Scaling Data-Constrained Language Models with Synthetic Data

Hirokazu Kiyomaru[1], Yusuke Oda[1], Takashi Kodama[1],
Chaoran Liu[1], Daisuke Kawahara[1,2]
[1]Research and Development Center for LLMs, National Institute of Informatics
[2]Waseda University
{kiyomaru, odashi, tkodama, cliu}@nii.ac.jp dkw@waseda.jp

Abstract

Large language models (LLMs) improve with more training data, but practical limits on data collection increasingly constrain further scaling. Advances in instruction-following LLMs have enabled controlled, high-quality text generation, making synthetic data a potential remedy. However, its effectiveness for pre-training non-English LLMs remains underexplored. We study this question for Japanese in a fixed token budget setting in which organic Japanese Web text constitutes only a small share, while far more organic English Web text and instruction-following LLMs capable of generating fluent Japanese are available. We compare three strategies to fill the data shortfall: generating synthetic Japanese text, repeating the limited Japanese Web text, and using English Web text. Experiments show that synthetic Japanese corpora outperform both baselines and approach the performance achieved when the entire token budget is filled with additional organic Japanese Web text. [1]

1 Introduction

Large language models (LLMs) improve with more training data (Kaplan et al., 2020). This scaling law has driven LLM development on ever-larger corpora, now surpassing 10 trillion tokens (Yang et al., 2025; Grattafiori et al., 2024; Gemma Team et al., 2025; DeepSeek-AI et al., 2024).

However, further scaling is increasingly constrained by practical data collection limits (Villalobos et al., 2024; DatologyAI, 2025). Because performance improves logarithmically with corpus size, achieving further gains requires exponentially more data (Kaplan et al., 2020), making additional corpus expansion progressively less practical.

For non-English languages, this problem is even more challenging. Aggregating the whole Common Crawl snapshots[2] yields only several hundred billion tokens for comparatively well-represented languages such as Chinese, German, and Japanese (Penedo et al., 2025), far below the multi-trillion-token scales.

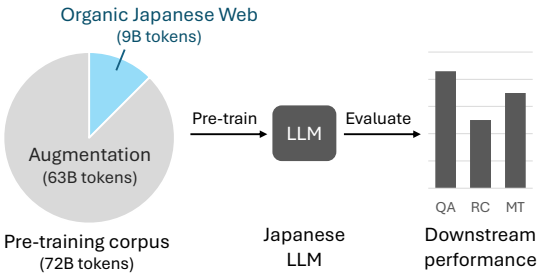


Figure 1: Our experimental setup. We pre-train Japanese LLMs under a fixed token budget (72B tokens) in which organic Japanese Web text constitutes only a small share (9B tokens out of 72B tokens). We compare different data augmentation strategies through evaluating Japanese downstream performance.

As a solution to this problem, synthetic data has emerged as a potential remedy (Gunasekar et al., 2023; Abdin et al., 2024; Maini et al., 2024; Cheng et al., 2024; Jiang et al., 2024; Qin et al., 2025; DatologyAI, 2025; Nguyen et al., 2025). Advances in instruction-following LLMs have enabled controlled, high-quality text generation, supporting the synthesis of large-scale, diverse text corpora that serve as effective resources for LLM pre-training. Recent work has shown that paraphrasing organic Web documents into multiple styles is an effective data augmentation to improves downstream performance (Maini et al., 2024; DatologyAI, 2025; Nguyen et al., 2025; Wang et al., 2025b). Appending synthetic document-grounded question–answer pairs to organic Web documents has also been found to boost downstream performance (Cheng et al., 2024; Jiang et al., 2024).

Despite these gains, previous studies have fo-

[1]Our synthetic Japanese corpus is available at https://huggingface.co/datasets/llm-jp/scaling-data-constrained-llms.

[2]https://www.commoncrawl.org/

Findings of the Association for Computational Linguistics: EACL 2026, pages 1002–1016
March 24-29, 2026

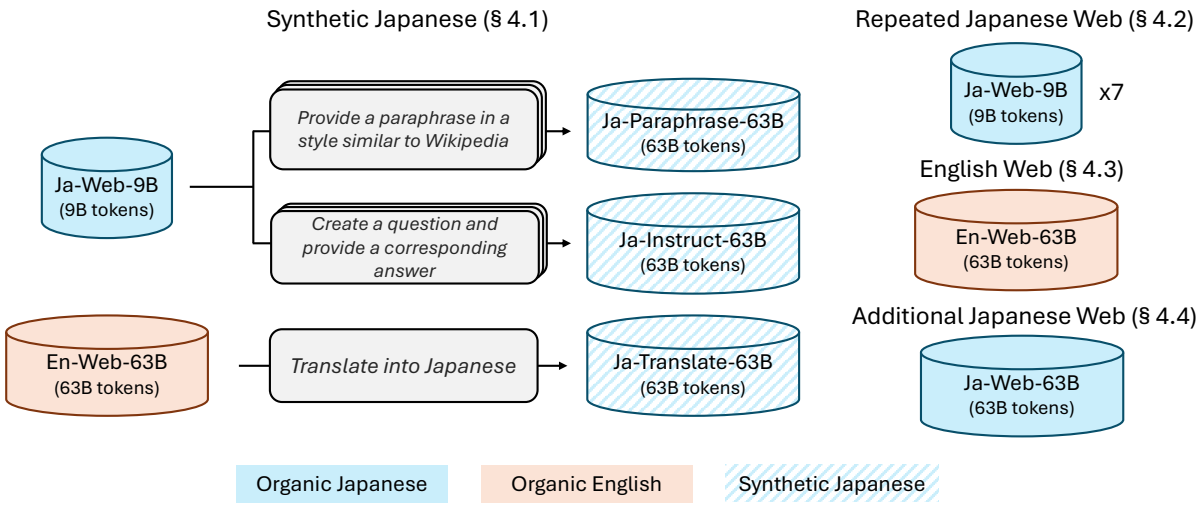


Figure 2: Overview of the data augmentation strategies we compare in this study. Starting from a constrained organic Japanese Web corpus (JA-WEB-9B) and a far larger organic English Web corpus (EN-WEB-63B), we construct three types of synthetic Japanese corpora: JA-PARAPHRASE-63B by generating paraphrases of JA-WEB-9B, JA-INSTRUCT-63B by producing document-grounded question–answer pairs from JA-WEB-9B, and JA-TRANSLATE-63B by translating EN-WEB-63B into Japanese. As baselines, we consider repeated use of JA-WEB-9B and direct use of EN-WEB-63B. In addition, as an idealized setting, we also evaluate a setting where the whole augmentation budget is filled with additional organic Japanese Web text (JA-WEB-63B). At training, each augmentation is combined with JA-WEB-9B to form a pre-training corpus of 72B tokens.

cused primarily on English, leaving open whether similar benefits hold for languages with less abundant Web text. This gap motivates our investigation of synthetic data for pre-training non-English LLMs.

