Why We Build Local Large Language Models: An Observational Analysis from 35 Japanese and Multilingual LLMs

Anonymous ACL submission

Abstract

Why do we build local large language models (LLMs)? What should a local LLM learn from the target language? Which abilities can be transferred from other languages? Do language-specific scaling laws exist? To explore these research questions, we evaluated 35 Japanese, English, and multilingual LLMs on 19 evaluation benchmarks for Japanese and English, taking Japanese as a local language. Adopting an observational approach, we analyzed correlations of benchmark scores, and conducted principal component analysis (PCA) on the scores to derive *ability factors* of local LLMs.

We found that training on English text can improve the scores of academic subjects in Japanese (JMMLU). In addition, it is unnecessary to specifically train on Japanese text to enhance abilities for solving Japanese code generation, arithmetic reasoning, commonsense, and reading comprehension tasks. In contrast, training on Japanese text could improve question-answering tasks about Japanese knowledge and English-Japanese translation, which indicates that abilities for solving these two tasks can be regarded as *Japanese abilities* for LLMs. Furthermore, we confirmed that the Japanese abilities scale with the computational budget for Japanese text.

1 Introduction

Major large language models (LLMs) are English-centric (*English LLMs* hereafter), e.g., Meta Llama 3 (Dubey et al., 2024), Mistral (Jiang et al., 2023), and Phi-3 (Abdin et al., 2024), due to the dominance of English on the internet and the global economy, which results in a limited focus on non-English languages. Several companies and research institutes have been actively developing LLMs that perform well on non-English texts (*local LLMs* hereafter), e.g., Bllossom (Choi et al., 2024), Chinese-LLaMA (Cui et al., 2024) and open-Cabrita (Larcher et al., 2023), driven by various motivations. These include advancing research and development in natural language processing, mitigating security risks associated with relying on a limited number of foreign companies, and promoting responsible artificial intelligence for their community.

Recently, dozens of Japanese-centric LLMs (*Japanese LLMs* hereafter) have been developed in Japan, such as Sarashina2[1], Llama 3.1 Swallow[2], and LLM-jp (LLM-jp et al., 2024). However, the advantages of training LLMs on non-English text, such as Japanese, remain underexplored. While LLMs have demonstrated high multilingual abilities, such as arithmetic reasoning (Shi et al., 2023) and machine translation (Briakou et al., 2023), Zhang et al. (2023) reported that they performed poorly on non-English commonsense reasoning. Although Fujii et al. (2024) reported that training on Japanese text improved question-answering (QA) tasks, the contribution of Japanese training data on each task has not been investigated.

A straightforward approach for analyzing the impact of Japanese training data in LLMs is to conduct ablation studies; more specifically, we prepare training data by changing the size and mix of Japanese data sets, train an LLM on the data, and measure its performance. However, this approach is inefficient due to the significant computational resources required. Even if such studies were conducted, it would remain unclear whether the findings could be generalized beyond specific design choices, such as variations in training data, the numbers of model parameters, or pre-training methods (from scratch or continual pre-training).

Instead of cost-intensive ablation studies, this paper adopts an observational approach (Ruan et al., 2024) for Japanese LLMs, leveraging the excep-

[1] https://huggingface.co/sbintuitions/sarashina2-7b

[2] https://swallow-llm.github.io/llama3-swallow.en.html

tionally active development of Japanese LLMs among non-English initiatives. Specifically, we evaluate publicly available 35 Japanese, English, and multilingual LLMs representing a variety of design choices. We also use 19 comprehensive evaluation benchmarks, including Japanese translations of English benchmarks, where the same tasks are evaluated in both languages. Our goal is to derive insights that are generalizable (unrestricted to specific design choices) through a quantitative analysis focusing on the following points.

First, to explore multilinguality, we analyzed score correlations across 19 task benchmarks for 35 LLMs, and conducted Principal Component Analysis (PCA) to represent the performance in a low-dimensional *ability space* (Ruan et al., 2024). We found that tasks such as academic subjects, code generation, and arithmetic reasoning exhibited strong cross-lingual correlations on their scores and were associated with the same ability factors across languages. This indicates strong multilingual transferability, suggesting that training in English text would also improve performance in Japanese for these tasks. Conversely, tasks such as QA about Japanese knowledge and English-Japanese translation exhibited weak correlations with other English tasks and were strongly associated with an independent ability factor, which indicates language-specific abilities.

Second, to examine language-specific scaling laws, we defined the language-specific computational budget as the product of the number of parameters and training tokens for each language (Hoffmann et al., 2022), and analyzed the log-linear relationship between these budgets and the ability factors obtained by PCA. We found that the English computational budget showed a strong correlation with the general ability factor but a weak correlation with the Japanese-specific ability factor. In contrast, the Japanese computational budget showed a strong correlation with the Japanese ability factor, suggesting that tasks such as QA about Japanese knowledge and English-Japanese translation scale with the amount of Japanese text and are difficult to learn solely on English texts.

2 Related Work

2.1 Correlations between Tasks and Ability Factors

Several prior studies have investigated the correlations between different task benchmarks and associated the task performance with a small number of ability factors (Ruan et al., 2024; Ni et al., 2024; Tiong et al., 2024). These studies have reported strong correlations between knowledge-based QA tasks and identified ability factors specific to arithmetic reasoning and code generation. Additionally, Ruan et al. (2024) observed the log-linear relationship between the computational budget and ability factors. However, these discussions are limited to English monolingual settings, leaving cross-language generalization and scaling laws in multilingual contexts, including Japanese and English as in our study, unexplored.

2.2 Effects of Training on Non-English Text

There is a growing number of studies examining both the promising and disappointing impacts of training local LLMs on target language data.

On the promising side, continual pre-training (continued pre-training) of strong English LLMs on non-English languages such as Chinese (Cui et al., 2024), Korean (Choi et al., 2024), Portuguese (Larcher et al., 2023), and Thai (Pipatanakul et al., 2023) has reported improvements on a variety of tasks in the target languages, including commonsense reasoning, reading comprehension, question answering, and academic subjects. Tejaswi et al. (2024) conducted systematic experiments under continual pre-training settings and reported that the effectiveness varies across different base LLMs.

On the disappointing side, Berend (2022) reported that multilingual training does not always improve performance due to the curse of multilingualty (Conneau et al., 2020). In addition, Holmström et al. (2023) reported that a Swedish LLM trained from scratch performed poorly compared to GPT-3, highlighting the difficulty of outperforming strong multilingual LLMs. Furthermore, English and multilingual LLMs reportedly show strong multilingual abilities on tasks such as arithmetic and commonsense reasoning (Shi et al., 2023) through cross-language generalization (Zhang et al., 2023). These findings suggest that the benefits of training on non-English text might be limited or, at the very least, task-dependent.

Despite these debates, there has not yet been a comprehensive and cross-lingual benchmarking using a wide variety of LLM families to assess the effect of training on non-English text.

3 Experimental Settings

3.1 Models

To obtain generalizable insights, we evaluated publicly available 35 Japanese, English, and Multilingual LLMs (see Table 1 in Appendix A.1 for the complete list), which represent diverse design choices, including training data, the number of model parameters, and pre-training approach. The evaluated models include: English LLMs (e.g., Llama 3 (Dubey et al., 2024), Mistral (Jiang et al., 2023), and Mixtral (Jiang et al., 2024)); Japanese LLMs continually pre-trained from English base LLMs on 18–175 billion tokens of Japanese text (e.g., Llama 3 Swallow (Fujii et al., 2024) and Llama 3 Youko (Sawada et al., 2024)); Japanese LLMs pre-trained primarily on 130–1,050 billion tokens of Japanese text from scratch (e.g., LLM-jp (LLM-jp et al., 2024) and Sarashina2; and multilingual LLMs pre-trained on multilingual data including Japanese (e.g., C4AI Command-R[3] and Qwen2 (Yang et al., 2024)). Notably, all the English LLM families that served as base models for the continually pre-trained Japanese LLMs were evaluated as well. We focused on base models and did not evaluate instruction-tuned models to examine the effect of pre-training and avoid the confounding effects of task-oriented instruction tuning.

To estimate the computational budget for each model, we collected data on the number of model parameters and the number of training tokens for Japanese, English, and total across all languages from official sources such as technical reports, press-release documents, and model cards. Refer to Appendix A.3 for details. For a continually pre-trained model, we calculated the total number of training tokens by summing the tokens used in both initial and continual pre-training stages.

3.2 Evaluation Tasks and Benchmarks

We evaluated the models using 19 evaluation benchmarks in both Japanese and English[4] , which is listed in Table 2 of Appendix A.2. These tasks were selected from the perspective of cross-lingual benchmarking and comprehensiveness for general-purpose LLMs. The evaluation was conducted using zero-shot or few-shot in-context learning settings depending on tasks. Refer to Appendix A.2 for details.

We employed some Japanese benchmarks corresponding to their English counterparts for cross-lingual benchmarking: code generation (JHumanEval (Sato et al., 2024) vs. HumanEval (Chen et al., 2021)), commonsense (JCommonsenseQA (Kurihara et al., 2022) vs. XWINO (Tikhonov and Ryabinin, 2021) and HellaSwag (Zellers et al., 2019)), arithmetic reasoning (MGSM (Shi et al., 2023) vs. GSM8K (Cobbe et al., 2021)), encyclopedic knowledge-based QA (JEMHopQA (Ishii et al., 2023) and NIILC (Sekine, 2003) vs. TriviaQA (Joshi et al., 2017)), reading comprehension (JSQuAD (Kurihara et al., 2022) vs. SQuAD2 (Rajpurkar et al., 2018)), and academic subjects (JMMLU (Yin et al., 2024) vs. MMLU (Hendrycks et al., 2021)). Notably, MGSM, JMMLU, and JHumanEval are translations of GSM8K, MMLU, and HumanEval, respectively. Cross-lingual correlations between these benchmarks provide insights into the multilinguality and language specificity of each task. It is also worth noting that JEMHopQA and NIILC are developed based on Japanese Wikipedia and include instances that assess knowledge specific to Japanese culture, such as history, geography, notable figures and society, making them suitable for evaluating how much LLMs acquire knowledge about Japan.

For comprehensiveness, inspired by the natural language processing taxonomy (Chang et al., 2024; Guo et al., 2023) and to capture as many ability factors as possible, we included additional task benchmarks beyond cross-lingual benchmarks. Specifically, we employed Japanese automatic summarization (XL-Sum (Hasan et al., 2021)), machine translation between English and Japanese (WMT20-en-ja and ja-en (Barrault et al., 2020)), English question answering (OpenBookQA (Mihaylov et al., 2018)), and logical reasoning (BigBench-Hard (Suzgun et al., 2023)). Because we posit that local LLMs serve as foundational models for the target language, our evaluation focused on fundamental knowledge and skills rather than domain-specific tasks (e.g., question answering in financial or medical domains). Furthermore, we excluded safety and bias-related tasks, as these should be addressed in the post-training stage.

