Is It Good Data for Multilingual Instruction Tuning or Just Bad Multilingual Evaluation for Large Language Models?

Pinzhen Chen[1] **Simon Yu**[1,2] **Zhicheng Guo**[3] **Barry Haddow**[1]
[1]University of Edinburgh [2]Northeastern University [3]Tsinghua University
pinzhen.chen@ed.ac.uk yu.chi@northeastern.edu
guo-zc21@mails.tsinghua.edu.cn bhaddow@ed.ac.uk

Abstract

Multilingual large language models are designed, claimed, and expected to cater to speakers of varied languages. We hypothesise that the current practices of fine-tuning and evaluating these models may not perfectly align with this objective owing to a heavy reliance on translation, which cannot cover language-specific knowledge but can introduce translation defects. It remains unknown whether the nature of the instruction data has an impact on the model output; conversely, it is questionable whether translated test sets can capture such nuances. Due to the often coupled practices of using translated data in both stages, such imperfections could have been overlooked. This work investigates these issues using controlled native or translated data during the instruction tuning and evaluation stages. We show that native or generation benchmarks reveal a notable difference between native and translated instruction data especially when model performance is high, whereas other types of test sets cannot. The comparison between round-trip and single-pass translations reflects the importance of knowledge from language-native resources. Finally, we demonstrate that regularization is beneficial to bridging this gap on structured but not generative tasks.[1]

1 Introduction

Instruction tuning, or supervised fine-tuning, can prepare a large language model (LLM) for better task generalization and natural interactions in downstream applications (Mishra et al., 2022; Wei et al., 2022; Sanh et al., 2022; Ouyang et al., 2022). Major efforts of building instruction datasets centre on English (Wei et al., 2022; Taori et al., 2023; Conover et al., 2023; Ivison et al., 2023), whereas the multilingual counterparts remain modest in size, variety, and coverage. Many multilingual instruction datasets have been seeded from English data and developed using translation as part of the pipeline (Li et al., 2023a; Chen et al., 2023b, 2024; Lai et al., 2023). A notable exception is Aya, a year-long project that invited volunteers around the globe to write and edit prompt-response examples in their native language (Singh et al., 2024), making it a language-native dataset.[2]

Although the Aya dataset was contributed by volunteers, it still carries a high social utility cost considering the personnel hours devoted. In contrast, translating existing resources by machine, or even by human, is a more convenient option. Nonetheless, translated data carry imperfections (Clark et al., 2020; Artetxe et al., 2020a): 1) it represents the culture and knowledge specific to the original language; 2) the translation process introduces translationese, an unnatural language style (Gellerstam, 1986; Baker, 1996), as well as errors since certain content or tasks can be corrupted, e.g. grammatical error correction. On the other hand, recent research discovered that instruction tuning is "superficial" where an LLM mainly learns the response format (Zhou et al., 2023), and it cannot enhance knowledge at the current scale (Ghosh et al., 2024). These insights imply that the shortcomings of translated data might not propagate into an instruction-tuned model. Hence, when widening language support, an important question arises: *Is translated data sufficient for instruction tuning?*

We regard "sufficiency" as the fact that, when used to fine-tune an LLM, translated instructions should lead to output quality similar to that of native data. Yet, examining this training data factor cannot be separated from carefully considering the evaluation protocol, because many existing multilingual benchmarks have been created via translation. This translation bias, if present in both training and test sets, could hinder a meaningful con-

[1]Our code and data will be made public at https://github.com/PinzhenChen/good-data-or-bad-eval.

[2]"Aya Dataset" but not "Aya Collection" which comprises many translated components.

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 9706–9726

clusion. We thus put forward our second research question: *If translated and native instruction data make a difference, would a translated benchmark capture it?* Subsequently, we use round-trip translated data to answer: *Which is the cause of the gap, translation defects or missing language-specific knowledge in instructions?* Finally, when translated data is hardly avoidable: *What techniques can we adopt to bridge the performance gap?*

This work systematically investigates native and translated data used during instruction tuning and evaluation. We experiment with eight models of varying sizes and data distributions and evaluate these models on nine benchmarks of different natures: translated versus native as well as classification versus generation. Empirical results suggest that a prudent choice in multilingual LLM evaluation is crucial. Foreshadowing the answers to the research questions raised earlier:

1. Native and translated data can lead to a performance gap on several benchmarks, especially when the model performance is strong.