In this work, we focus on Japanese as a non-English language and examine pre-training from scratch under a fixed token budget where organic Japanese Web text constitutes only a small fraction of the token budget, whereas far more organic English Web text and off-the-shelf LLMs capable of generating fluent Japanese are readily accessible. Japanese serves as a suitable testbed for several reasons. Japanese is a medium-resource language with substantial societal and commercial demand yet far less high-quality, deduplicated Web text than English. Its typology (e.g., mixed scripts, rich morphology, and relatively free word order) differs distinctly from English, providing a test of whether synthetic pipelines generalize beyond potentially English-centric assumptions. In addition, there exist mature evaluation suites in Japanese that enable reproducible performance measurement. Figure 1 illustrates our experimental setup. Our central question is how best to compose the pre-training corpus mixture to maximize performance on downstream tasks in Japanese.

We compare three strategies to compensate for the data shortage: generating synthetic Japanese text, repeating the limited organic Japanese Web text, and using organic English Web text. For synthetic Japanese text, we construct three different corpora: paraphrases of the limited Japanese Web text (Maini et al., 2024; DatologyAI, 2025; Nguyen et al., 2025; Wang et al., 2025b), document-grounded instruction data derived from the limited Japanese Web text (Cheng et al., 2024; Jiang et al., 2024), and Japanese translation of the English Web text (Wang et al., 2025a). Figure 2 illustrates an overview.

Our experiments show that synthetic Japanese corpora generated via paraphrasing and instruction-formatting not only outperform baselines that either repeat scarce Japanese Web text or directly incorporate English Web text but also match or even surpass the performance achieved when the entire token budget is filled with additional organic Japanese Web text. Our findings support synthetic data as an effective strategy for scaling under data scarcity.

2 Related Work

2.1 Scaling Law

The performance of LLMs follows empirical scaling laws, increasing logarithmically with model size, corpus size, and compute budget (Kaplan

et al., 2020). Scaling corpus size while keeping model size tractable is now a well-established strategy for training practically useful LLMs (Touvron et al., 2023; Grattafiori et al., 2024). However, since advancing along the scaling curve demands exponentially more data, practical limits on data collection increasingly constrain further scaling (Villalobos et al., 2024; DatologyAI, 2025).

Previous work has explored multi-epoch training as a simple remedy for data scarcity (Muennighoff et al., 2023; Xue et al., 2023). However, these studies find that repeating the same corpus only partially compensates for data scarcity; as the number of epochs increases, marginal gains diminish, and the risks of memorization and overfitting rise. These observations motivate increasing unique tokens rather than additional duplicates. Consequently, there is growing interest in LLM-generated synthetic data, which can produce high-quality textual content at scale.

2.2 Synthetic Data for LLM Pre-training

Synthetic data has received increasing attention as a means to break the data bottleneck in LLM pre-training (Gunasekar et al., 2023; Abdin et al., 2024; Maini et al., 2024; Cheng et al., 2024; Jiang et al., 2024; Qin et al., 2025; DatologyAI, 2025; Nguyen et al., 2025). This growing interest is largely driven by advances in instruction-following LLMs, which have enabled controlled and high-quality text generation, making synthetic data a practical resource for pre-training.

One representative strategy for synthesizing LLM pre-training data is to paraphrase existing documents (Maini et al., 2024; DatologyAI, 2025; Nguyen et al., 2025; Wang et al., 2025b). Previous work has shown that LLM-based paraphrasing of Web text can improve pre-training efficiency (Maini et al., 2024). Pre-training on paraphrases, rather than reusing identical surface forms, reduces rote memorization and encourages learning of latent semantic structure, improving generalization (Allen-Zhu and Li, 2023).

Another effective strategy is to augment each document with synthesized question–answer pairs grounded in that document, yielding document-grounded instruction data (Cheng et al., 2024; Jiang et al., 2024). This strategy also emphasizes semantic understanding over memorization by requiring LLMs to answer questions grounded in the source documents (Allen-Zhu and Li, 2023).

For non-English languages, translation-based synthesis serves as an effective strategy, where abundant English resources are translated into the target language (Wang et al., 2025a; Doshi et al., 2024). Wang et al. (2025a) demonstrate the effectiveness of pre-training LLMs on translations produced from FineWeb-Edu (Penedo et al., 2024), a high-quality English Web corpus.

3 Problem Setting

Figure 1 shows an overview. Mirroring typical constraints in non-English LLM development, we study Japanese LLM pre-training from scratch under a fixed token budget, where Japanese Web text constitutes only a small fraction of the budget, while far more English Web text and off-the-shelf LLMs capable of fluent Japanese are accessible. Our goal is to determine corpus mixtures that maximize performance on Japanese downstream tasks.

We experiment with three model sizes: 1.8B, 3.8B, and 7.2B parameters. While these models are modest in size compared to current state-of-the-art LLMs (Yang et al., 2025; Grattafiori et al., 2024; Gemma Team et al., 2025; DeepSeek-AI et al., 2024), they still deliver meaningful performance on Japanese downstream tasks[3] while enabling systematic exploration across a broad range of experimental settings within a practical compute budget.

We pre-train each model on a fixed budget of 72B tokens. According to Hoffmann et al. (2022), this budget corresponds to approximately $2\times$, $1\times$, and $0.5\times$ the compute-optimal pre-training token counts for the 1.8B-, 3.8B-, and 7.2B-parameter models, respectively We assume access to a 9B-token organic Japanese Web corpus (JA-WEB-9B) and a 63B-token organic English Web corpus (EN-WEB-63B). For all experimental settings, we draw 9B tokens from JA-WEB-9B and allocate the remaining 63B tokens to candidate sources detailed in Section 4.