3.3 Definition of the Computational Budgets

The Chinchilla scaling laws (Hoffmann et al., 2022) propose an approximation for training FLOPs as

[3] https://huggingface.co/CohereForAI/c4ai-command-r-v01

[4] The evaluation scores for each model will be publicly available on Zenodo with a CC-BY Attribute license upon acceptance (for blind review).

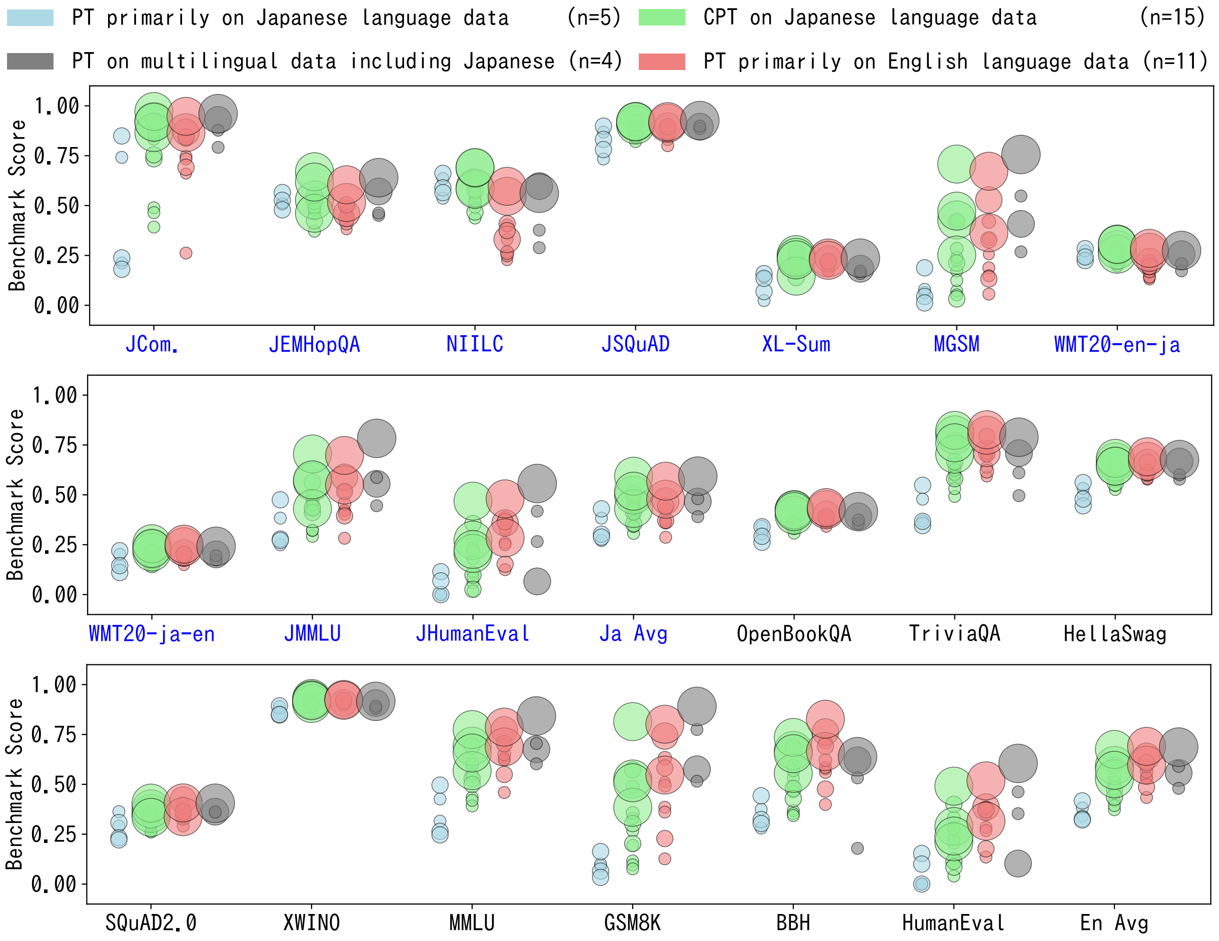


Figure 1: Task performance grouped by primary language of LLMs. Bubble size indicates the number of parameters.

$C \approx 6ND$, where C represents the training FLOPs, N is the number of parameters, and D is the number of training tokens. Following this formula, we define ND_l as the computational budget, where D_l is the training tokens for the language l.

3.4 Evaluation Framework and Environment

We evaluated all 35 LLMs on 19 task benchmarks by using a custom implementation[5] of existing evaluation frameworks such as llm-jp-eval (Han et al., 2024) and the Language Model Evaluation Harness[6]. Refer to Table 3 for the details of implementations used for evaluation. We used NVIDIA A100 GPUs mostly for the evaluations.

4 Experimental Results

Based on the experimental setting explained in the previous section, we obtained a matrix of benchmark scores $X \in \mathbb{R}^{M \times D}$, where M and D are the numbers of LLMs and benchmarks, respectively ($M = 35$ and $D = 19$ in this study) and an element $X_{i,j}$ presents the score of the LLM i on the benchmark j. In this section, we analyze the matrix X to unveil the difference in training strategies for Japanese LLMs (§ 4.1), similarity of benchmarks in terms of LLM performance (§ 4.2), ability factors of LLMs (§ 4.3). We then confirm that our findings about ability factors are aligned with the scaling laws (§ 4.4) and generalizable to different training strategies (§ 4.5).

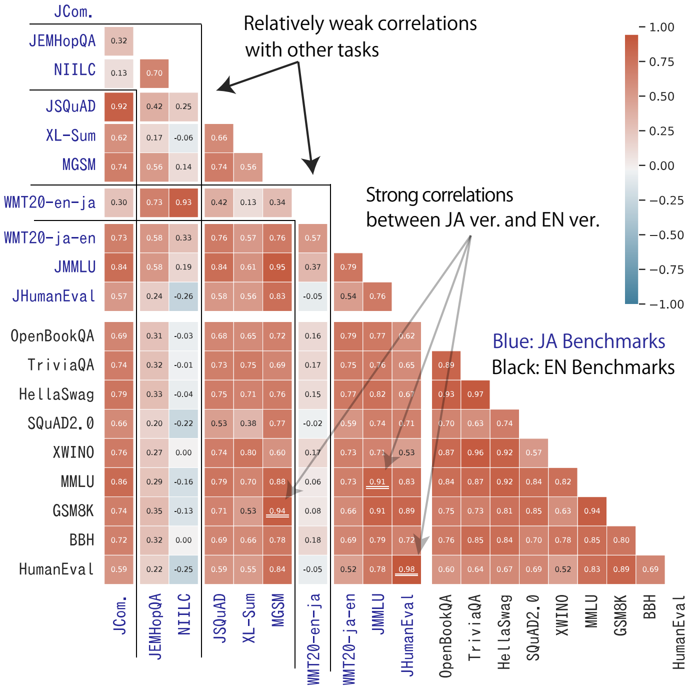
4.1 Comparison of Benchmark Scores by Pre-trained Languages

Figure 1 presents a bubble chart showing the benchmark score distributions grouped by the primary language of the LLMs: Japanese continually pretrained (light blue), Japanese trained from scratch (green), English (red), and Multilingual (gray). The variable n in each group represents the number of models included.

On overall, it is evident that LLMs with larger parameters tend to achieve higher scores in each group. When comparing benchmark scores for

[5] Our implementation has been available on Github, but is hidden here for blind review.

[6] https://zenodo.org/records/10256836



Relatively weak correlations with other tasks

Strong correlations between JA ver. and EN ver.

Blue: JA Benchmarks
Black: EN Benchmarks

Figure 2: Pearson correlation matrix among task benchmarks ($n = 35$).

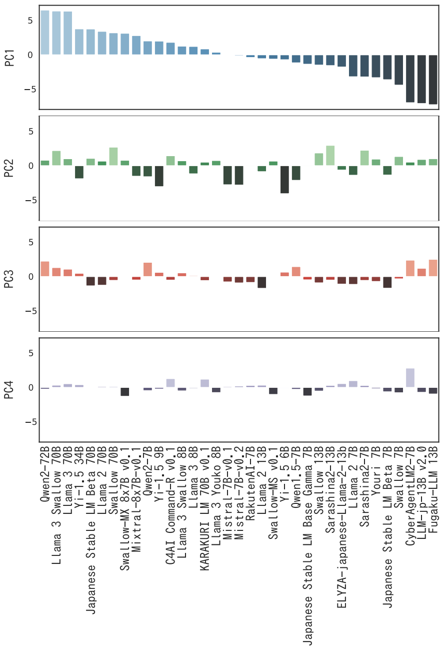


Figure 3: Principal component scores for each LLM.

smaller models, there is a clear tendency for LLMs continually pre-trained on Japanese text (the green bubbles) to outperform English LLMs (the red bubbles) on Japanese benchmarks (shown in blue) except JHumanEval and MGSM. This indicates the effectiveness of continual pre-training on Japanese text. The advantage is particularly evident in tasks such as Japanese QA (NIILC) and English-Japanese translation (WMT20-en-ja). Refer to Appendix B for detailed discussion. Considering their smaller number of parameters, Japanese LLMs trained from scratch (green bubbles) achieve competitive scores on most Japanese benchmarks, with the exceptions of the arithmetic reasoning (MGSM) and the code-generation (JHumanEval).

4.2 Correlation Between Evaluation Benchmarks and Language-Specific Performance

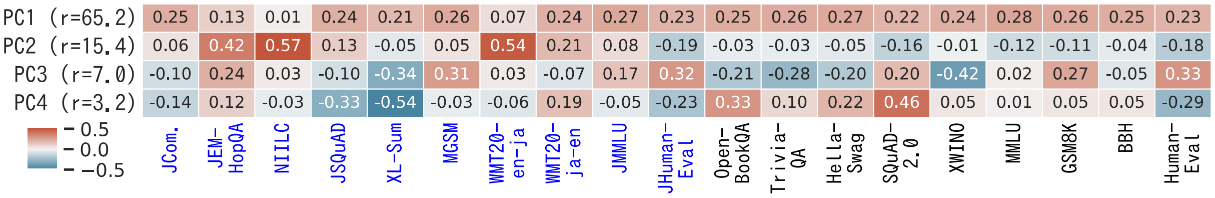
To group benchmarks based on the similarities of LLM performance, we computed a Pearson correlation between two benchmarks a and b. More specifically, let the column vectors $X_{:,a}$ and $X_{:,b}$ represent the array of two benchmarks a and b, we compute the Pearson correlation coefficient $\mathrm{corr}(X_{:,a}, X_{:,b})$. Figure 2 shows the Pearson correlation matrix, revealing two key findings[7]:

First, we observed strong cross-lingual correlations on certain tasks: academic subjects (MMLU vs. JMMLU: 0.91), arithmetic reasoning (GSM8K vs. MGSM: 0.94), and code generation (HumanEval vs. JHumanEval: 0.98). In other words, for these tasks, when LLMs perform well on the English benchmarks, they are also likely to perform well on the corresponding Japanese benchmarks. This suggests that multilinguality outweighs language specificity in these tasks, and that LLMs may generalize abilities acquired through training primarily on English text.