2. Such a difference is more pronounced on benchmarks that are natively created (TyDi QA, CMMLU, C-Eval) or generative in nature (XQuAD, open-ended QA) compared to translated structured tests (MT/HT-MMLU).

3. Round-trip translation from native data outperforms single-pass translation from English data, implying that missing language-specific knowledge could be more detrimental than having translation defects.

4. Regularization during instruction tuning time, e.g. using a lower learning rate or multilingual instruction tuning, can be beneficial if translated data has to be used. It can close the native-translated performance gap on structured tasks but not generative tasks.

These insights mean that opposite conclusions can be made when different combinations of instruction and test sets are adopted. Based on the findings, we recommend multilingual LLMs be evaluated on a range of benchmarks to include language-native and generative tasks.

2 Related Work

2.1 Instruction tuning data

Instruction data can be created by writing questions and responses from scratch (Conover et al., 2023; Singh et al., 2024), collecting user-system interactions (Köpf et al., 2023), or templating structured data instances into natural texts (Mishra et al., 2022; Sanh et al., 2022). It is also feasible to distil large language models by feeding existing examples (Taori et al., 2023; Wei et al., 2023). Stemming from English data, many multilingual instruction datasets, especially open-ended question-response pairs, have been created via machine translation (Muennighoff et al., 2023; Chen et al., 2023a,b, 2024; Chai et al., 2024; Lai and Nissim, 2024). Slightly differently, Li et al. (2023a) translated English questions into multiple languages but used GPT to generate responses to avoid translationese. These options are more affordable than creating language-native data directly, but they are not flawless since they can introduce generation errors and knowledge-language mismatches.

In recent research progress on LLM instruction tuning, the "superficial alignment hypothesis" (Zhou et al., 2023) might offer some relief to these concerns. It claims that a strong foundation model mostly learns the response template from instruction tuning—therefore the translation artefacts or language-specific knowledge would not be overly consumed (Ghosh et al., 2024). To our knowledge, there is no prior work that systematically compared native and translated instruction data.

2.2 Multilingual LLM evaluation

Machine translation has been used to extend several benchmarks to more languages (Conneau et al., 2018; Artetxe et al., 2020b; Dumitrescu et al., 2021; Bandarkar et al., 2024, and the list is growing). Many studies exploring multilingualism in LLMs yielded findings based on translated instruction data and/or translated evaluation sets, from the earlier mT5 to the concurrent Llama 3.1 (Xue et al., 2020; Cañete et al., 2023; Ahuja et al., 2023; Cui et al., 2023; Puduppully et al., 2023; Yang et al., 2023; Lai et al., 2023; Kew et al., 2023; Chen et al., 2024; Singh et al., 2024; Ji and Chen, 2024; Liu et al., 2024; Shaham et al., 2024; Dubey et al., 2024). While these works have significantly pushed the boundary of multilingualism in LLMs, we attempt to revisit the effect of using translated data.

Clark et al. (2020) discussed the disadvantages of using translated tests: they incorporate translationese and represent the source language's knowledge; Artetxe et al. (2020a) revealed how minor translation artefacts can significantly impact evaluation outcomes. It has been shown and argued that,

albeit intuitively, translated training data improves scores on test data created via translation (Singh et al., 2019; Artetxe et al., 2020a). The machine translation community found that translated test input "can have a detrimental effect on the accuracy of evaluation" (Läubli et al., 2020; Graham et al., 2020; Farhad et al., 2021). This paper demonstrates that by altering the nature of the instruction or evaluation data, evaluation can lead to different conclusions for LLM instruction tuning.

Our comparative analysis of native and translated data also relates to understanding the integrity of LLM evaluation and the representation of language-specific knowledge from a meta-evaluation perspective. It is the expectation of the users that an LLM should not merely exhibit linguistic fluency but also embed the culture tied to the languages. We believe this to be a crucial and timely topic in the current LLM landscape. Earlier, Lyu et al. (2024) examined various mechanisms of obtaining LLM responses. Concurrently, Gema et al. (2024) found correctness issues in a specific benchmark; Etxaniz et al. (2024) showed that models can have distinct behaviours on local and global knowledge; Gu et al. (2024) called for transparency in choosing formatting and configurations. In comparison, our work looks at multilingual evaluation from the dimension of data characteristics.