4 Augmentation Strategies

To address the data shortfall, we examine augmentation with synthetic Japanese text (§ 4.1). We compare this approach against two baselines: repeated sampling of limited Japanese Web corpus (§4.2) and direct use of English Web text (§ 4.3). For reference, we also consider an idealized setting that fills the shortfall with additional organic

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

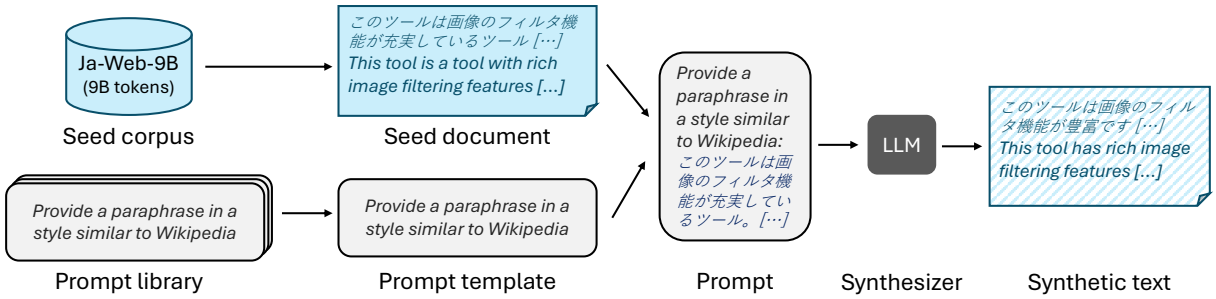


Figure 3: Our synthesis pipeline for JA-PARAPHRASE-63B. A seed document is sampled from the seed corpus and paired with a prompt template sampled from the prompt library. They are combined into a concrete prompt. Conditioned on the prompt, the synthesizer generates synthetic Japanese text. The same procedure applies to other synthetic corpora by varying the seed corpus and/or the prompt library.

Japanese Web text (§4.4). Figure 2 illustrates an overview.

4.1 Synthetic Japanese Text

Figure 3 illustrates our pipeline to generate synthetic Japanese text. In order to build a synthetic corpus, we determine the seed corpus and the prompt library, a collection of prompt templates. For each synthetic text instance, we randomly sample a seed document from the seed corpus and a prompt template from the prompt library, insert the document into the prompt template to form a concrete prompt, and query the synthesizer to generate Japanese text conditioned on that prompt. We repeat this procedure until the required budget of 63B tokens is attained.

By varying the seed corpus and the prompt library, we construct three synthetic corpora introduced below. Across all synthetic corpora, we fix the synthesizer and its decoding parameters (e.g., temperature) so that observed performance differences are attributed to the generation procedure rather than synthesizer variation.

4.1.1 JA-PARAPHRASE-63B

We construct a synthetic corpus (JA-PARAPHRASE-63B) by paraphrasing the limited organic Japanese Web corpus, JA-WEB-9B. We follow Maini et al. (2024) and paraphrase existing Japanese documents into the following four styles: (1) Easy: text that even a toddler will understand, (2) Medium: text such as that found on Wikipedia, (3) Hard: terse and abstruse text, and (4) QA: text in a conversational question-answer (QA) format. We employ one prompt template per style, yielding a prompt library with four prompt templates. The prompt templates can be found in Appendix A.1.

4.1.2 JA-INSTRUCT-63B

We build a synthetic corpus (JA-INSTRUCT-63B) by augmenting each document in JA-WEB-9B with synthetic, document-grounded QA pairs. Following Cheng et al. (2024), we instantiate four QA formats: (1) Free-form: a document-grounded question with an open-ended answer, (2) Multiple-choice: a document-grounded question with candidate options and a selected answer, (3) Free-form + CoT: a free-form answer accompanied by a rationale (chain-of-thought; (Wei et al., 2022)), and (4) Multiple-choice + CoT: a multiple-choice answer accompanied by a rationale. Synthesized QA pairs are appended to the original documents. We employ one prompt template per QA format, resulting in a prompt library with four prompt templates. The prompt templates can be found in Appendix A.2.

4.1.3 JA-TRANSLATE-63B

We construct a synthetic corpus named JA-TRANSLATE-63B by translating the organic English Web corpus EN-WEB-63B into Japanese. We use a single prompt template shown in Appendix A.3 for this synthetic corpus.

4.2 Repeated Japanese Web Text

As a baseline, we perform multi-epoch training on the limited Japanese Web corpus, JA-WEB-9B. Given the 72B-token budget with 9B tokens already allocated to JA-WEB-9B, the remaining 63B tokens are supplied by seven additional passes over JA-WEB-9B, yielding eight epochs ($8 \times 9\text{B} = 72\text{B}$).

4.3 English Web Text

As an additional baseline, we consider the direct use of organic English Web text, allocating the

remaining 63B tokens to EN-WEB-63B. This baseline serves two purposes. First, it enables a head-to-head comparison with JA-TRANSLATE-63B, evaluating whether translating the same English sources into Japanese yields higher Japanese downstream performance than training directly on the original English sources. Second, it reflects common practice in many non-English-focused LLMs, which incorporate substantial English Web text to support English-related functionality (e.g., translation to/from English) and to leverage cross-lingual transfer that can benefit the target language (LLM-jp et al., 2024; Sengupta et al., 2023; Yoo et al., 2024; Conneau et al., 2019; Xue et al., 2020; Wang et al., 2024).

4.4 Additional Japanese Web Text

As an attempted upper bound, we consider filling the 63B-token shortfall with additional organic Japanese Web text that is non-overlapping with JA-WEB-9B, which we refer to as JA-WEB-63B. This oracle setting provides a reference for purely using Japanese Web data and helps contextualize the potential and limitations of the other augmentation strategies.

5 Experiments

We pre-trained LLMs from scratch under a fixed budget of 72B tokens, where 9B tokens were reserved for the limited organic Japanese Web corpus JA-WEB-9B, leaving 63B tokens for allocation to the competing data augmentation strategies described in Section 4. We evaluated their downstream task performance on Japanese to quantify the effectiveness of each data augmentation strategy.

5.1 Tokenizer

We used the LLM-jp-3 tokenizer, a subword tokenizer developed for the LLM-jp-3 model series.[4] Its vocabulary was curated to yield sensible segmentations for both Japanese and English. This choice made token counts comparable across English and Japanese, so Japanese and English corpora with the same number of tokens contained roughly comparable information.

5.2 Web Corpora

We constructed JA-WEB-9B by sampling Japanese documents totaling 9B tokens from FineWeb2 (Penedo et al., 2025), a multilingual Web corpus derived from Common Crawl. For JA-WEB-63B, we sampled additional Japanese documents of 63B tokens again from FineWeb2 with no overlap documents with JA-WEB-9B. EN-WEB-63B was obtained by sampling English documents of 63B tokens from FineWeb (Penedo et al., 2024), an English Web corpus built from Common Crawl. In all corpora, we relied on the datasets' built-in pre-processing (filtering, deduplication, language identification, etc.) and applied no additional filtering or normalization beyond sampling.