Second, QA tasks about Japanese knowledge (JEMHopQA, NIILC) and an English-Japanese translation task (WMT20-en-ja) exhibit relatively weak correlations with other tasks respectively. In particular, NIILC shows negative correlations with most English tasks, and WMT20-en-ja does nearly no correlations with them. These facts suggest that performance on these tasks may be determined by factors different from those influencing other tasks.

While we observe strong linear correlations between JMMLU, MGSM, and JHumanEval and their English counterparts, given that these are derived from English benchmarks, readers may be concerned that cross-lingual correlations of these benchmarks are overestimated. An straightforward workaround would be to evaluate using random, non-overlapping subsets of instances for each language. Instead of implementing this directly, we

[7] We confirmed that using Spearman's rank correlation produced no significant differences in the findings.



	JCom.	JEM-HopQA	NIILC	JSQuAD	XL-Sum	MGSM	WMT20-en-ja	WMT20-ja-en	JMMLU	JHuman-Eval	Open-BookQA	Trivia-QA	Hella-Swag	SQuAD-2.0	XWINO	MMLU	GSM8K	BBH	Human-Eval
PC1 (r=65.2)	0.25	0.13	0.01	0.24	0.21	0.26	0.07	0.24	0.27	0.23	0.25	0.26	0.27	0.22	0.24	0.28	0.26	0.25	0.23
PC2 (r=15.4)	0.06	0.42	0.57	0.13	-0.05	0.05	0.54	0.21	0.08	-0.19	-0.03	-0.03	-0.05	-0.16	-0.01	-0.12	-0.11	-0.04	-0.18
PC3 (r=7.0)	-0.10	0.24	0.03	-0.10	-0.34	0.31	0.03	-0.07	0.17	0.32	-0.21	-0.28	-0.20	0.20	-0.42	0.02	0.27	-0.05	0.33
PC4 (r=3.2)	-0.14	0.12	-0.03	-0.33	-0.54	-0.03	-0.06	0.19	-0.05	-0.23	0.33	0.10	0.22	0.46	0.05	0.01	0.05	0.05	-0.29

Figure 4: Factor Loadings of principal components for each benchmark ($n = 35$; r is the variance explained; blue: Japanese benchmarks; black: English benchmarks).

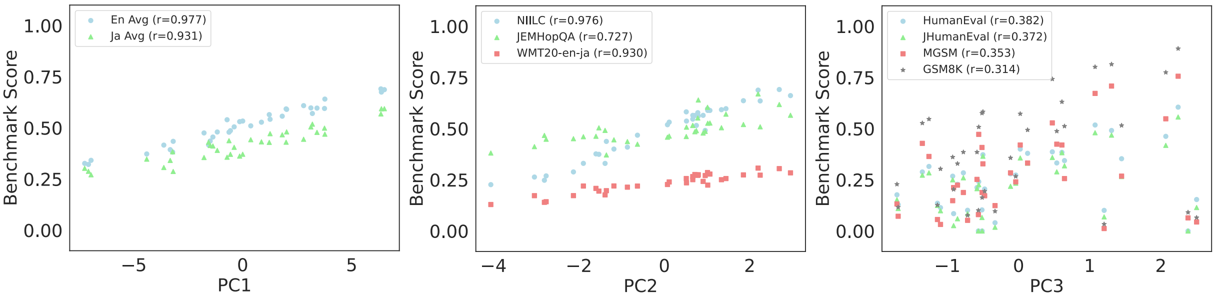


Figure 5: Relationship between principal component scores and raw benchmark scores with significant factor loadings: PC1 vs En/Ja average [left], PC2 vs Japanese knowledge-based QA and En-Ja translation [center], and PC3 vs code-generation and arithmetic reasoning [right] ($n = 35$; r is the pearson correlation coefficient).

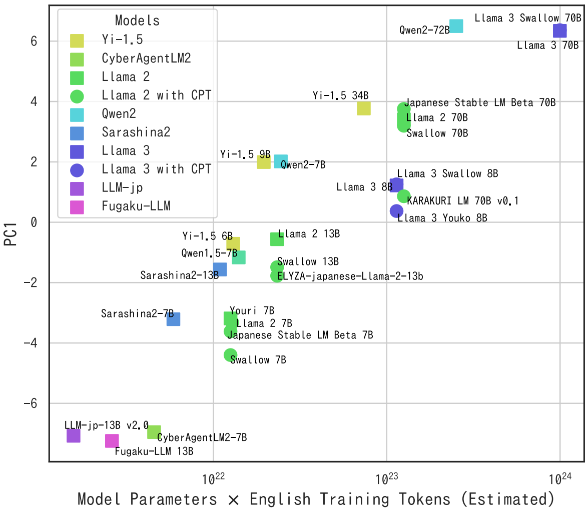


Figure 6: Relationship between the computational budget for English and PC1 scores ($n = 27$).

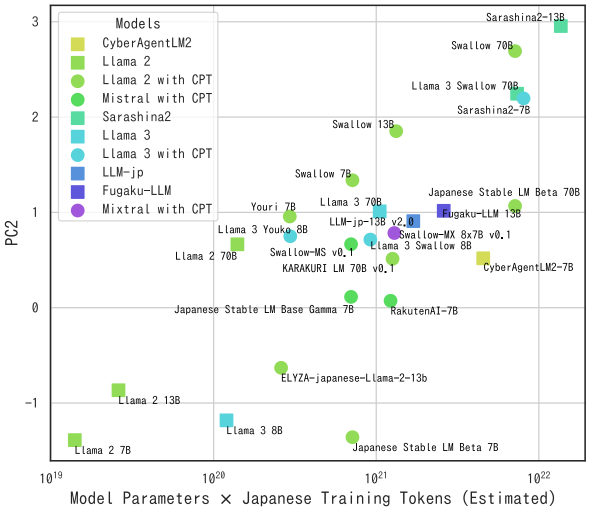


Figure 7: Relationship between the computational budget for Japanese and PC2 scores ($n = 25$).

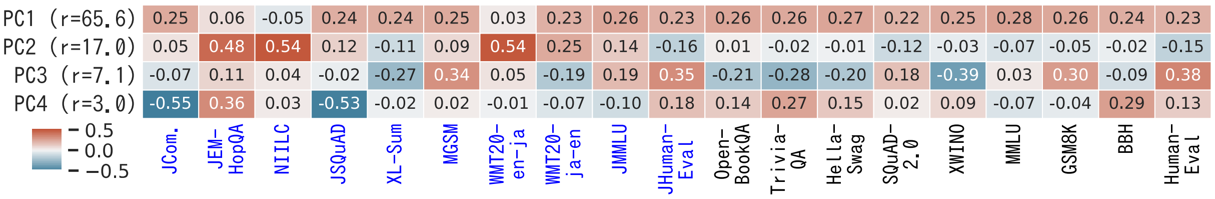
approximated the accuracy variation introduced by such splits using the estimated standard error (SE) following Biderman et al. (2024) and confirmed that impact of the fluctuation by the SE is negligible on the observed linear trends. For example, MGSM has 250 instances, and the SE for an accuracy of 0.5 is approximately $\sqrt{\frac{0.5(1-0.5)}{250}} \approx 0.032$. In contrast, the observed standard deviation of accuracy across LLMs was 0.246, sufficiently larger than the SE.

4.3 Principal Component Analysis (PCA)

We observed benchmark groups from the correlation matrix in the previous subsection. In order to identify ability factors of LLMs, we apply Principal Component Analysis (PCA)[8] to project the task performance into a low-dimensional ability space.

Formally, we first standardize each column of X to have a mean of zero and a standard deviation of one: $\hat{X}$. Next, we perform eigendecomposition of the correlation matrix as $\hat{X}^\top \hat{X} = U \Lambda U^\top$, where

[8] We used the `sklearn.decomposition.PCA()` method from the `scikit-learn` package.



	JCom.	JEM-HopQA	NIILC	JSQuAD	XL-Sum	MGSM	WMT20-en-ja	WMT20-ja-en	JMMLU	JHuman-Eval	Open-BookQA	Trivia-QA	Hella-Swag	SQuAD-2.0	XWINO	MMLU	GSM8K	BBH	Human-Eval
PC1 (r=65.6)	0.25	0.06	-0.05	0.24	0.24	0.25	0.03	0.23	0.26	0.23	0.26	0.26	0.27	0.22	0.25	0.28	0.26	0.24	0.23
PC2 (r=17.0)	0.05	0.48	0.54	0.12	-0.11	0.09	0.54	0.25	0.14	-0.16	0.01	-0.02	-0.01	-0.12	-0.03	-0.07	-0.05	-0.02	-0.15
PC3 (r=7.1)	-0.07	0.11	0.04	-0.02	-0.27	0.34	0.05	-0.19	0.19	0.35	-0.21	-0.28	-0.20	0.18	-0.39	0.03	0.30	-0.09	0.38
PC4 (r=3.0)	-0.55	0.36	0.03	-0.53	-0.02	0.02	-0.01	-0.07	-0.10	0.18	0.14	0.27	0.15	0.02	0.09	-0.07	-0.04	0.29	0.13

Figure 8: Factor loadings of principal components for each benchmark ($n = 20$: only with models trained from scratch; r is the variance explained; blue: Japanese benchmarks; black: English benchmarks).

$U = [u_1, u_2, \ldots, u_D]$, and $u_j \in \mathbb{R}^D$ is the j-th unit-length eigenvector. We then select the top four principal components (PCs), as their cumulative fraction of variance explained (r; contribution ratio) is 90.8% (= 65.2% + 15.4% + 7.0% + 3.2% from PC1 to PC4). We define the eigenvectors corresponding to PC1 to PC4, $U_4 = [u_1, u_2, u_3, u_4] \in \mathbb{R}^{D\times4}$ as the factor loadings and compute corresponding PC scores as $S_4 = \hat{X}U_4$. Given that U is an orthonormal matrix and the total variance explained by PC1–PC4 is approximately 90%, the original matrix can be approximated as the product of PC scores and factor loadings: $\hat{X} \approx S_4 U_4^\top$.

In this way, we decompose standardized benchmark scores $\hat{X}$ into the product of LLM-specific principal component scores (ability factors) $S_4 \in \mathbb{R}^{M\times4}$ in Figure 3 and benchmark-specific factor loadings $U_4 \in \mathbb{R}^{D\times4}$ in Figure 4, which represent the associations between the ability factors and task performances[9].