3 Experiment Design

3.1 Instruction data

The focus of our study is to answer the research questions on the nature of instruction data and evaluation data as well as their impact on a trustworthy evaluation. We experiment with non-English training and test data created through distinct procedures: **created natively** and **translated**. We run monolingual instruction tuning: an LLM is fine-tuned in a single language every time to prevent potential cross-lingual influences.

Languages We study model performance in three languages—Spanish (es), Russian (ru), and Chinese (zh)—with the following considerations: 1) these languages cover a combination of different language families and writing scripts; 2) they are medium-to-high resourced, where the quality of the data, native or translated instructions, is satisfactory; 3) their presence in LLM pre-training data is significant, so we can expect reasonable output quality.

Native data We use the training split in the Aya dataset (Singh et al., 2024), which was written from scratch and then edited by human annotators in their native language. The Spanish, Russian, and Chinese training sets have a size of 3854, 423, and 3944 each.

Translated data We generate translated data equivalent in volume to the native data. This is done by sampling Aya's English split to match the size of native data in each language and translating the sample to that language. We always translate the instructions and responses separately. Two distinct versions of translated data are obtained via Google Translate and Cohere Command R[3]. Google Translate is a well-known commercial translation engine, whereas Command R, a large language model, is capable of adhering to more customised guidelines. Technically, we prompt Command R to maintain the original data formatting while translating, as illustrated Appendix A.1.

3.2 Close-ended evaluation

We perform automatic evaluations on close-ended tasks, where a model is expected to generate a predefined response given a question. The evaluation covers multilingual understanding and reasoning tasks commonly used to benchmark LLMs. These test sets come from various sources such as native annotation, human translation, and machine translation. All evaluations are conducted using `lm-evaluation-harness` (Gao et al., 2023) with default settings unless stated below.

Native benchmarks We first evaluate our instruction-tuned models on test sets that have been constructed from scratch by native speakers, on which we hypothesize a performance gap between native and translated instruction fine-tuning.

- **TyDi QA**: created by inviting native speakers to write down questions related to articles shown to them (Clark et al., 2020). We use its Russian split. We run 1-shot prompting and measure models' F1 scores.
- **CMMLU** (Li et al., 2023b) and **C-Eval** (Huang et al., 2024): both are multi-disciplinary tests containing questions on the Chinese culture and domain, made from resources in Chinese. We prompt with 5-shot examples and use accuracy as the metric.

[3] https://docs.cohere.com/docs/command-r

Unfortunately, we could not identify a native benchmark that assesses general knowledge in Spanish.

Translated benchmarks We use four translated benchmarks including both human-translated and machine-translated test sets. Most of these cover the three languages we study.

- **XQuAD**: a question answering task requiring text extraction from a given context (Artetxe et al., 2020b), human-translated from the English SQuAD (Rajpurkar et al., 2016). Evaluation is done in a 0-shot setting. We adopt two metrics: a strict string-level exact match (EM) and a lenient "include" checking whether the reference is a substring of the model generation.

- **MGSM**: grade school mathematics questions human-translated from the English GSM8K (Cobbe et al., 2021; Shi et al., 2023). We provide 5-shot examples with chain-of-thought and measure exact token match.

- **MT-MMLU**: Lai et al. (2023)'s ChatGPT-translated multilingual MMLU (Hendrycks et al., 2021), designated as MT-MMLU in our work. We use 5-shot prompting and accuracy as the metric.

- **HT-MMLU**: a professionally human-translated (HT) edition[4] of MMLU released when our camera-ready paper is being prepared. Section 4.4 offers a preliminary study of model behaviours on HT-MMLU and MT-MMLU to understand the impact of human and machine translation.

3.3 Open-ended generation

We then evaluate models with open-ended question answering (QA) under controlled translated and native settings:

- **Translated**: 50 English questions from OpenAssistant (OASST1; Köpf et al., 2023) and then human-translated by Chen et al. (2024). We use the translated questions in Spanish, Russian, and Chinese.