5.3 Synthetic Corpora

We applied the synthesis pipeline described in Section 4.1 to build the synthetic corpora. Concretely, each seed document was tokenized with the LLM-jp-3 tokenizer and split into non-overlapping 512-token chunks, and the synthesis procedure was applied independently to every chunk. For seed documents that produced multiple chunks, we aggregated the chunk-level generations as follows: for JA-PARAPHRASE-63B and JA-TRANSLATE-63B, we concatenated all generated texts into a single synthetic sample; for JA-INSTRUCT-63B, we appended all generated texts to the end of the original seed document to form one instance.

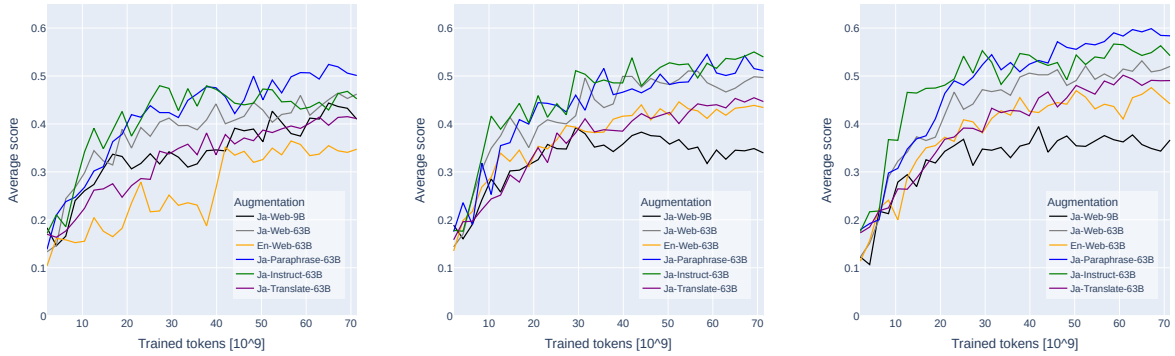
We used Qwen3-14B (Yang et al., 2025) to synthesize the Japanese corpora, motivated by its strong performance in Japanese and its permissive license. The model operated in non-reasoning mode using the decoding parameters recommended in the official documentation (temperature = 0.6, top-p = 0.95, top-k = 20). For efficient batched generation, we employed vLLM (Kwon et al., 2023).

Corpus synthesis was conducted on a GPU cluster with nodes each equipped with eight NVIDIA H200 GPUs, totaling approximately 80 node-days of compute.

5.4 Pre-Training

We pre-trained Transformer-based language models (Vaswani et al., 2017; Radford et al., 2019) with 1.8B, 3.8B, and 7.2B parameters, largely following the configurations used in the LLM-jp-3 series. Appendix B details the training configurations, including the architectural and optimization hyperparameters. On a cluster with eight NVIDIA H200 GPUs per node, each pre-training run for the 1.8B-, 3.8B-, and 7.2B-parameter models required approximately 3, 7, and 12 node-days, respectively.

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



(a) Results of the 1.8B models. (b) Results of the 3.8B models. (c) Results of the 7.2B models.

Figure 4: The average performance throughout pre-training.

(a) Results of the 1.8B models trained on the corpus with different amount of JA-PARAPHRASE-63B.

(b) Results of the 1.8B models trained on the corpus with different amount of JA-INSTRUCT-63B.

(c) Results of the 1.8B models trained on the corpus with different amount of JA-TRANSLATE-63B.

Figure 5: The average performance throughout pre-training. Each number in parentheses indicates the number of tokens sampled from the corpora. The remaining token budgets were filled by performing multi-epoch training.

5.5 Evaluation

We used the llm-jp-eval framework.[5] This framework covers a broad range of Japanese tasks and supplies standardized few-shot learning protocols (Brown et al., 2020), making it well-suited for assessing Japanese base LLMs prior to post-training. We selected benchmark tasks that require Japanese-language generation, are not overly artificial, and deliver performance above chance level as pre-training progresses. Eventually, we evaluated the performance on the following tasks:

- **JSQuAD** (Kurihara et al., 2022): Japanese reading comprehension; we evaluated character-level F1.
- **NIILC** (Sekine, 2003): Knowledge-focused Japanese QA; we evaluated character-level F1.
- **JEMHopQA** (Ishii et al., 2024): Multi-hop Japanese QA requiring reasoning; we evaluated character-level F1.
- **ALT** (Thu et al., 2016): Japanese–English parallel corpus; we evaluated En→Ja translation quality with COMET (Rei et al., 2020).

[5] https://github.com/llm-jp/llm-jp-eval. We used the version 1.4.1 for the experiments.

5.6 Results

Figure 4 shows the trajectory of the average score on the target benchmarks throughout pre-training, while Table 1 provides per-task scores obtained at the final checkpoints.

Across all model sizes, JA-PARAPHRASE-63B and JA-INSTRUCT-63B performed best. These synthetic corpora outperformed the baselines, JA-WEB-9B and EN-WEB-63B, and even matched or exceeded JA-WEB-63B, an oracle setting that filled the remaining token budget with additional organic Japanese Web text.

The comparison between JA-TRANSLATE-63B and EN-WEB-63B showed a scale-dependent effect. For the 1.8B-parameter models, translating

Augmentation	JSQuAD	NIILC	JEMHopQA	ALT	AVG
JA-WEB-9B	**.4326**	.1772	.2702	.7597	.4099
EN-WEB-63B	.2637	.1827	.2128	.7297	.3472
JA-PARAPHRASE-63B	.4354	**.3406**	**.4174**	.8109	**.5011**
JA-INSTRUCT-63B	.4175	.2837	.2958	**.8131**	.4525
JA-TRANSLATE-63B	.3562	.1854	.3138	.7892	.4112
JA-WEB-63B	.4931	.2624	.2828	.8103	.4622

(a) Task-wise performance of 1.8B-parameter pre-trained models.

