Chung, D. Bahri, T. Schuster, S. Zheng *et al.*, "UL2: Unifying Language Learning Paradigms," in *International Conference on Learning Representations*, 2023.

[10] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann *et al.*, "PaLM: Scaling Language Modeling with Pathways," 2022. [Online]. Available: https://arxiv.org/abs/2204.02311

[11] T. L. Scao, A. Fan, C. Akiki, E. Pavlick, S. Ilić, D. Hesslow, R. Castagné, A. S. Luccioni, F. Yvon, M. Gallé *et al.*, "BLOOM: A 176B-Parameter Open-Access Multilingual Language Model," 2022. [Online]. Available: https://arxiv.org/abs/2211.05100

[12] S. Biderman, H. Schoelkopf, Q. Anthony, H. Bradley, K. O'Brien, E. Hallahan, M. A. Khan, S. Purohit, U. S. Prashanth, E. Raff *et al.*, "Pythia: A Suite for Analyzing Large Language Models Across Training and Scaling," 2023. [Online]. Available: https://arxiv.org/abs/2304.01373

[13] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar *et al.*, "LLaMA: Open and Efficient Foundation Language Models," 2023. [Online]. Available: https://arxiv.org/abs/2302.13971

[14] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale *et al.*, "Llama 2: Open foundation and fine-tuned chat models," 2023. [Online]. Available: https://arxiv.org/abs/2307.09288

[15] J. Wei, Y. Tay, R. Bommasani, C. Raffel, B. Zoph, S. Borgeaud, D. Yogatama, M. Bosma, D. Zhou, D. Metzler *et al.*, "Emergent abilities of large language models," *Transactions on Machine Learning Research*, 2022.

[16] J. Wei, M. Bosma, V. Zhao, K. Guu, A. W. Yu, B. Lester, N. Du, A. M. Dai, and Q. V. Le, "Finetuned language models are zero-shot learners," in *International Conference on Learning Representations*, 2022.

[17] V. Sanh, A. Webson, C. Raffel, S. Bach, L. Sutawika, Z. Alyafeai, A. Chaffin, A. Stiegler, A. Raja, M. Dey *et al.*, "Multitask Prompted Training Enables Zero-Shot Task Generalization," in *International Conference on Learning Representations*, 2022.

[18] A. Köpf, Y. Kilcher, D. von Rütte, S. Anagnostidis, Z.-R. Tam, K. Stevens, A. Barhoum, N. M. Duc, O. Stanley, R. Nagyfi *et al.*, "Openassistant conversations – democratizing large language model alignment," 2023. [Online]. Available: https://arxiv.org/abs/2304.07327

[19] H. W. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, Y. Li, X. Wang, M. Dehghani, S. Brahma *et al.*, "Scaling Instruction-Finetuned Language Models," 2022. [Online]. Available: https://arxiv.org/abs/2210.11416

[20] Databricks, "Dolly," https://github.com/databrickslabs/dolly, 2023.

[21] Vicuna, "Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90% ChatGPT Quality," https://vicuna.lmsys.org/, 2023.

[22] R. Taori, I. Gulrajani, T. Zhang, Y. Dubois, X. Li, C. Guestrin, P. Liang, and T. B. Hashimoto, "Stanford Alpaca: An Instruction-following LLaMA model," https://github.com/tatsu-lab/stanford_alpaca, 2023.

[23] G. Zhang, Y. Shi, R. Liu, R. Yuan, Y. Li, S. Dong, Y. Shu, Z. Li, Z. Wang, C. Lin, W. Huang, and J. Fu, "Chinese

[7] https://huggingface.co/datasets/RyokoAI/ShareGPT52K
[8] https://github.com/teknium1/GPTeacher

open instruction generalist: A preliminary release," 2023. [Online]. Available: https://arxiv.org/abs/2304.07987

[24] Y. Cui, Z. Yang, and X. Yao, "Efficient and effective text encoding for chinese llama and alpaca," 2023. [Online]. Available: https://arxiv.org/abs/2304.08177

[25] M. Hirano, M. Suzuki, and H. Sakaji, "llm-japanese-dataset v0: Construction of Japanese Chat Dataset for Large Language Models and its Methodology," 2023. [Online]. Available: https://arxiv.org/abs/2305.12720

[26] Y. Tanaka, Y. Murawaki, D. Kawahara, and S. Kurohashi, "Improving a Japanese Typo Dataset and Typo Correction System Based on Wikipedia's Revision History," in *Proceedings of the Twenty-seventh Annual Meeting of the Association for Natural Language Processing*, 2021, pp. 1540–1545, (in Japanese). [Online]. Available: https://www.anlp.jp/proceedings/annual_meeting/2021/pdf_dir/E8-3.pdf

[27] H. Tanioka, K. Kimura, K. Takaoka, R. Nakatani, and Y. Uchida, "Automatic Generation of Japanese Question-Answering Pairs," in *Fourth Asia Pacific Corpus Linguistics Conference (APCLC 2018)*, 2018.