- **Native**: 50 questions in Spanish, Russian, and Chinese, directly sampled from OASST1. We only use the first-round queries in multi-turn conversations.

[4] https://huggingface.co/datasets/openai/MMMLU

Given the open-ended nature, there is no gold response to compare a model generation against. To avoid expensive human evaluation at scale, we use LLM-as-a-judge, which has shown a strong correlation with human judgement (Zheng et al., 2024). We use two LLM judges other than the translators to avoid LLM preference bias: GPT-4-Turbo and Command R+.[5] The judges directly score each instruction-response pair according to a 5-point Likert scale (1 to 5), which can avoid position bias in response comparison. The total score for a model therefore ranges between 50 to 250. The exact wording of the judging prompt is the same for both LLMs and is attached in Appendix A.2 Figure 7.

4 Experiments and Analysis

4.1 Technical setup

Base models We fine-tune base models of different sizes from three sources: 1) Llama 2 at 7B, trained on 2T tokens with a 32K vocabulary and released in Jul 2023 (Touvron et al., 2023); 2) Gemma at 2B and 7B (circa 8.54B parameters), trained on 3T and 6T tokens respectively with a 256K vocabulary and released in Feb 2024 (Gemma Team et al., 2024); 3) Qwen 1.5 at 0.5B, 1.8B, 4B, 7B, and 14B released in Feb 2024 (Qwen Team, 2024).[6]

Instruction tuning Let I represent an instruction and $Y = y_1, y_2, ..., y_{|Y|}$ a sequence of output tokens. The instruction is first templated into a pre-defined format, denoted as $\mathcal{T}(I)$. We fine-tune an LLM parameterised by θ by optimising the log-likelihood on the output tokens only: $\mathcal{L}(Y, \mathcal{T}(I); \theta) = -\log P(Y|\mathcal{T}(I); \theta)$.

We apply low-rank adaptation where the base model is loaded in 8-bit and frozen during training (Hu et al., 2022; Dettmers et al., 2023). We attach to all key, query, and value matrices a low-rank adapter with a rank of 8, an alpha of 16, and a dropout of 0.05. The learning rate is set to 10^{-4} and the effective batch size to 64. All models are given a training budget of 10 epochs and we validate perplexity on held-out instruction data after each epoch to keep the best checkpoint. We used a combination of NVIDIA 3090-24G, A100-40G, and A100-80G GPUs. Fine-tuning took 1 to 7 hours depending on the model and data size.

[5] Both accessed via API in Apr 2024.

[6] All models were up-to-date when the experiments were conducted in Apr 2024 but became one generation behind by the time the paper was accepted in Sep 2024.

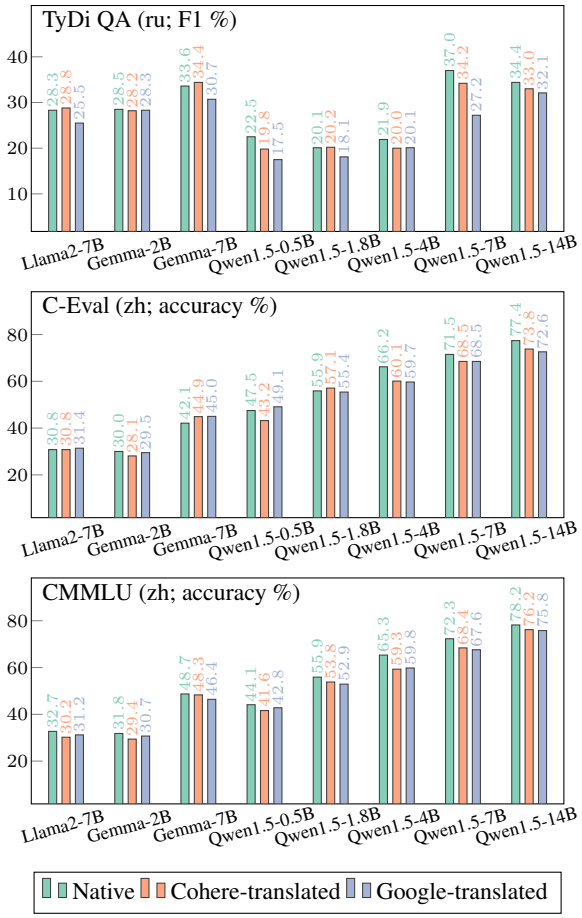

Figure 1: Results on native close-ended test sets: native instruction-tuned models have an edge.