Augmentation	JSQuAD	NIILC	JEMHopQA	ALT	AVG
JA-WEB-9B	.3127	.0813	.1721	.7922	.3396
EN-WEB-63B	.4175	.2839	.2388	.7967	.4342
JA-PARAPHRASE-63B	.4890	**.3883**	.3429	.8252	.5114
JA-INSTRUCT-63B	**.6137**	.3466	**.3638**	**.8347**	**.5397**
JA-TRANSLATE-63B	.3487	.2508	.3753	.8119	.4467
JA-WEB-63B	.4962	.3949	.2578	.8381	.4968

(b) Task-wise performance of 3.8B-parameter pre-trained models.

Augmentation	JSQuAD	NIILC	JEMHopQA	ALT	AVG
JA-WEB-9B	.2198	.2676	.2049	.7733	.3664
EN-WEB-63B	.3989	.3021	.2479	.8187	.4419
JA-PARAPHRASE-63B	.6270	**.4429**	**.4273**	**.8382**	**.5839**
JA-INSTRUCT-63B	**.6602**	.3812	.2978	.8309	.5425
JA-TRANSLATE-63B	.4159	.3227	.3888	.8344	.4905
JA-WEB-63B	.5187	.3909	.3232	.8473	.5200

(c) Task-wise performance of 7.2B-parameter pre-trained models.

Table 1: Task-wise performance of final pre-training checkpoints. Boldface denotes the best non-oracle result (i.e., excluding JA-WEB-63B) in each subtable.

English to Japanese yielded clear gains on downstream tasks. However, this advantage largely diminished for the 3.8B- and 7.2B-parameter models, where training directly on English Web data performed comparably. This suggests that larger models possess stronger cross-lingual generalization abilities, reducing the necessity for explicit translation.

Multi-epoch training on JA-WEB-9B consistently underperformed compared to using additional organic text from JA-WEB-63B. This degradation aligns with prior findings on the diminishing returns of repeated exposure (Muennighoff et al., 2023; Xue et al., 2023).

6 Analysis

6.1 Post-Training

While the synthetic corpora improved downstream task performance during pre-training, the source of these gains remains unclear. The improvements may arise from enhanced general language modeling, but they could also reflect instruction-following abilities implicitly learned from stylized synthetic text, which can often be acquired through lightweight post-training. Indeed, the highest-performing corpora, JA-PARAPHRASE-63B and JA-INSTRUCT-63B, contain data resembling instruction-style text.

To disentangle these effects, we conducted supervised fine-tuning for the final checkpoints of all pre-trained models. We used the training split provided by the llm-jp-eval framework,[5] which aggregated the training portions of the benchmark datasets, including those used in our experiments. Refer to Appendix C for details.

Table 2 shows the results. Even after post-training, JA-PARAPHRASE-63B and JA-INSTRUCT-63B remained the best-performing corpora. However, their performance advantages over the baselines were notably smaller than those in the pre-training results, suggesting that part of the improvement observed during pre-training can be reproduced via lightweight post-training. Neverthe-

Augmentation	JSQuAD	NIILC	JEMHopQA	ALT	AVG
JA-WEB-9B	.8496	.3462	.4504	.7860	.6081
EN-WEB-63B	.8618	.2543	.4683	.8029	.5968
JA-PARAPHRASE-63B	.8698	**.3938**	.4899	.8083	.6405
JA-INSTRUCT-63B	**.9126**	.3765	**.4979**	**.8238**	**.6527**
JA-TRANSLATE-63B	.8502	.2830	.4827	.8097	.6064
JA-WEB-63B	.8791	.4029	.4547	.8189	.6389

(a) Task-wise performance of 1.8B-parameter fine-tuned models.

Augmentation	JSQuAD	NIILC	JEMHopQA	ALT	AVG
JA-WEB-9B	.8725	**.4339**	.4779	.8233	.6519
EN-WEB-63B	.8872	.3084	.4647	**.8566**	.6292
JA-PARAPHRASE-63B	.8847	.4196	**.4831**	.8391	.6566
JA-INSTRUCT-63B	**.9120**	.4255	.4532	.8518	**.6606**
JA-TRANSLATE-63B	.8792	.3401	.4532	.8390	.6279
JA-WEB-63B	.9045	.4509	.5403	.8597	.6889

(b) Task-wise performance of 3.8B-parameter fine-tuned models.

Augmentation	JSQuAD	NIILC	JEMHopQA	ALT	AVG
JA-WEB-9B	.8759	.4140	.4415	.8254	.6392
EN-WEB-63B	.8940	.3618	.4510	.8704	.6443
JA-PARAPHRASE-63B	.8930	**.5061**	.5218	.8520	.6932
JA-INSTRUCT-63B	**.9185**	.4673	**.5344**	**.8548**	**.6938**
JA-TRANSLATE-63B	.8878	.4035	.5326	.8561	.6700
JA-WEB-63B	.9114	.5008	.5061	.8695	.6970

(c) Task-wise performance of 7.2B-parameter fine-tuned models.

Table 2: Task-wise performance of fine-tuned model checkpoints (1.8B, 3.8B, 7.2B). Boldface denotes the best non-oracle result (excluding JA-WEB-63B) in each subtable.

less, the remaining performance margins indicate that synthetic corpora also contribute to better general language modeling.

6.2 Data Leakage

The apparent performance gains from the synthetic corpora might stem from data leakage between evaluation benchmarks and training corpora. Following Chowdhery et al. (2022), we assessed contamination by measuring n-gram overlap between the training corpora and evaluation instances. Specifically, Chowdhery et al. (2022) consider an instance contaminated if at least 70% of its word-level 8-grams appear in the training data. To approximate this criterion for Japanese, we assumed an average word length of approximately 4 characters in English and 2 characters in Japanese, and therefore conducted contamination detection based on character-level 16-gram coverage.

Table 3 shows the result. Across all conditions, contamination is negligible, and importantly, we observe no increase attributable to synthetic data augmentation. The modest contamination in JSQuAD is expected, as this benchmark derives passages from Japanese Wikipedia, and Web corpora typically contain a significant volume of Wikipedia text. Overall, these results indicate that the observed performance improvements are unlikely to stem from direct data leakage.

6.3 Data-constrained Training

While synthetic corpora substantially improved performance, generating large-scale synthetic data incurs considerable computational cost. To explore more cost-efficient alternatives, we considered partially covering the 63B-token shortfall and compensating the remaining tokens via additional training epochs. Specifically, we evaluated two settings: (i) synthesizing 27B tokens and training for two epochs, and (ii) synthesizing 9B tokens and training for four epochs.