The first principal component (PC1) has relatively uniform factor loadings. As shown in Figure 5 left, LLMs with higher PC1 scores tend to have higher average benchmark scores in both English and Japanese, suggesting that PC1 represents a general ability factor. It represents the average performance across most benchmark scores, including commonsense and reading comprehension in Japanese. It is also noteworthy that the factor loadings for the three Japanese benchmarks, NIILC, WMT20-en-ja, and JEMHopQA, are relatively small, suggesting that these benchmark scores are more closely associated with a different principal component.

The second principal component (PC2) shows concentrated factor loadings on JEMHopQA, NIILC, and WMT20-en-ja, and relatively small factor loadings on JCommonsenseQA and JSQuAD, indicating the abilities of (encyclopedic) knowledge about Japan and English-Japanese translation. In fact, Figure 3 shows that LLMs pre-trained on Japanese text, such as Swallow and Sarashina2 families, have high PC2 scores, which will be analyzed in detail in § 4.4. Additionally, as shown in Figure 5 center, the higher PC2, the higher benchmark scores on those tasks. For instance, the magin of NIILC accuracy between LLMs with the lowest and highest PC2 scores is approximately 40 points. Considering that PC1 has relatively low factor loadings for these benchmarks, PC2 represents Japanese-specific abilities, such as QA about Japanese knowledge and English-Japanese translation, whereas PC1 represents the general abilities.

The third principal component (PC3) shows concentrated factor loadings on MGSM, GSM8K, JHumanEval, and HumanEval, representing abilities of multilingualism, language-agnostic arithmetic reasoning, and code generation. As shown in Figure 5 right, there is a moderate trend suggesting that higher PC3 score are associated with higher benchmark scores on code-generation and arithmetic-reasoning. While we observe some outliers in the lower right corner, which correspond to LLM-jp-13B v2.0, CyberAgentLM2-7B, and Fugaku-LLM 13B, we think the PC3 scores for these LLMs might be overestimated due to compensation for their excessively low PC1 scores (as seen in Figure 3). This interpretation is supported by the alternative factor analysis using Promax rotation (Appendix C.1), where we observed diminished principal component scores for arithmetic reasoning and code generation in these LLMs.

Finally, the fourth principal component (PC4) shows positive factor loadings for some English benchmarks. However, strong English LLMs, such as Llama-3-70B, do not show higher PC4 scores compared to Japanese LLMs like CyberAgentLM2-7B. In addition, given that the variance explained by PC4 is only 3.2%, PC4 is likely to correspond to residuals that are difficult to interpret in a way tied to specific benchmarks or abilities.

[9] Technically, since the signs and magnitudes of the PC scores and factor loadings are arbitrary, we adjusted the signs for ease of interpretation and normalized the factor loading vectors to have an L_2 norm of 1.

4.4 Scaling Laws between Ability Factors and Computational Budget

In § 4.3, we made two key observations: 1) PC2 represents Japanese ability while PC1 represents a general ability; 2) LLMs pre-trained on Japanese text tend to have higher PC2 scores. Based on these observations, we explore the language-specific scaling laws by examining the log-linear relationship between the computational budgets (§ 3.3) and principal components, which are expected to represent different abilities.

Figure 6 shows the scatter plot with the English computational budget (log scale) and PC1. It reveals that the general ability (PC1) scales with the English computational budget (Pearson's $\rho = 0.916$)[10]. Figure 7 shows the scatter plot with the Japanese computational budget (log scale) and PC2. We can see that the Japanese ability (PC2) moderately scales with the Japanese computational budget ($\rho = 0.779$). We also confirmed that the correlation between PC2 and the English or total computational budget is much weaker ($\rho = 0.164$ and 0.186, respectively). These findings indicate that PC2 and associated Japanese task performances scale with an increase in Japanese training tokens, thereby supporting our claim in § 4.3 that "PC2 represents Japanese ability." Furthermore, we argue that the source of Japanese ability lies in the computational budget allocated to Japanese texts.

4.5 PCA for Japanese LLMs Trained from Scratch

In order to remove the influence of high compute budgets for English LLMs, we excluded LLMs continually pre-trained in Japanese and focused only on 20 LLMs trained from scratch. Figure 8 shows factor loadings of PCs with the 20 LLMs, which also identified similar ability factors to those found in § 4.3. We omit the results of relationships between computational budgets and English and Japanese abilities, but observed the consistent correlations with Figures 6 and 7 (see Figures 13 and 14 in Appendix C.2).

5 Conclusion and Future Work

In this paper, we evaluated the performance of 35 Japanese, English, and Multilingual LLMs on 19 task benchmarks that assess the abilities in both Japanese and English. We then analyzed the cross-task and cross-lingual correlations of benchmark scores, mapped the performance in a low-dimensional ability space, and explored the relationship between ability factors and computational budgets for English and Japanese. The correlation analysis showed strong multilingual abilities in academic knowledge, code generation, and arithmetic reasoning tasks. This suggests that, in order to enhance the abilities of these tasks, there is no strong motivation for using Japanese training data.

The low-dimensional factor analysis using PCA identified three ability factors. PC1 represents the general ability and affects nearly all tasks except for QA about Japanese knowledge and English-Japanese translation. PC1 follows a scaling law with the computational budget for English. Complementing PC1, PC2 represents the ability for QA about Japanese knowledge and English-Japanese translation. Interestingly, PC2 follows a scaling law with the computational budget for Japanese data. Although PC3 represents multilingual abilities in arithmetic reasoning and code generation, we have not reached the point of identifying a scaling law that it follows.

From these analyses, we concluded that the advantage of building local LLMs by training on Japanese text is particularly evident in acquiring local knowledge written in Japanese and enhancing the ability to translate from English. This conclusion is likely to characterize Japanese LLMs.

We consider two directions as future work. First, we plan to extend the analysis with more LLMs and evaluation tasks to discover additional insights. This includes using LLMs with unique designs, for example, Phi family (Li et al., 2023; Abdin et al., 2024), which were trained on synthetic text. We also want to add evaluation tasks such as Japanese logical reasoning and standardized admission exams. The second direction is to extend our analysis and findings to other languages. We believe that the conclusion of this paper can be generalized to: the advantage of building local LLMs by training in a language is acquiring local knowledge written in the language and enhancing the ability to translate from English to the language. This direction is nontrivial because conducting LLM experiments properly requires a deep understanding of the target languages and cultures. We hope this paper serves as a catalyst for analysis in other languages.

[10] The correlation with the logarithm of the total computational budget was slightly higher ($\rho = 0.938$). Still, given the weak correlation with the Japanese computational budget, we concluded that it scales more with the English computational budget.

Limitations

Observational Approach

This study uses an observational approach, relying solely on the evaluation results of the LLMs. We did our best to collect open LLMs and benchmarks that are available as of writing and to evaluate LLMs correctly by ourselves. Still, the findings, including those from factor analysis, may be influenced by the selection of models and evaluation task benchmarks. Although we assessed the statistical error of factor loadings using leave-one-out cross-validation on the analyzed LLMs (see Figure 17 in the appendix) and confirmed that the standard deviations were small relative to the absolute values, this does not guarantee that our findings remain valid as a new best practice for designing LLMs emerges.

English Predominance in the Training Data

In § 4.4, we discussed that the general ability was acquired through training on English text, based on the solid log-linear relationship between the computational budget for English and PC1. However, this could be a limitation of our observational approach. Specifically, most of the LLMs evaluated in this study were trained on the English-centric data, with at least half of the data in English. When we have LLMs predominantly trained in Japanese, different findings might emerge.

Generalization of Findings beyond Japanese

This study focuses solely on Japanese LLMs as an instance of non-English and local LLMs. It is unclear whether the findings are applicable to other non-English LLMs. However, we want to emphasize that *even evaluating LLMs accurately in languages that we are familiar with is not an easy task.* Some LLMs scored zero for a benchmark, and we ended up debugging the problem only to find that they require a special token or even a line break in the prompt to obtain a valid generation. Implementation details (e.g., probabilistic decoding) also affected the performance of LLMs in downstream tasks. Technology has not yet advanced to the point where simply submitting a local LLM to a leaderboard yields reliable evaluation results effortlessly. Therefore, accurately conducting evaluation experiments with a lot of LLMs and benchmarks, as done in this study, requires a deep understanding of the target language, which sets a high bar for us.

Ethical Considerations

This study does not evaluate the safety aspects of LLMs, such as harmlessness or honesty (Askell et al., 2021), which are considered to be largely shaped by pre-training data. The same applies when developing local LLMs — they are likely to absorb social group-specific biases (Yanaka et al., 2024), stereotypes, and racism. Consequently, there is a concern that we may be overlooking an inconvenient side effect: it might be unavoidable for local LLMs to reinforce social biases specific to the target language.

Acknowledgments

(Removed for blind review)

References

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, and Harkirat Behl et al. 2024. Phi-3 technical report: A highly capable language model locally on your phone. arXiv:2404.14219.

01. AI, Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Heng Li, Jiangcheng Zhu, and Jianqun Chen et al. 2024. Yi: Open foundation models by 01.AI. arXiv:2403.04652.

Amanda Askell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, Andy Jones, Nicholas Joseph, Ben Mann, and Nova DasSarma et al. 2021. A general language assistant as a laboratory for alignment. arXiv:2112.00861.

Loïc Barrault, Magdalena Biesialska, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Matthias Huck, and Eric et al. Joanis. 2020. Findings of the 2020 conference on machine translation. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1–55.

Loubna Ben Allal, Niklas Muennighoff, Logesh Kumar Umapathi, Ben Lipkin, and Leandro von Werra. 2022. A framework for the evaluation of code generation models. `https://github.com/bigcode-project/bigcode-evaluation-harness`.

Gábor Berend. 2022. Combating the curse of multilinguality in cross-lingual WSD by aligning sparse contextualized word representations. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2459–2471.

Stella Biderman, Hailey Schoelkopf, Lintang Sutawika, Leo Gao, Jonathan Tow, Baber Abbasi, Alham Fikri Aji, Pawan Sasanka Ammanamanchi, Sidney Black, Jordan Clive, Anthony DiPofi, Julen Etxaniz, Benjamin Fattori, Jessica Zosa Forde, Charles Foster, Jeffrey Hsu, Mimansa Jaiswal, Wilson Y. Lee, Haonan Li, Charles Lovering, Niklas Muennighoff, Ellie Pavlick, Jason Phang, Aviya Skowron, Samson Tan, Xiangru Tang, Kevin A. Wang, Genta Indra Winata, François Yvon, and Andy Zou. 2024. Lessons from the trenches on reproducible evaluation of language models. arXiv:2112.00861.