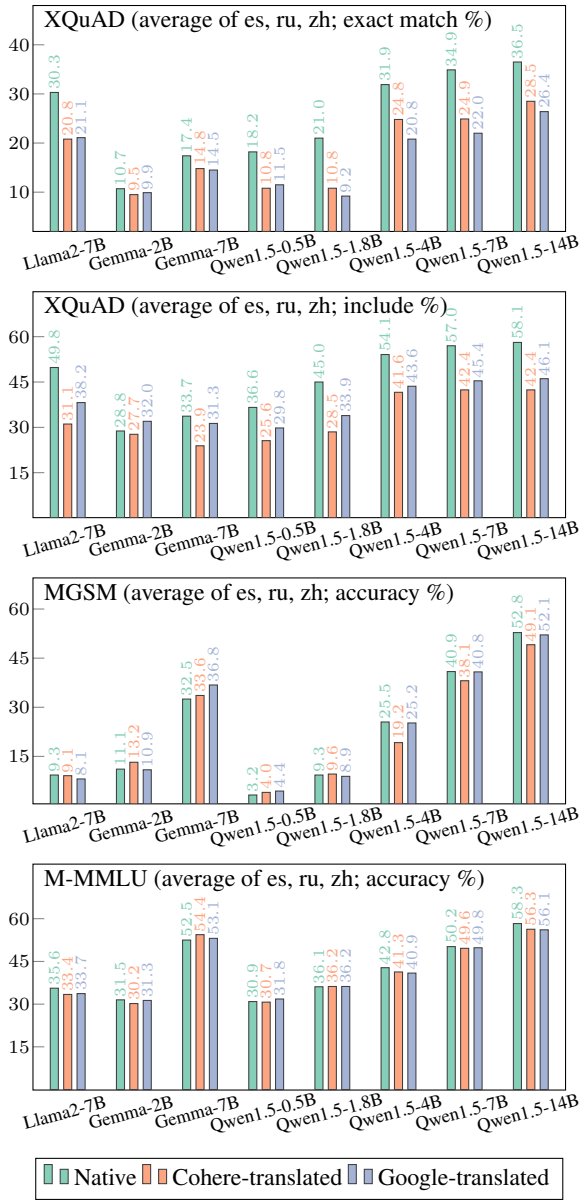

Figure 2: Results on translated close-ended test sets: native instruction-tuned models are superior on XQuAD, but all data conditions have comparable results on MGSM and MT-MMLU.

4.2 Is there a gap, and on what?

We display results for the native tests, TyDi QA, C-Eval, and CMMLU, in Figure 1. It shows that models fine-tuned with native instructions surpass those fine-tuned with translated data in most cases with consistent patterns across the two languages.

In terms of translated multilingual benchmarks, Figure 2 exhibits diverging trends. On the XQuAD benchmark, using native instruction data consistently and significantly outperforms translated data under both metrics, however, it loses the advantage on MGSM and MT-MMLU.

For open-ended QA, we show different combinations of the test data (native or human-translated) and judges (GPT-4-Turbo or Command R+) in Figure 3. The largest native-translated discrepancy occurs when models are tested on translated questions and judged by GPT-4-Turbo. When testing on translated questions and judged by Command R+, native data is slightly ahead when the model size is big. In other cases, native data is not better than translated data. These results also suggest that the LLM-as-a-judge metric affects empirical results too. However, it is difficult to arrive at a clear conclusion since we do not have transparent access to the data used in GPT or Command models—it might be the case that these models have been instruction-tuned with translated data.