[28] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-Rank Adaptation of Large Language Models," in *International Conference on Learning Representations*, 2022, pp. 1–13.

[29] S. Mangrulkar, S. Gugger, L. Debut, Y. Belkada, and S. Paul, "PEFT: State-of-the-art Parameter-Efficient Fine-Tuning methods," https://github.com/huggingface/peft, 2022.

[30] S. Rajbhandari, J. Rasley, O. Ruwase, and Y. He, "ZeRO: Memory Optimizations toward Training Trillion Parameter Models," in *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, 2020, pp. 1–16.

[31] K. Kurihara, D. Kawahara, and T. Shibata, "JGLUE: Japanese General Language Understanding Evaluation," in *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, 2022, pp. 2957–2966.

[32] N. S. Keskar, B. McCann, L. R. Varshney, C. Xiong, and R. Socher, "CTRL: A Conditional Transformer Language Model for Controllable Generation," 2019. [Online]. Available: http://arxiv.org/abs/1909.05858

[33] P. Keung, Y. Lu, G. Szarvas, and N. A. Smith, "The Multilingual Amazon Reviews Corpus," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, 2020.

[34] L. Gao, J. Tow, S. Biderman, S. Black, A. DiPofi, C. Foster, L. Golding, J. Hsu, K. McDonell, N. Muennighoff *et al.*, "A framework for few-shot language model evaluation," 2021. [Online]. Available: https://doi.org/10.5281/zenodo.5371628

[35] F. Jelinek, R. L. Mercer, L. R. Bahl, and J. K. Baker, "Perplexity—a measure of the difficulty of speech recognition tasks," *The Journal of the Acoustical Society of America*, vol. 62, no. S1, pp. S63–S63, 1977.

[36] N. Shimizu, N. Rong, and T. Miyazaki, "Visual Question Answering Dataset for Bilingual Image Understanding: A Study of Cross-Lingual Transfer Using Attention Maps," in *Proceedings of the 27th International Conference on Computational Linguistics*. Association for Computational Linguistics, 2018, pp. 1918–1928.

[37] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Gray *et al.*, "Training language models to follow instructions with human feedback," in *Advances in Neural Information Processing Systems*, A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho, Eds., 2022.

[38] Y. Wang, S. Mishra, P. Alipoormolabashi, Y. Kordi, A. Mirzaei, A. Naik, A. Ashok, A. S. Dhanasekaran, A. Arunkumar, D. Stap *et al.*, "Super-NaturalInstructions: Generalization via Declarative Instructions on 1600+ NLP Tasks," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2022, pp. 5085–5109.

[39] B. Peng, C. Li, P. He, M. Galley, and J. Gao, "Instruction Tuning with GPT-4," 2023. [Online]. Available: https://arxiv.org/abs/2304.03277

[40] C. Zhou, P. Liu, P. Xu, S. Iyer, J. Sun, Y. Mao, X. Ma, A. Efrat, P. Yu, L. Yu *et al.*, "LIMA: Less Is More for Alignment," 2023. [Online]. Available: https://arxiv.org/abs/2305.11206

[41] Z.-X. Yong, H. Schoelkopf, N. Muennighoff, A. F. Aji, D. I. Adelani, K. Almubarak, M. Saiful Bari, L. Sutawika, J. Kasai, A. Baruwa *et al.*, "BLOOM+1: Adding language support to BLOOM for Zero-Shot prompting," 2022. [Online]. Available: https://arxiv.org/abs/2212.09535

[42] P. He, X. Liu, J. Gao, and W. Chen, "DeBERTa: Decoding-enhanced BERT with Disentangled Attention," in *International Conference on Learning Representations*, 2021.

[43] P. He, J. Gao, and W. Chen, "DeBERTaV3: Improving DeBERTa using ELECTRA-Style Pre-Training with Gradient-Disentangled Embedding Sharing," in *International Conference on Learning Representations*, 2023.

[44] OpenAI, "GPT-4 Technical Report," 2023. [Online]. Available: https://arxiv.org/abs/2303.08774

[45] J. W. Rae, S. Borgeaud, T. Cai, K. Millican, J. Hoffmann, F. Song, J. Aslanides, S. Henderson, R. Ring, S. Young *et al.*, "Scaling Language Models: Methods, Analysis & Insights from Training Gopher," 2022. [Online]. Available: https://arxiv.org/abs/2112.11446

[46] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer," *Journal of Machine Learning Research*, vol. 21, no. 140, pp. 1–67, 2020.