Eleftheria Briakou, Colin Cherry, and George Foster. 2023. Searching for needles in a haystack: On the role of incidental bilingualism in PaLM's translation capability. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9432–9452.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, and Yidong et al. Wang. 2024. A survey on evaluation of large language models. *Association for Computing Machinery Transactions on Intelligent Systems and Technology*, 15(3):1–45.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, and Greg Brockman et al. 2021. Evaluating large language models trained on code. arXiv:2107.03374.

ChangSu Choi, Yongbin Jeong, Seoyoon Park, Inho Won, HyeonSeok Lim, SangMin Kim, Yejee Kang, Chanhyuk Yoon, Jaewan Park, and Yiseul et al. Lee. 2024. Optimizing language augmentation for multilingual large language models: A case study on Korean. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation*, pages 12514–12526.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, and Reiichiro Nakano et al. 2021. Training verifiers to solve math word problems. arXiv:2110.14168.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451.

Yiming Cui, Ziqing Yang, and Xin Yao. 2024. Efficient and effective text encoding for Chinese LLaMA and Alpaca. arXiv:2304.08177.

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

Kazuki Fujii, Taishi Nakamura, Mengsay Loem, Hiroki Iida, Masanari Ohi, Kakeru Hattori, Hirai Shota, Sakae Mizuki, Rio Yokota, and Naoaki Okazaki. 2024. Continual pre-training for cross-lingual LLM adaptation: Enhancing Japanese language capabilities. In *Proceedings of the 1st Conference on Language Modeling*.

Zishan Guo, Renren Jin, Chuang Liu, Yufei Huang, Dan Shi, Supryadi, Linhao Yu, Yan Liu, Jiaxuan Li, and Bojian Xiong et al. 2023. Evaluating large language models: A comprehensive survey. arXiv:2310.19736.

Namgi Han, Nobuhiro Ueda, Masatoshi Otake, Satoshi Katsumata, Keisuke Kamata, Hirokazu Kiyomaru, Takashi Kodama, Saku Sugahara, Bowen Chen, and Hiroshi Matsuda et al. 2024. llm-jp-eval: Automatic evaluation tool for Japanese large language models [llm-jp-eval: 日本語大規模言語モデルの自動評価ツール] (in Japanese). In *Proceedings of the 30th Annual Meeting of the Association for Natural Language Processing*, pages 2085–2089.

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

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

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, and Aidan et al. Clark. 2022. An empirical analysis of compute-optimal large language model training. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, pages 1–15.

Oskar Holmström, Jenny Kunz, and Marco Kuhlmann. 2023. Bridging the resource gap: Exploring the efficacy of English and multilingual LLMs for Swedish. In *Proceedings of the Second Workshop on Resources and Representations for Under-Resourced Languages and Domains*, pages 92–110.

Ai Ishii, Naoya Inoue, and Satoshi Sekine. 2023. Construction of a Japanese multi-hop QA dataset for QA systems capable of explaining the rationale [根拠を説明可能な質問応答システムのための日本語マルチホップQAデータセット構築] (in Japanese). In *the 29th Annual Meeting of Japanese Association for Natural Language Processing (NLP2023)*, pages 2088–2093.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, and Lucile Saulnier et al. 2023. Mistral 7B. arXiv:2310.06825.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, and Florian Bressand et al. 2024. Mixtral of experts. arXiv:2401.04088.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 1601–1611.

KARAKURI Inc. 2024. KARAKURI LM 70B v0.1. Hugging Face: karakuri-ai/karakuri-lm-70b-v0.1.

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.

Celio Larcher, Marcos Piau, Paulo Finardi, Pedro Gengo, Piero Esposito, and Vinicius Caridá. 2023. Cabrita: closing the gap for foreign languages. arXiv:2308.11878.

Yuanzhi Li, Sébastien Bubeck, Ronen Eldan, Allie Del Giorno, Suriya Gunasekar, and Yin Tat Lee. 2023. Textbooks are all you need II: phi-1.5 technical report. arXiv:2309.05463.

LLM-jp, Akiko Aizawa, Eiji Aramaki, Bowen Chen, Fei Cheng, Hiroyuki Deguchi, Rintaro Enomoto, Kazuki Fujii, Kensuke Fukumoto, and Takuya Fukushima et al. 2024. LLM-jp: A cross-organizational project for the research and development of fully open Japanese LLMs. arXiv:2407.03963.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391.

Jinjie Ni, Fuzhao Xue, Xiang Yue, Yuntian Deng, Mahir Shah, Kabir Jain, Graham Neubig, and Yang You. 2024. Mixeval: Deriving wisdom of the crowd from LLM benchmark mixtures. arXiv:2406.06565.

Kunat Pipatanakul, Phatrasek Jirabovonvisut, Potsawee Manakul, Sittipong Sripaisarnmongkol, Ruangsak Patomwong, Pathomporn Chokchainant, and Kasima Tharnpipitchai. 2023. Typhoon: Thai large language models. arXiv:2312.13951.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 784–789.

RakutenGroup, Aaron Levine, Connie Huang, Chenguang Wang, Eduardo Batista, Ewa Szymanska, Hongyi Ding, Hou Wei Chou, Jean-François Pessiot, and Johanes Effendi et al. 2024. RakutenAI-7B: Extending large language models for Japanese. arXiv:2403.15484.

Yangjun Ruan, Chris J. Maddison, and Tatsunori Hashimoto. 2024. Observational scaling laws and the predictability of language model performance. arXiv:2405.10938.

Akira Sasaki, Masato Hirakawa, Shintaro Horie, Tomoaki Nakamura, Sam Passaglia, and Daisuke Oba. 2023. ELYZA-japanese-Llama-2-13b. `https://huggingface.co/elyza/ELYZA-japanese-Llama-2-13b`.

Yui Sato, Shiho Takano, Teruno Kajiura, and Kimiro Kuramitsu. 2024. Do large language models transfer knowledge across languages through additional Japanese training? [llmは日本語追加学習により言語間知識転移を起こすのか？] (in Japanese). In *Proceedings of the 30th Annual Meeting of the Association for Natural Language Processing*, pages 2897–2901.

Kei Sawada, Tianyu Zhao, Makoto Shing, Kentaro Mitsui, Akio Kaga, Yukiya Hono, Toshiaki Wakatsuki, and Koh Mitsuda. 2024. Release of pre-trained models for the Japanese language. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 13898–13905.

Satoshi Sekine. 2003. Development of a question answering system targeting encyclopedias [百科事典を対象とした質問応答システムの開発] (in Japanese). In *Proceedings of the 9th Annual Meeting of the Association for Natural Language Processing*, pages 637–640.

Freda Shi, Mirac Suzgun, Markus Freitag, Xuezhi Wang, Suraj Srivats, Soroush Vosoughi, Hyung Won Chung, Yi Tay, Sebastian Ruder, and Denny Zhou et al. 2023. Language models are multilingual chain-of-thought reasoners. In *Proceedings of the Eleventh International Conference on Learning Representations*.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, and Denny et al. Zhou. 2023. Challenging BIG-bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics*, pages 13003–13051.

Qwen Team. 2024. Introducing Qwen1.5.

Atula Tejaswi, Nilesh Gupta, and Eunsol Choi. 2024. Exploring design choices for building language-specific llms. In *Findings of the Association for Computational Linguistics*, pages 10485–10500.

Alexey Tikhonov and Max Ryabinin. 2021. It's all in the heads: Using attention heads as a baseline for cross-lingual transfer in commonsense reasoning. In *Findings of the Association for Computational Linguistics*, pages 3534–3546.

Anthony Tiong, Junqi Zhao, Boyang Li, Junnan Li, Steven Hoi, and Caiming Xiong. 2024. What are we measuring when we evaluate large vision-language models? an analysis of latent factors and biases. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3427–3454.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, and Shruti Bhosale et al. 2023. Llama 2: Open foundation and fine-tuned chat models. arXiv:2307.09288.

Hitomi Yanaka, Namgi Han, Ryoma Kumon, Jie Lu, Masashi Takeshita, Ryo Sekizawa, Taisei Kato, and Hiromi Arai. 2024. Analyzing social biases in Japanese large language models. arXiv:2406.02050.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, and Fei Huang et al. 2024. Qwen2 technical report. arXiv:2407.10671.

Ziqi Yin, Hao Wang, Kaito Horio, Daisuke Kawahara, and Satoshi Sekine. 2024. Investigating the relationship between prompt politeness and large language model performance [プロンプトの丁寧さと大規模言語モデルの性能の関係検証] (in Japanese). In *Proceedings of the 30th Annual Meeting of the Association for Natural Language Processing*, pages 1803–1808.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800.

Xiang Zhang, Senyu Li, Bradley Hauer, Ning Shi, and Grzegorz Kondrak. 2023. Don't trust ChatGPT when your question is not in English: A study of multilingual abilities and types of LLMs. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7915–7927.

A Details of the Experimental Setup

A.1 Evaluated Models

Table 1 shows a list of LLMs evaluated in this study. The table includes the name, the number of active parameters during inference, the base model from which the model was continually pre-trained, the language distribution of the training corpus, the total number of training tokens, the reported or estimated number of training tokens in English and Japanese, and the reference of each model. § A.3 explains the method used to estimate the number of language-specific training tokens. *CPT* stands for *continual pre-training*.

A.2 Evaluation Tasks and Benchmarks

Table 2 provides an overview of the evaluation benchmarks used in this study. The table includes the benchmark name, a brief description, the language of the task, the metric for scoring the model's output, the experimental setting (e.g., few-shot, zero-shot, chain-of-thought), and the reference of each benchmark. The scale of evaluation metrics is normalized between 0 and 1, and *EM* means *exact match*.

A.3 Estimating the Number of Training Tokens

The numbers of language-specific training tokens (in billions) were either obtained from or calculated based on official sources such as technical reports, release documents, or model cards. When an exact number was unavailable in the source, we used the following estimates:

- Ratio of Japanese training tokens:
  - Llama 2, Llama 3: 0.1%
  - Mistral, Mixtral: 0%
  - Full-scratch Japanese LLMs: 50%
  - Japanese LLMs with CPT: 100%
- Ratio of English training tokens:
  - Qwen1.5, Qwen2: 50%
  - Yi-1.5: 70%
  - Llama 2: 89.7%
  - Llama 3: 95%

A symbol '–' in Table 1 indicates that the number could not be obtained or estimated despite our best efforts. We excluded these LLMs from the analysis of the scaling laws in § 4.4.

A.4 Evaluation Framework

Table 3 reports a list of evaluation frameworks used in this study. The table shows the framework name, a brief description, and the reference of the framework. We slightly customized these evaluation frameworks to cover benchmarks that are not officially supported and to implement workarounds for LLMs; for example, some LLMs require special tokens or line breaks in the prompt to generate valid outputs. We will release the customized implementation upon acceptance.

A.5 Details of LLM Grouping

Table 4 shows the breakdown of LLM groups used in Figure 1.

B Analysis of the Evaluation Results

This section presents detailed observations that complement the explanation in § 4.1.