Overall, we see that in terms of model performance, native data can surpass translated data under some evaluations, which suggests that translated instruction data is not always sufficient. While these observations have been made from the aspect of data/model performance, they cannot be decoupled from the potential test set imperfections. Assuming native data should lead to better metric numbers, it has been revealed that two types of eval-

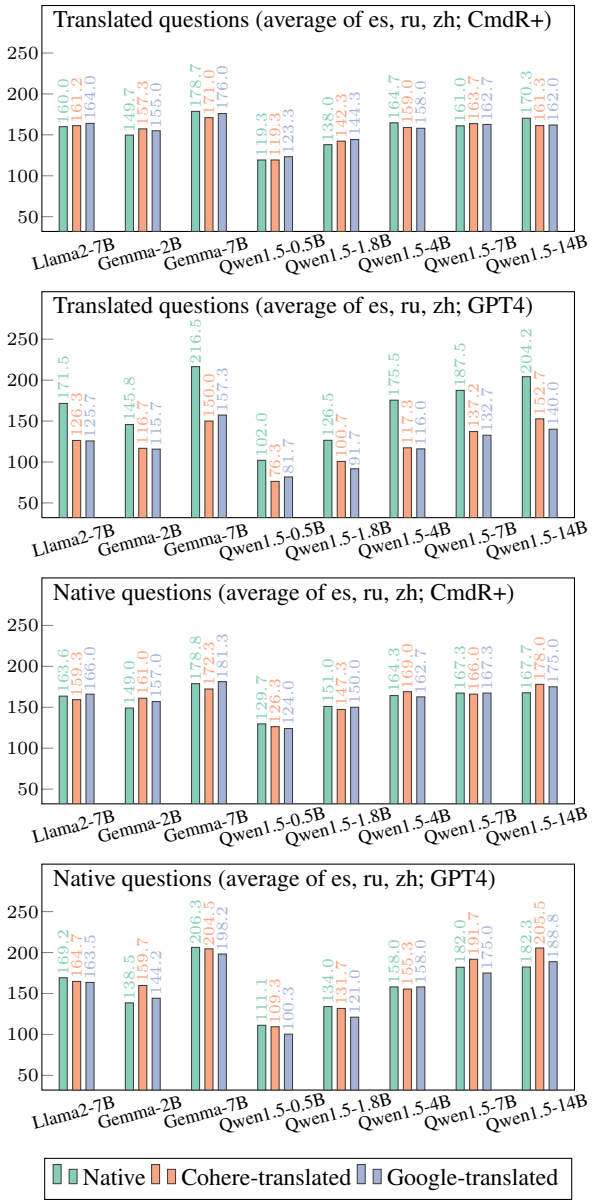

Figure 3: Results on native and translated open-ended question answering: native instruction-tuned models are superior for translated questions when judged by GPT-4-Turbo, but all data conditions result in similar numbers in other cases.

uation benchmarks are more effective in reflecting this: 1) those that originate in the test language itself (TyDi QA, C-Eval, and CMMLU) and 2) those that are generative in nature (XQuAD and open-ended questions) even though they could have been translated from English.

4.3 When is the gap obvious?

We hypothesise that the output quality difference between using native and translated data would be more noticeable when a model's overall performance is better—namely, the subtle translation bias might not be pivotal if a model's capability

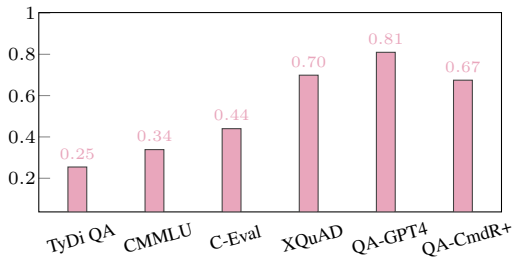

Figure 4: Pearson's correlation between *native data performance* and *native-translated performance difference* for various benchmarks: weaker correlation for structured tasks and stronger correlation for generative tasks.

is low enough that many instances are incorrectly predicted in the first place. Hence, for each previous benchmark where native data outperforms translated data, we run a post hoc analysis on the correlation between the native data performance and the native-translated performance gap.

We average the Cohere-translated and Google-translated scores to represent the final score for translated data. The score difference between models instruction-tuned on native data and translated data can then be defined as $\Delta S = S_{\text{native}} - \frac{1}{2}(S_{\text{cohere}} + S_{\text{google}})$, where S_{native}, S_{cohere}, and S_{google} stand for model scores on native, Cohere-translated, and Google-translated data respectively. Then, we compute the Pearson correlation coefficient $r_{\Delta S, S_{\text{native}}}$ between ΔS and S_{native} for each test set. It is worth noting that we consider all individual languages' scores instead of the averaged number across languages where applicable.