Figure 5 presents the results for the 1.8B-parameter models. Across most settings, performance remained comparable to the full-budget

	JA-WEB-9B	JA-PARAPHRASE-63B	JA-INSTRUCT-63B	JA-TRANSLATE-63B	JA-WEB-63B
JSQuAD	.045	.004	.046	.000	.165
NIILC	.020	.020	.020	.010	.025
JEMHopQA	.000	.000	.000	.000	.000
ALT	.000	.000	.000	.000	.000

Table 3: Proportion of evaluation instances flagged as contaminated. A score of 1 indicates that all instances in the dataset are classified as contaminated.

training, indicating that moderate data reuse through multi-epoch training can largely offset a reduced amount of synthetic data. One exception was JA-PARAPHRASE-63B, for which repeated exposure led to a slight performance degradation. Overall, these results suggest that scaling synthetic data does not require fully eliminating the data shortfall; instead, partial synthesis combined with additional epochs can yield similar downstream performance at lower computational cost.

7 Conclusions

We studied LLM pre-training under a strict data budget. Experiments showed that synthetic corpora effectively mitigate data scarcity; across model sizes, paraphrasing and instruction-style synthesis yielded the strongest downstream gains, often matching or surpassing filling the data shortfall with additional organic text. These gains persisted even after post-training for instruction following, suggesting synthetic corpora contributed to better general language modeling.

Our results suggest practical guidance for pre-training non-English LLMs: (1) expand target-language data by paraphrasing or instruction-formatting rather than duplicating a small corpus; and (2) partially filling the shortfall with synthetic text and training for a moderate number of epochs can approach the performance of far larger synthetic budgets.

Limitations

Synthesizer Variation Our synthetic corpora were generated exclusively with Qwen3-14B (Yang et al., 2025). We made this choice to (i) control computational cost and (ii) hold the synthesizer fixed across settings so that performance differences can be attributed to data construction rather than synthesizer variation. Qwen3-14B was selected because it reliably performs paraphrasing, document-grounded QA generation, and translation. In addition, Qwen3-14B is available under the Apache-2.0 license, which facilitates broad reusability of the resultant synthetic data. While recent work suggests that paraphrasing benefits are largely robust to the choice of the synthesizer (DatologyAI, 2025), we have not directly verified that here. Establishing robustness across multiple synthesizers remains future work.

Language Scope Our study focused solely on Japanese. As discussed in the Introduction, Japanese provides a suitable testbed. Nevertheless, our central conclusion that increasing target-language coverage via paraphrasing or instruction-style reformulation is more effective than adding English data or naively repeating a limited target-language corpus should be tested in other languages before being taken as general.

Assumption of Access to a Target-Language–Fluent LLMs Our pipeline assumes access to an LLM that is sufficiently fluent in the target language. This assumption may not hold for some low-resource languages. Although prior work reports that paraphrase-based gains can be robust across synthesizers (DatologyAI, 2025), the availability and quality of target-language generators will vary. Adapting the pipeline to settings without a strong target-language LLM is an important direction.

Potential Risks

Pre-training on synthetic text entails specific risks:

- **Inherited bias from the synthesizer.** Synthetic data can encode the synthesizer model's social, topical, and stylistic biases, which may then be transferred to the target model.
- **Bias amplification via bootstrapping.** When models trained on synthetic data are used to produce new synthetic data, feedback loops can magnify existing biases, propagate errors, and reduce distributional diversity.

In this study, we did not perform targeted filtering or de-biasing for these issues. Users who train

on our corpora should implement appropriate safeguards (e.g., bias/toxicity screening, source diversification, and independent fairness audits) to mitigate these risks.

Use of AI Assistants

We used privacy-preserving AI assistants solely for the purpose of improving writing quality.

Acknowledgments

We used ABCI 3.0 provided by AIST and AIST Solutions with support from "ABCI 3.0 Development Acceleration Use."

References

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J. Hewett, Mojan Javaheripi, Piero Kauffmann, James R. Lee, Yin Tat Lee, Yuanzhi Li, Weishung Liu, Caio C. T. Mendes, Anh Nguyen, Eric Price, Gustavo de Rosa, Olli Saarikivi, and 8 others. 2024. Phi-4 technical report. *arXiv preprint arXiv:2412.08905*.

Zeyuan Allen-Zhu and Yuanzhi Li. 2023. Physics of language models: Part 3.1, knowledge storage and extraction. *arXiv preprint arXiv:2309.14316*.

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, and 12 others. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Daixuan Cheng, Yuxian Gu, Shaohan Huang, Junyu Bi, Minlie Huang, and Furu Wei. 2024. Instruction pre-training: Language models are supervised multitask learners. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 2529–2550, Miami, Florida, USA. Association for Computational Linguistics.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, and 48 others. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

DatologyAI. 2025. Beyondweb: Lessons from scaling synthetic data for trillion-scale pretraining. *arXiv preprint arXiv:2508.10975*.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, and 181 others. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Meet Doshi, Raj Dabre, and Pushpak Bhattacharyya. 2024. Pretraining language models using translationese. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 5843–5862.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, and 197 others. 2025. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Harkirat Singh Behl, Xin Wang, Sébastien Bubeck, Ronen Eldan, Adam Tauman Kalai, Yin Tat Lee, and Yuanzhi Li. 2023. Textbooks are all you need. *arXiv preprint arXiv:2306.11644*.

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

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, and 3 others. 2022. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*.

Kaito Horio, Eiki Murata, Hao Wang, Tatsuya Ide, Daisuke Kawahara, Takato Yamazaki, Kenta Shinzato, Akifumi Nakamachi, Shengzhe Li, and Toshinori Sato. 2025. Verification of chain-of-thought prompting in japanese. In *Proceedings of the 37th Annual Conference of the Japanese Society for Artificial Intelligence*. In Japanese.

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, Torino, Italia. ELRA and ICCL.

Zhengbao Jiang, Zhiqing Sun, Weijia Shi, Pedro Rodriguez, Chunting Zhou, Graham Neubig, Xi Lin, Wen-tau Yih, and Srini Iyer. 2024. Instruction-tuned language models are better knowledge learners. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5421–5434, Bangkok, Thailand. Association for Computational Linguistics.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Ai Kawazoe, Ribeka Tanaka, Koji Mineshima, and Daisuke Bekki. 2015. A framework for constructing multilingual inference problem sets: Highlighting similarities and differences in semantic phenomena between english and japanese. In *Proceedings of the 1st International Workshop on the Use of Multilingual Language Resources in Knowledge Representation Systems*.