B.1 Performance Difference between the Pre-trained Languages

Figure 1 reveals a notable observation: the scores of Japanese LLMs pre-trained from scratch (the blue box) are consistently lower than those of continually pre-trained models. This may be due to the relatively small number of parameters of the LLMs in this category (e.g. CyberAgentLM2-7B, Sarashina2-7B, Fugaku-LLM 13B), as well as the limited training budget (i.e., number of training tokens) available for developing LLMs from scratch. This highlights a challenge in developing local LLMs in Japan.

Additionally, compared to other groups, multilingual LLMs (the black box) performed significantly better in arithmetic reasoning (MGSM and GSM8K) and code generation (JHumanEval and HumanEval) tasks. However, we believe that this does not reflect the overall strength of multilingual LLMs, but rather the strengths of Qwen family (Yang et al., 2024), which represents three out of four LLMs in this group.

B.2 Variations in Task Scores

Figure 1 highlights tasks with both high and low score variances. Tasks with low score variances can be grouped into two categories:

1. Benchmarks evaluated with n-gram based metrics (e.g. WMT20-ja-en and WMT20-en-ja with BLEU, and XL-Sum with ROUGE-2).

2. Tasks requiring essential skills (e.g. JSQuAD and SQuAD2.0 (reading comprehension), and OpenBookQA and XWINO (commonsense)).

In contrast, tasks with high score variances can be grouped into two categories:

1. Tasks requiring specific capabilities (e.g. MGSM, GSM8K (arithmetic reasoning), JHumanEval and HumanEval (code generation))

2. Knowledge-intensive tasks (e.g. NIILC, JMMLU, MMLU, and TriviaQA)

The scores for these tasks heavily depend on whether a model possesses the necessary capabilities or specialized knowledge, which leads to a greater variance.

C Robustness Check of Findings Obtained from Experimental Results

To test the robustness of the findings presented in § 4, we conducted two additional analyses using different methods and settings: the use of maximum likelihood estimation and Promax rotation[13] instead of PCA (in § 4.3); and exclusion of continually pre-trained models to focus on models trained from scratch.

C.1 Maximum Likelihood Estimation and Promax Rotation

Figure 10 presents factor loadings with Promax rotation applied. This figure reveals two factors similar to those identified in § 4.3: ability factor for arithmetic reasoning and code generation (Factor 2 for PC3), and ability factor Japanese (Factor 3 for PC2). In contrast, the first factor (Factor 1) seems to represent English ability, not the general ability (PC1), since the loading scores are strongly positive on the English task benchmarks such as OpenBookQA, TriviaQA, HellaSwag, and XWINO.

Additionally, the fourth factor (Factor 4) seems to be a distinct ability factor for Japanese at first glance since the loading scores are strongly positive on two Japanese task benchmarks (JCom. and JSQuAD). However, the correlation coefficient with the logarithm of the computational budget for Japanese is as small as 0.241, much lower than that of the computational budget for English (0.788). Figure 9 shows small Factor 4 scores on Japanese LLMs, such as Llama 3 Youko 8B, Japanese Stable LM Beta 7B, CyberAgentLM2-7B, LLM-jp-13B v2.0 and Fugaku-LLM 13B. Even strong Japanese LLMs (e.g., Llama 3 Swallow 70B, Japanese Stable LM Base Gamma 7B) do not show high scores compared to non-Japanese LLMs. Therefore, the fourth factor should be considered as a residual that is difficult to interpret; therefore, commonsense tasks and reading comprehension do not determine Japanese abilities.

To sum, these results confirm two similar factors to those identified in § 4.3 (an ability factor for arithmetic reasoning and code generation, and a Japanese ability factor) and two unique factors (an English ability factor and a residual factor).

C.2 Analysis with only Full-scratch Models

We removed continually pre-trained LLMs, which are categorized as *LLMs continually pre-trained on Japanese text* in Table 4 and conducted the same analysis as in § 4.2 to § 4.4.

Figure 15 shows the Pearson correlation matrix of benchmark scores. The figure reveals that JEMHopQA, NIILC (QA about Japanese knowledge) and WMT20-en-ja (English-Japanese translation) are weakly correlated with other tasks. In addition, the figure shows strong correlations across languages in benchmarks of arithmetic reasoning (GSM8K vs. MGSM), academic subjects (MMLU vs. JMMLU), and code generation (HumanEval vs. JHumanEval). These findings are consistent with those identified with continually pre-trained LLMs in § 4.2.

Figure 16 shows the factor loadings for each task benchmark. The figure highlights four factors: a general ability factor with uniform scores on each benchmark (PC1); a Japanese ability factor with high scores on JEMHopQA, NIILC, and WMT20-en-ja (PC2); an ability factor for arithmetic reasoning and code generation with high scores on HumanEval, JHumanEval, MSGM, and GSM8K (PC3); and a residual factor that is difficult to interpret (PC4). These observations are consistent with those obtained with continually pre-trained LLMs in § 4.3.

Lastly, we examined the relationship between the computational budget for English and PC1 (Figure 13) and the one between the computational budget for Japanese and PC2 (Figure 14). Figure 13 exhibits a strong positive correlation between PC1 (general ability) and computational budget for English ($\rho = 0.923$), and Figure 14

[13] We used the `factor_analyzer.FactorAnalyzer()` and `factor_analyzer.Rotator()` method from the `factor_analyzer` package.

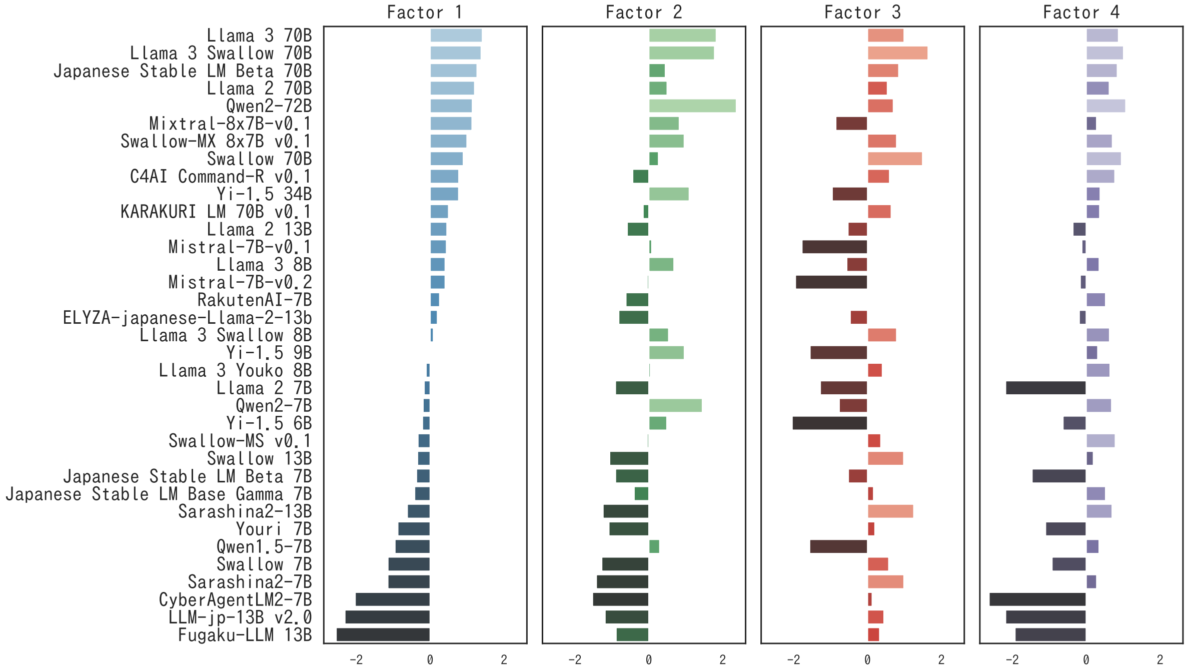
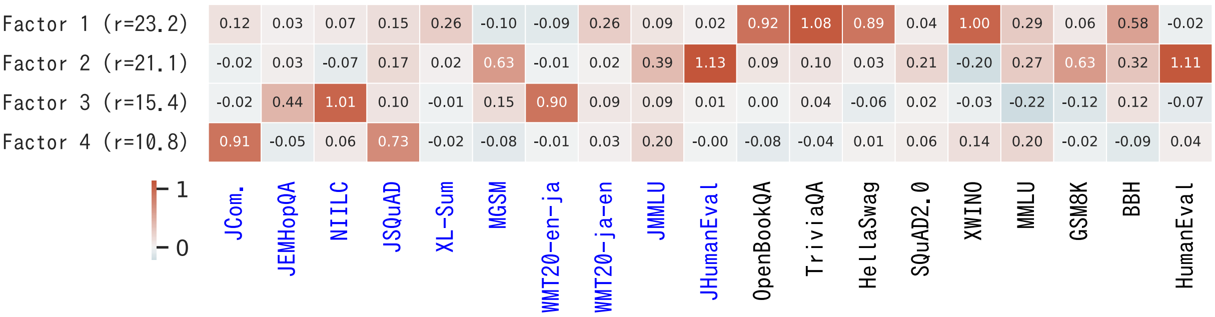


Figure 9: Factor scores for each model with Promax rotation applied.



	JCom.	JEMHopQA	NIILC	JSQuAD	XL-Sum	MGSM	WMT20-en-ja	WMT20-ja-en	JMMLU	JHumanEval	OpenBookQA	TriviaQA	HellaSwag	SQuAD2.0	XWINO	MMLU	GSM8K	BBH	HumanEval
Factor 1 (r=23.2)	0.12	0.03	0.07	0.15	0.26	-0.10	-0.09	0.26	0.09	0.02	0.92	1.08	0.89	0.04	1.00	0.29	0.06	0.58	-0.02
Factor 2 (r=21.1)	-0.02	0.03	-0.07	0.17	0.02	0.63	-0.01	0.02	0.39	1.13	0.09	0.10	0.03	0.21	-0.20	0.27	0.63	0.32	1.11
Factor 3 (r=15.4)	-0.02	0.44	1.01	0.10	-0.01	0.15	0.90	0.09	0.09	0.01	0.00	0.04	-0.06	0.02	-0.03	-0.22	-0.12	0.12	-0.07
Factor 4 (r=10.8)	0.91	-0.05	0.06	0.73	-0.02	-0.08	-0.01	0.03	0.20	-0.00	-0.08	-0.04	0.01	0.06	0.14	0.20	-0.02	-0.09	0.04

Figure 10: Factor loadings by task with Promax rotation applied ($n = 35$; r represents a contribution; blue and black colors correspond to Japanese and English task benchmarks, respectively).

indicates a moderate positive correlation between PC2 (Japanese ability) and computation budget for Japanese ($\rho = 0.779$). These relationships are the same as those confirmed with continually pre-trained LLMs in § 4.4.