[47] P. F. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, and D. Amodei, "Deep reinforcement learning from human preferences," in *Advances in neural information processing systems*, vol. 30, 2017, pp. 4299–4307.

[48] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," 2017. [Online]. Available: https://arxiv.org/abs/1707.06347

[49] S. Chaudhary, "Code Alpaca: An Instruction-following LLaMA model for code generation," https://github.com/sahil280114/codealpaca, 2023.

[50] E. Wang, "Alpaca-LoRA," https://github.com/tloen/alpaca-lora, 2023.

[51] Q. Zhang, M. Chen, A. Bukharin, P. He, Y. Cheng, W. Chen, and T. Zhao, "Adaptive budget allocation for parameter-efficient fine-tuning," in *International Conference on Learning Representations*, 2023.

[52] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. de Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly, "Parameter-efficient transfer learning for nlp," 2019. [Online]. Available: https://arxiv.org/abs/1902.00751

[53] Z. Lin, A. Madotto, and P. Fung, "Exploring Versatile Generative Language Model Via Parameter-Efficient Transfer Learning," in *Findings of the Association for Computational Linguistics: EMNLP 2020*. Online: Association for Computational Linguistics, Nov. 2020, pp. 441–459.

[54] Z. Hu, Y. Lan, L. Wang, W. Xu, E.-P. Lim, R. K.-W. Lee, L. Bing, X. Xu, and S. Poria, "LLM-Adapters: An Adapter Family for Parameter-Efficient Fine-Tuning of Large Language Models," 2023. [Online]. Available: https://arxiv.org/abs/2304.01933

[55] X. L. Li and P. Liang, "Prefix-tuning: Optimizing continuous prompts for generation," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, 2021, pp. 4582–4597.

[56] B. Lester, R. Al-Rfou, and N. Constant, "The power of scale for parameter-efficient prompt tuning," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2021, pp. 3045–3059.

APPENDIX

Prompts Used in JNLI and MARC-ja

Example of 1-shot prompt used in JNLI (v0.2)

Please answer the relationship between the premise and the hypothesis from[2] entailment, contradiction, and[2] neutral.

Constraints:
- If the hypothesis can be derived from the premise using logical or common sense knowledge, output[2] entailment
- If the premise and the hypothesis are incompatible, output[2] contradiction
- If neither of the above, output[2] neutral

Premise: Two women are jumping to catch a frisbee in the grass.
Hypothesis: The two women are holding a tray with donuts on it.
Relationship:[2] entailment

Premise: There are two children, and bananas and kiwis are placed next to the mixer.
Hypothesis: There are children with droppers at the table where the mixer is placed.
Relationship: [2]

──── Example of 1-shot prompt used in JNLI (v0.3) [9] ────

Below is a combination of instructions explaining the task and contextual inputs. Write a response that adequately meets the request.

Instructions:
Please answer the relationship between the given premise and hypothesis.

Choose your output from the following: [2]
entailment
contradiction
neural

Input:
Premise: Two women are jumping to catch a frisbee in the grass.
Hypothesis: The women are trying to catch a frisbee.

Response: [2]
entailment

Instructions:
Please answer the relationship between the given premise and hypothesis.

Choose your output from the following: [2]
entailment
contradiction
neural

Input:
Premise: There are two children, and bananas and kiwis are placed next to the mixer.
Hypothesis: There are children with droppers at the table where the mixer is placed.

Response:
[2]
-

──── Example of 1-shot prompt used in MARC-ja (v0.2) ────

Please classify the product review into either negative or positive sentiment. Please lowercase the output. [2]

Product Review: I like country and initially thought of buying the CD, the movie has a decent story, it's okay
Sentiment:[2] positive

Product Review: I enjoyed it till the end. Personally, I wanted to see more dance scenes. I hope it will be staged.
Sentiment: [2]

──── Example of 1-shot prompt used in MARC-ja (v0.3) [9] ────

Below is a combination of instructions explaining the task and contextual inputs. Write a response that adequately meets the request.

Instructions:
Please classify the following product review into either a positive or negative sentiment class.

Input:
I like country and initially thought of buying the CD, the movie has a decent story, it's okay

Response:
Positive

Instructions:
Please classify the following product review into either a positive or negative sentiment class.

Input:
I enjoyed it till the end. Personally, I wanted to see more dance scenes. I hope it will be staged.

Response:
[2]
-

---

[9]Although the same instructions are repeated within the same prompt, the template from the implementation reference (https://github.com/Stability-AI/lm-evaluation-harness/tree/jp-stable) is used as is.