Takahiro Kubo and Hiroki Nakayama. 2018. chabsa: Aspect based sntiment analysis dataset in japanese. https://github.com/chakki-works/chABSA-dataset/blob/master/doc/chabsa-aspect-based.pdf. Accessed: 2025-10-06.

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

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*.

LLM-jp, Akiko Aizawa, Eiji Aramaki, Bowen Chen, Fei Cheng, Hiroyuki Deguchi, Rintaro Enomoto, Kazuki Fujii, Kensuke Fukumoto, Takuya Fukushima, Namgi Han, Yuto Harada, Chikara Hashimoto, Tatsuya Hiraoka, Shohei Hisada, Sosuke Hosokawa, Lu Jie, Keisuke Kamata, Teruhito Kanazawa, and 63 others. 2024. Llm-jp: A cross-organizational project for the research and development of fully open japanese llms. *arXiv preprint arXiv:2407.03963*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *Proceedings of the Seventh International Conference on Learning Representations*.

Pratyush Maini, Skyler Seto, Richard Bai, David Grangier, Yizhe Zhang, and Navdeep Jaitly. 2024. Rephrasing the web: A recipe for compute and data-efficient language modeling. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14044–14072, Bangkok, Thailand. Association for Computational Linguistics.

Niklas Muennighoff, Alexander Rush, Boaz Barak, Teven Le Scao, Nouamane Tazi, Aleksandra Piktus, Sampo Pyysalo, Thomas Wolf, and Colin A Raffel. 2023. Scaling data-constrained language models. *Advances in Neural Information Processing Systems*, 36:50358–50376.

Thao Nguyen, Yang Li, Olga Golovneva, Luke Zettlemoyer, Sewoong Oh, Ludwig Schmidt, and Xian Li. 2025. Recycling the web: A method to enhance pre-training data quality and quantity for language models. In *Proceedings of the Second Conference on Language Modeling*.

NICT. 2011. Japanese-english bilingual corpus of wikipedia's kyoto articles. https://alaginrc.nict.go.jp/WikiCorpus/index_E.html. Accessed: 2025-10-06.

Kazumasa Omura, Daisuke Kawahara, and Sadao Kurohashi. 2020. A Method for Building a Commonsense Inference Dataset based on Basic Events. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2450–2460, Online. Association for Computational Linguistics.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. 2024. The fineweb datasets: Decanting the web for the finest text data at scale. *Advances in Neural Information Processing Systems*, 37:30811–30849.

Guilherme Penedo, Hynek Kydlíček, Vinko Sabolčec, Bettina Messmer, Negar Foroutan, Amir Hossein Kargaran, Colin Raffel, Martin Jaggi, Leandro Von Werra, and Thomas Wolf. 2025. Fineweb2: One pipeline to scale them all — adapting pre-training data processing to every language. In *Proceedings of the Second Conference on Language Modeling*.

Zeyu Qin, Qingxiu Dong, Xingxing Zhang, Li Dong, Xiaolong Huang, Ziyi Yang, Mahmoud Khademi, Dongdong Zhang, Hany Hassan Awadalla, Yi R. Fung,

Weizhu Chen, Minhao Cheng, and Furu Wei. 2025. Scaling laws of synthetic data for language model. In *Proceedings of the Second Conference on Language Modeling*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

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)*, pages 2685–2702, Online. Association for Computational Linguistics.

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

Neha Sengupta, Sunil Kumar Sahu, Bokang Jia, Satheesh Katipomu, Haonan Li, Fajri Koto, William Marshall, Gurpreet Gosal, Cynthia Liu, Zhiming Chen, Osama Mohammed Afzal, Samta Kamboj, Onkar Pandit, Rahul Pal, Lalit Pradhan, Zain Muhammad Mujahid, Massa Baali, Xudong Han, Sondos Mahmoud Bsharat, and 13 others. 2023. Jais and jais-chat: Arabic-centric foundation and instruction-tuned open generative large language models. *arXiv preprint arXiv:2308.16149*.

Gerald Shen, Zhilin Wang, Olivier Delalleau, Jiaqi Zeng, Yi Dong, Daniel Egert, Shengyang Sun, Jimmy Zhang, Sahil Jain, Ali Taghibakhshi, Markel Sanz Ausin, Ashwath Aithal, and Oleksii Kuchaiev. 2024. Nemo-aligner: Scalable toolkit for efficient model alignment. *arXiv preprint arXiv:2405.01481*.

Freda Shi, Mirac Suzgun, Markus Freitag, Xuezhi Wang, Suraj Srivats, Soroush Vosoughi, Hyung Won Chung, Yi Tay, Sebastian Ruder, Denny Zhou, Dipanjan Das, and Jason Wei. 2022. Language models are multilingual chain-of-thought reasoners. *arXiv preprint arXiv:2210.03057*.

Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. 2019. Megatron-lm: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*.

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 (Volume 4: Student Research Workshop)*, pages 57–68, Toronto, Canada. Association for Computational Linguistics.

Masashi Takeshita, Rafal Rzpeka, and Kenji Araki. 2023. Jcommonsensemorality: Japanese dataset for evaluating commonsense morality understanding. In *Proceedings of The Twenty Nineth Annual Meeting of The Association for Natural Language Processing (NLP2023)*, pages 357–362. In Japanese.

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'16)*, pages 1574–1578, Portorož, Slovenia. European Language Resources Association (ELRA).

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. *arXiv preprint arXiv:2302.13971*.

Nobuhiro Ueda. 2023. Wikipedia annotated corpus. https://github.com/ku-nlp/WikipediaAnnotatedCorpus. Accessed: 2025-10-06.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Pablo Villalobos, Anson Ho, Jaime Sevilla, Tamay Besiroglu, Lennart Heim, and Marius Hobbhahn. 2024. Position: Will we run out of data? limits of llm scaling based on human-generated data. In *Proceedings of the Forty-first International Conference on Machine Learning*.

Hetong Wang, Pasquale Minervini, and Edoardo M Ponti. 2024. Probing the emergence of cross-lingual alignment during llm training. *arXiv preprint arXiv:2406.13229*.

Jiayi Wang, Yao Lu, Maurice Weber, Max Ryabinin, David Adelani, Yihong Chen, Raphael Tang, and Pontus Stenetorp. 2025a. Multilingual language model pretraining using machine-translated data. *arXiv preprint arXiv:2502.13252*.