In this way, we could confirm the findings observed in § 4.2 to § 4.4 even with the LLMs built from scratch, which indicates the robustness of the findings against the construction methods of LLMs.

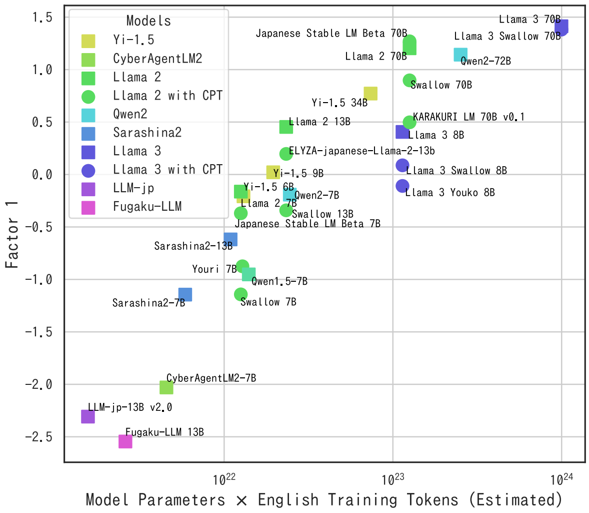


Figure 11: Relationship between the computational budget for English and Factor 1 ($n = 27$).

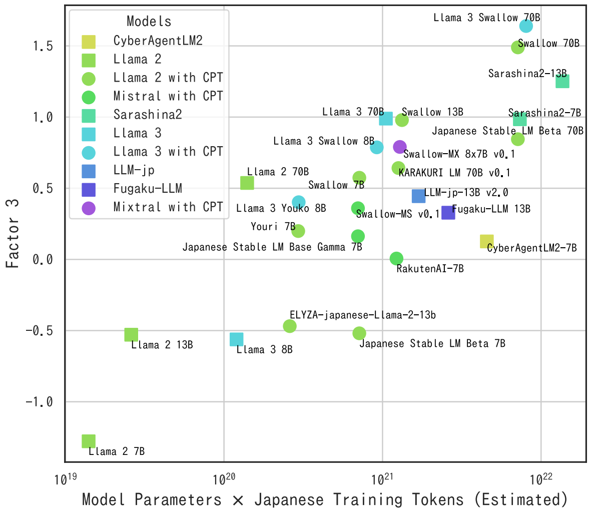


Figure 12: Relationship between the computational budget for Japanese and Factor 3 ($n = 27$).

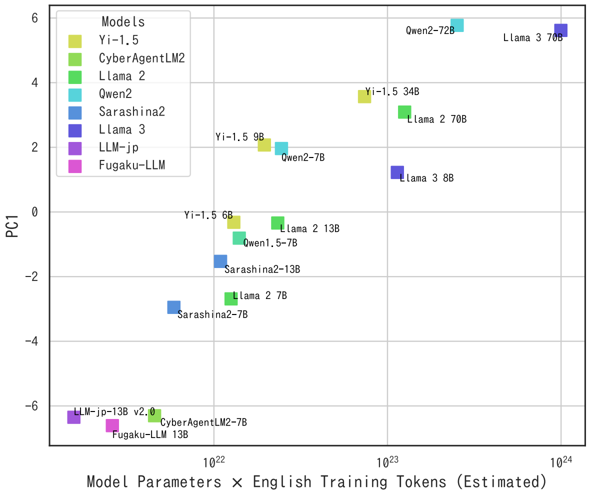


Figure 13: Relationship between the computational budget for English and PC1 ($n = 16$; only with models trained from scratch).

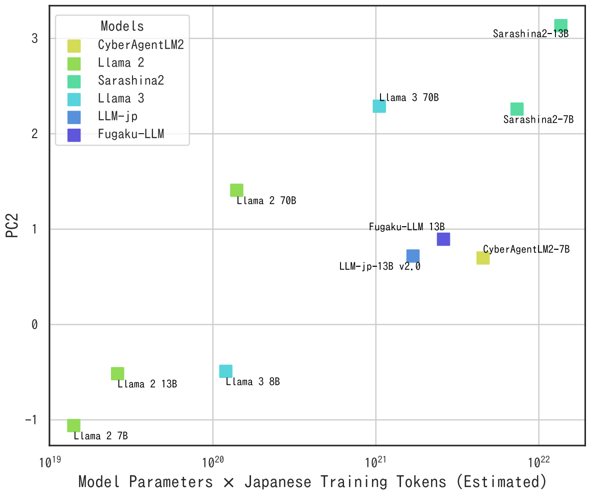


Figure 14: Relationship between the computational budget for Japanese and PC2 ($n = 10$; only with models trained from scratch).

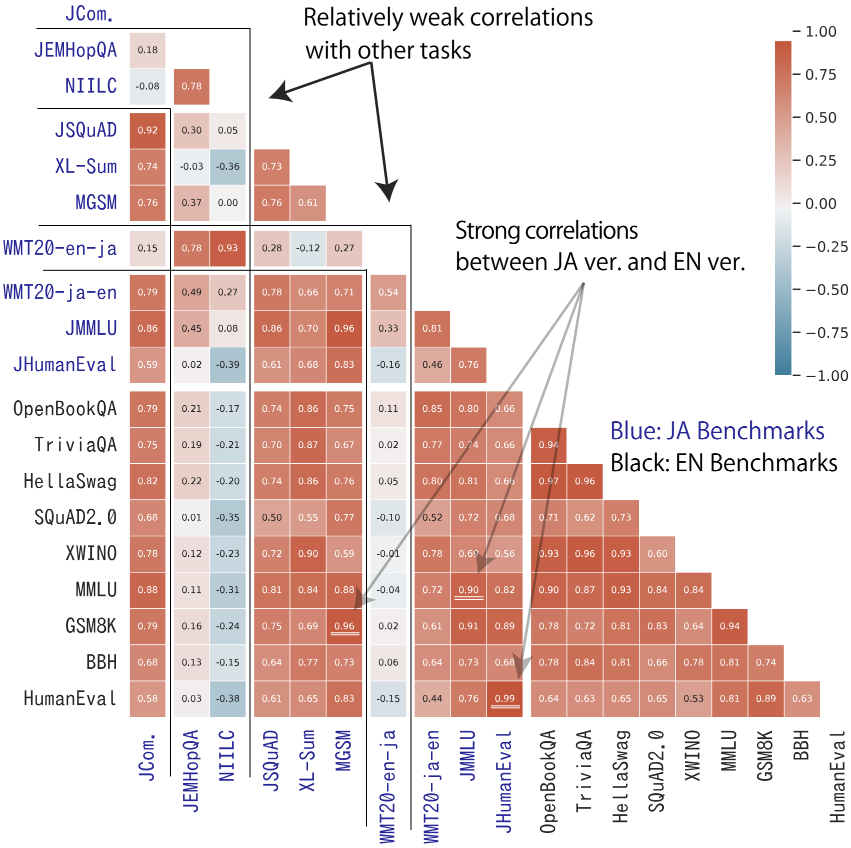


Figure 15: Pearson correlation matrix among benchmark scores ($n = 20$; only with models trained from scratch).

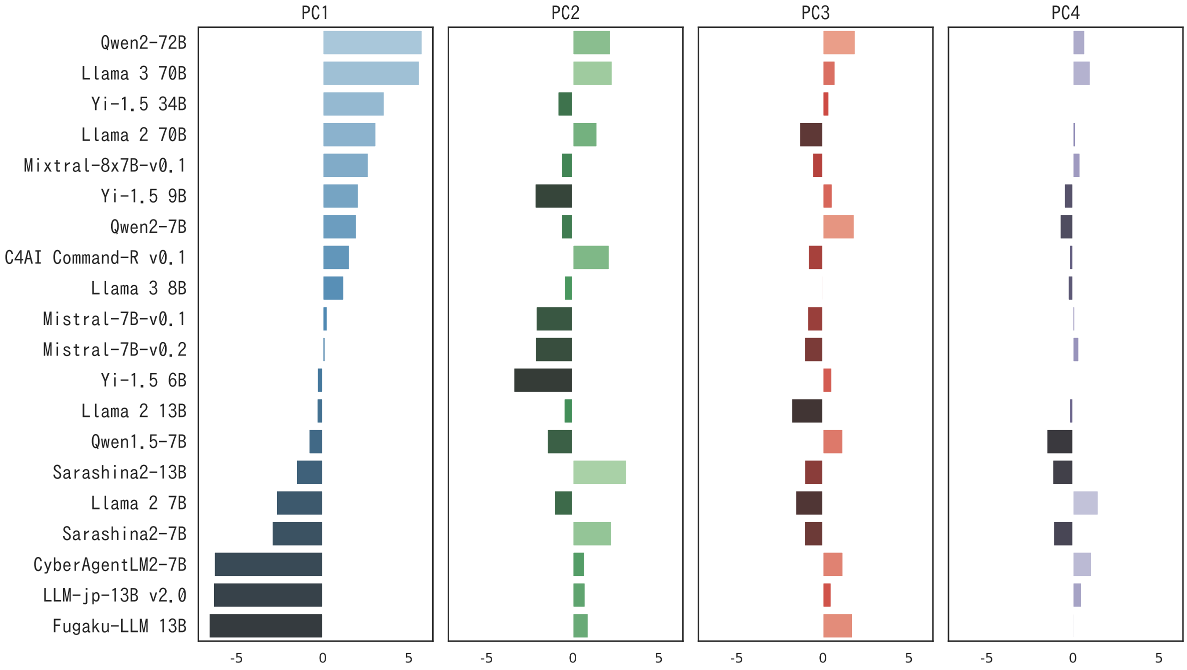


Figure 16: Principal component scores for each model ($n = 20$; only with models trained from scratch).

Table 1: List of evaluated LLMs (the number of tokens is in billions [Bil], including estimates).