Zengzhi Wang, Fan Zhou, Xuefeng Li, and Pengfei Liu. 2025b. Octothinker: Mid-training incentivizes reinforcement learning scaling. *arXiv preprint arXiv:2506.20512*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Fuzhao Xue, Yao Fu, Wangchunshu Zhou, Zangwei Zheng, and Yang You. 2023. To repeat or not to repeat: Insights from scaling llm under token-crisis. In *Advances in Neural Information Processing Systems*, volume 36, pages 59304–59322. Curran Associates, Inc.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2020. mt5: A massively multilingual pre-trained text-to-text transformer. *arXiv preprint arXiv:2010.11934*.

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.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

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, Miami, Florida, USA. Association for Computational Linguistics.

Kang Min Yoo, Jaegeun Han, Sookyo In, Heewon Jeon, Jisu Jeong, Jaewook Kang, Hyunwook Kim, Kyung-Min Kim, Munhyong Kim, Sungju Kim, Donghyun Kwak, Hanock Kwak, Se Jung Kwon, Bado Lee, Dongsoo Lee, Gichang Lee, Jooho Lee, Baeseong Park, Seongjin Shin, and 377 others. 2024. Hyperclova x technical report. *arXiv preprint arXiv:2404.01954*.

A Prompt Templates

We list the prompt templates used to construct our synthetic Japanese corpora.

A.1 JA-PARAPHRASE-63B

We list prompt templates used to construct the JA-PARAPHRASE-63B corpus.

Easy Style

> For the following paragraph give me a paraphrase of the same using a very small vocabulary and extremely simple sentences that a toddler will understand. The content must be written in Japanese:
> {{ input }}

Medium Style

> For the following paragraph give me a diverse paraphrase of the same in high quality Japanese language as in sentences on Wikipedia. The content must be written in Japanese:
> {{ input }}

Hard Style

> For the following paragraph give me a paraphrase of the same using very terse and abstruse language that only an erudite scholar will understand. Replace simple words and phrases with rare and complex ones. The content must be written in Japanese:
> {{ input }}

QA Style

> Convert the following paragraph into a conversational format with multiple tags of "Question:" followed by "Answer:". The content must be written in Japanese:
> {{ input }}

A.2 JA-INSTRUCT-63B

We list the prompt templates used to construct the JA-INSTRUCT-63B corpus.

Free-form

> Create a question and provide a corresponding answer based on the given paragraph. The content should be formatted with "Question:" followed by "Answer:". The content must be written in Japanese.
> Input: {{ input }}

Multiple-choice

> Create a multiple-choice question and provide a corresponding answer based on the given paragraph. The content should be formatted with "Question:" followed by "Answer:". The question section should include four answer choices. The content must be written in Japanese.
> Input: {{ input }}

Free-form + CoT

> Create a question and provide a corresponding answer based on the given paragraph. The

	1.8B	3.8B	7.2B
Layers	24	28	32
Hidden size	2,048	3,072	4,096
FFN hidden size	7,168	8,192	11,008
Attention heads	16	24	32
Learning rate	3e-4	3e-4	2e-4

Table 4: Pre-training configurations.

content should be formatted with "Question:" followed by "Reasoning:" and "Answer:". The reasoning section should provide reasoning steps leading to the final answer. The content must be written in Japanese.
Input: {{ input }}

Multiple-choice + CoT

Create a multiple-choice question and provide a corresponding answer based on the given paragraph. The content should be formatted with "Question:" followed by "Reasoning:" and "Answer:". The question section should include four answer choices. The reasoning section should provide reasoning steps leading to the final answer. The content must be written in Japanese.
Input: {{ input }}

A.3 JA-TRANSLATE-63B

We show the prompt template used to construct the JA-TRANSLATE-63B corpus.

Translate the following paragraph into Japanese:
{{ input }}

B Pre-training Configurations

Table 4 summarizes the architectural and optimization hyperparameters. We optimized the causal language modeling objective with AdamW (Loshchilov and Hutter, 2019) and employed a cosine learning-rate schedule with 2,000 warmup steps, after which the learning rate decayed to one tenth of its peak value over training. We used a batch size of 512 and a maximum sequence length of 2,048 tokens. We built our codebase on top of Megatron-LM (Shoeybi et al., 2019). Each model consumed the 72B tokens during its pre-training, corresponding to roughly 68,000 training steps.

C Fine-tuning Configurations

In fine-tuning, we used the training split provided by the llm-jp-eval framework,[5] which aggregated the training portions of the benchmark datasets. Table 5 shows the list of the datasets we used. Each model was fine-tuned for three epochs with a learning rate of 2e-5 under cosine learning rate scheduling. We built our code base on NeMo-Aligner (Shen et al., 2024).

Task Category	Dataset	# of examples
Natural Language Inference	Jamp (Sugimoto et al., 2023)	8,955
	JaNLI (Yanaka and Mineshima, 2021)	12,312
	JNLI (Kurihara et al., 2022)	18,065
	JSeM (Kawazoe et al., 2015)	12,667
	JSICK (Yanaka and Mineshima, 2022)	4,500
Question Answering	JEMHopQA (Ishii et al., 2024)	953
	NIILC (Sekine, 2003)	715
Reading Comprehension	JSQuAD (Kurihara et al., 2022)	56,573
Multiple Choice QA	JCommonsenseMorality (Takeshita et al., 2023)	13975
	JCommonsenseQA (Kurihara et al., 2022)	8045
	KUCI (Omura et al., 2020)	83127
Entity Linking	chABSA (Kubo and Nakayama, 2018)	2,572
Fundamental Analysis	Wikipedia Annotated Corpus: Coreference (Ueda, 2023)	1,515
	Wikipedia Annotated Corpus: Dependency (Ueda, 2023)	1,517
	Wikipedia Annotated Corpus: NER (Ueda, 2023)	881
	Wikipedia Annotated Corpus: PAS Analysis (Ueda, 2023)	1,514
	Wikipedia Annotated Corpus: Reading (Ueda, 2023)	1,517
Mathematical Reasoning	MAWPS (Horio et al., 2025)	16
	MGSM (Shi et al., 2022)	8
Machine Translation	ALT En→Ja (Thu et al., 2016)	17,972
	ALT: Ja→En (Thu et al., 2016)	17,972
	WikiCorpus: En→Ja (NICT, 2011)	28061
	WikiCorpus: Ja→En (NICT, 2011)	28061
Semantic Textual Similarity	JSTS (Kurihara et al., 2022)	11205
Human Examination	MMLU (Hendrycks et al., 2021)	285
	JMMLU (Yin et al., 2024)	32

Table 5: Datasets used for supervised fine-tuning. Task category labels follow those defined in the llm-jp-eval framework[5].