Model name	Num of params	Construction method	Source of CPT	Corpus	Training tokens	EN tokens	JA tokens	Reference
Yi-1.5 6B	6	PT	–	ZH,EN, Code	3600	2170	–	AI et al. (2024)
CyberAgentLM2-7B	7	PT	–	JA,EN	1300	650	650	cyberagent/calm2-7b
Japanese Stable LM Base Gamma 7B	7	CPT	Mistral-7B-v0.1	JA,EN	–	–	100	stabilityai/japanese-stablelm-base-gamma-7b
Japanese StableLM Beta 7B	7	CPT	Llama2 7B	JA,EN	2100	1794	102	stabilityai/japanese-stablelm-base-beta-7b
Llama 2 7B	7	PT	–	EN	2000	1794	2	Touvron et al. (2023)
Mistral-7B-v0.1	7	PT	–	EN	–	–	–	Jiang et al. (2023)
Mistral-7B-v0.2	7	PT	–	EN	–	–	–	Jiang et al. (2023)
Qwen1.5-7B	7	PT	–	–	4000	2000	–	Team (2024)
Qwen2-7B	7	PT	–	ZH,EN, Code+27	7000	3500	–	Yang et al. (2024)
RakutenAI-7B	7	CPT	Mistral-7B-v0.1	JA,EN	–	–	175	RakutenGroup et al. (2024)
Sarashina2-7B	7	PT	–	JA,EN	2100	840	1050	sbintuitions/sarashina2-7b
Swallow 7B	7	CPT	Llama2 7B	JA,EN	2100	1794	102	Fujii et al. (2024)
Swallow-MS v0.1	7	CPT	Mistral-7B-v0.1	JA,EN, Code	–	–	100	Fujii et al. (2024)
Youri 7B	7	CPT	Llama2 7B	JA,EN	2040	1834	42	Sawada et al. (2024)
Llama 3 8B	8	PT	–	EN	15000	14250	15	Dubey et al. (2024)
Llama 3 Swallow 8B	8	CPT	Llama3 8B	JA,EN, Code	15100	14250	115	Fujii et al. (2024)
Llama 3 Youko 8B	8	CPT	Llama3 8B	JA,EN	15022	14250	37	Sawada et al. (2024)
Yi-1.5 9B	9	PT	–	ZH,EN, Code	3100	2170	–	AI et al. (2024)
ELYZA-japanese-Llama-2-13b	13	CPT	Llama2 13B	JA	2018	1794	20	Sasaki et al. (2023)
Fugaku-LLM 13B	13	PT	–	JA,EN	400	200	200	Fugaku-LLM/Fugaku-LLM-13B
Llama 2 13B	13	PT	–	EN	2000	1794	2	Touvron et al. (2023)
LLM-jp-13B v2.0	13	PT	–	JA,EN, Code	260	120	130	LLM-jp et al. (2024)
Sarashina2-13B	13	PT	–	JA,EN	2100	840	1050	sbintuitions/sarashina2-13b
Swallow 13B	13	CPT	Llama2 13B	JA,EN	2100	1794	102	Fujii et al. (2024)
Yi-1.5 34B	34	PT	–	ZH,EN, Code	3100	2170	–	AI et al. (2024)
C4AI Command-R v0.1	35	PT	–	JA,EN, ZH+8	–	–	–	CohereForAI/c4ai-command-r-v01
Mixtral-8x7B-v0.1	12.879	PT	–	EN	–	–	–	Jiang et al. (2024)
Swallow-MX 8x7B v0.1	12.879	CPT	Mixtral-8x7B-Instruct-v0.1	JA,EN	–	–	100	Fujii et al. (2024)
Japanese Stable LM Beta 70B	70	CPT	Llama2 70B	JA,EN	2100	1794	102	stabilityai/japanese-stablelm-base-beta-70b
KARAKURI LM 70B v0.1	70	CPT	Llama2 70B	JA,EN	2016	1794	18	KARAKURI Inc. (2024)
Llama 2 70B	70	PT	–	EN	2000	1794	2	Touvron et al. (2023)
Llama 3 70B	70	PT	–	EN	15000	14250	15	Dubey et al. (2024)
Llama 3 Swallow 70B	70	CPT	Llama3 70B	JA,EN, Code	15100	14250	115	Fujii et al. (2024)
Swallow 70B	70	CPT	Llama2 70B	JA,EN	2100	1794	102	Fujii et al. (2024)
Qwen2-72B	72	PT	–	ZH,EN, Code+27	7000	3500	–	Yang et al. (2024)

Table 2: List of benchmarks used for evaluation.

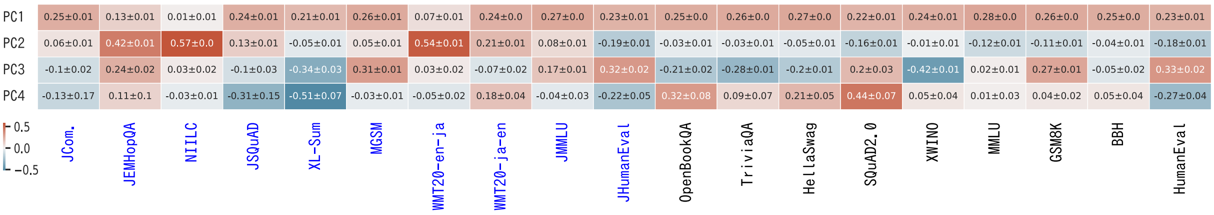
Name	Description	Lang.	Eval. metric[11,12]	Exp. setup	Reference
JcommonsenseQA (JCom.)	Multiple-choice questions with 5 options based on a knowledge base	JA	Acc.	4-shot	Kurihara et al. (2022)
JEMHopQA	Free-form question answering to evaluate knowledge and reasoning ability	JA	Char F1	4-shot	Ishii et al. (2023)
NIILC	Free-form question answering where answers can be obtained from an encyclopedia	JA	Char F1	4-shot	Sekine (2003)
JSQuAD	Free-form question answering on Wikipedia articles	JA	Char F1	4-shot	Kurihara et al. (2022)
XL-Sum	Generating summaries from BBC articles	JA	ROUGE-2	1-shot	Hasan et al. (2021)
MGSM	Japanese translation of the primary school math word problem dataset (GSM8K)	JA	Acc. (EM)	4-shot	Shi et al. (2023)
WMT20(en-ja)	English-Japanese translation of news articles	JA	BLEU	4-shot	Barrault et al. (2020)
WMT20(ja-en)	Japanese-to-English translation of news articles	JA	BLEU	4-shot	Barrault et al. (2020)
JMMLU	Japanese translation of the multiple-choice benchmark MMLU (53 subjects)	JA	Acc.	5-shot	Yin et al. (2024)
JHumanEval	Japanese translation of HumanEval	JA	pass@1	0-shot 10 trials	Sato et al. (2024)
OpenBookQA	Multiple-choice questions based on scientific knowledge and common sense	EN	Acc.	4-shot	Mihaylov et al. (2018)
TriviaQA	Free-form question answering based on trivia knowledge	EN	Acc. (EM)	4-shot	Joshi et al. (2017)
HellaSwag	Multiple-choice questions to predict the next event	EN	Acc.	4-shot	Zellers et al. (2019)
SQuAD2	Free-form question answering based on a supporting document	EN	Acc. (EM)	4-shot	Rajpurkar et al. (2018)
XWINO	Binary-choice questions to identify the antecedent of a pronoun in a sentence	EN	Acc.	4-shot	Tikhonov and Ryabinin (2021)
MMLU	Multiple-choice questions across 57 subjects	EN	Acc.	5-shot	Hendrycks et al. (2021)
GSM8K	Primary school math word problem dataset	EN	Acc. (EM)	4-shot	Cobbe et al. (2021)
BBH	23 challenging tasks from the BIG-Bench dataset	EN	Acc. (EM)	3-shot CoT	Suzgun et al. (2023)
HumanEval	Evaluation of code generation ability via unit tests	EN	pass@1	0-shot 10 trials	Chen et al. (2021)

Table 3: List of evaluation frameworks.

Name	Description	Reference
LLM-jp eval (1.3.0)	Automatic evaluation tool for Japanese LLMs	Han et al. (2024)
JP Language Model Evaluation Harness (commit #9b42d41)	An evaluation framework for Japanese LLMs	zenodo.10256836
Language Model Evaluation Harness (0.4.2)	An evaluation framework for LLMs	zenodo.10256836
Code Generation LM Evaluation Harness (commit #0261c52)	An evaluation framework for code generation task	Ben Allal et al. (2022)

Table 4: Breakdown of LLM groups used in Figure 1.

Category	Models
Japanese LLMs pre-trained from scratch	CyberAgentLM2-7B, Sarashina2-7B, Sarashina2-13B, Fugaku-LLM 13B, LLM-jp-13B v2.0
LLMs continually pre-trained on Japanese text	Japanese Stable LM Base Gamma 7B Japanese Stable LM Beta 7B, RakutenAI-7B, Swallow 7B, Swallow-MS v0.1, Youri 7B, Llama 3 Swallow 8B, Llama 3 Youko 8B, ELYZA-japanese-Llama-2-13b, Swallow 13B, Swallow-MX 8x7B v0.1, Japanese Stable LM Beta 70B, KARAKURI LM 70B v0.1, Llama 3 Swallow 70B, Swallow 70B
Eglnish LLMs	Yi-1.5 6B, Llama 2 7B, Mistral-7B-v0.1, Mistral-7B-v0.2, Llama 3 8B, Yi-1.5 9B, Llama 2 13B, Yi-1.5 34B, Mixtral-8x7B-v0.1, Llama 2 70B, Llama 3 70B
Multilingual LLMs	C4AI Command-R v0.1, Qwen1.5-7B, Qwen2-7B, Qwen2-72B



	JCom.	JEMHopQA	NIILC	JSQuAD	XL-Sum	MGSM	WMT20-en-ja	WMT20-ja-en	JMMLU	JHumanEval	OpenBookQA	TriviaQA	HellaSwag	SQuAD2.0	XWINO	MMLU	GSM8K	BBH	HumanEval
PC1	0.25±0.01	0.13±0.01	0.01±0.01	0.24±0.01	0.21±0.01	0.26±0.01	0.07±0.01	0.24±0.0	0.27±0.0	0.23±0.01	0.25±0.0	0.26±0.0	0.27±0.0	0.22±0.01	0.24±0.01	0.28±0.0	0.26±0.0	0.25±0.0	0.23±0.01
PC2	0.06±0.01	0.42±0.01	0.57±0.0	0.13±0.01	-0.05±0.01	0.05±0.01	0.54±0.01	0.21±0.01	0.08±0.01	-0.19±0.01	-0.03±0.01	-0.03±0.01	-0.05±0.01	-0.16±0.01	-0.01±0.01	-0.12±0.01	-0.11±0.01	-0.04±0.01	-0.18±0.01
PC3	-0.1±0.02	0.24±0.02	0.03±0.02	-0.1±0.03	-0.34±0.03	0.31±0.01	0.03±0.02	-0.07±0.02	0.17±0.01	0.32±0.02	-0.21±0.02	-0.28±0.01	-0.2±0.01	0.2±0.03	-0.42±0.01	0.02±0.01	0.27±0.01	-0.05±0.02	0.33±0.02
PC4	-0.13±0.17	0.11±0.1	-0.03±0.01	-0.31±0.15	-0.51±0.07	-0.03±0.01	-0.05±0.02	0.18±0.04	-0.04±0.03	-0.22±0.05	0.32±0.08	0.09±0.07	0.21±0.05	0.44±0.07	0.05±0.04	0.01±0.03	0.04±0.02	0.05±0.04	-0.27±0.04

Figure 17: Leave-One-Out CV statistics: mean and standard deviations of the factor loadings ($n = 35$, blue: Japanese benchmarks, black: English benchmarks).