We cover all benchmarks where a clear native-translated difference has been observed earlier. Open-ended question answering is abbreviated as QA-GPT4 and QA-CmdR+ depending on the LLM judge used. The outcome is shown in Figure 4: the correlation between ΔS and S_{native} is weak for structured tasks like TyDi QA, C-MMLU, and C-Eval, but very strong for tasks involving generation like XQuAD and open-ended QA. This pattern indicates that 1) concerning the instruction data, the nature of being native or translated shines through as the model performance gets higher; 2) on the evaluation end, such data difference leaves a more pronounced gap on generative benchmarks. On a related note, in Kew et al. (2023)'s study, it is shown that cross-lingual transfer is more prominent in generative tasks but less in classification tasks. Altogether, it might be conjectured that the instruction data quality plays a more crucial role when a model is evaluated by generative tasks as opposed to classification tasks.

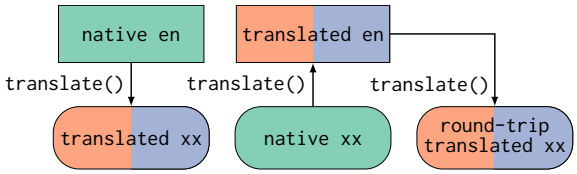

Figure 5: Round-trip translation (via English) produces translated data sharing the same origin as native data.

4.4 What potentially causes the gap?

Knowledge mismatch vs translation defects in instructions Translating instruction data introduces these imperfections. To understand which accounts more for model degradation, we disentangle the two elements in instructions using round-trip translation (RTT): we translate native data from one language into another and then translate it back to the original language, as illustrated in Figure 5. By doing so, we can have a "translated" dataset that preserves the same knowledge and domain as the original data but contains translation defects.

We construct the RTT version of Russian and Chinese instruction data from their native data with Cohere or Google translation pivoting via English. This follows the same procedure used to obtain translated instruction data in Section 3.1, except that the translation workflow is now done twice: X→English followed by English→X. We then compare models trained on RTT data with those trained on data translated from English on native benchmarks (TyDi QA and CMMLU).

Regarding TyDi QA in Table 1, we notice mixed results for Cohere translation but a relative advantage in RTT with Google translation. For CMMLU in Table 2, models with RTT (test language-origin) are uniformly better than those with data translated from English. RTT's strong performance—despite having undergone the translation process twice which likely leaves more translationese and errors—signifies the importance of incorporating native knowledge when widening language support in multilingual language models.

Human vs machine translated test sets Comparing MT-MMLU and HT-MMLU results can reveal the impact of human and machine translation on the evaluation end. This comparison is carefully controlled where both test sets have the same questions originating in English and testing the same knowledge. During testing, the same set of demonstrations is prepared for the same question across the two tests. We list Spanish and Chinese results in Table 3 which are very similar on the two

Base Model	Cohere		Google	
	RTT ru-origin	translated en-origin	RTT ru-origin	translated en-origin
Llama2-7B	25.9	**28.8**	**25.7**	25.5
Gemma-7B	29.4	**34.4**	**33.3**	30.7
Qwen1.5-4B	**23.0**	20.0	**22.4**	20.1
Qwen1.5-7B	**35.5**	34.2	**34.9**	27.2
Qwen1.5-14B	30.4	**33.0**	30.7	**32.1**

Table 1: Results for models trained on RTT data (ru-origin) or translated data (en-origin) on TyDi QA (ru).

Base Model	Cohere		Google	
	RTT zh-origin	translated en-origin	RTT zh-origin	translated en-origin
Llama2-7B	**31.6**	30.2	**32.2**	31.2
Gemma-7B	**48.6**	48.3	**48.4**	46.4
Qwen1.5-4B	**63.7**	59.3	**64.6**	59.8
Qwen1.5-7B	**68.9**	68.4	**70.5**	67.6
Qwen1.5-14B	**77.5**	76.2	**77.4**	75.8

Table 2: Results for models trained on RTT data (zh-origin) or translated data (en-origin) on CMMLU (zh).

Base Model	Data	Spanish		Chinese	
		MT-MMLU	HT-MMLU	MT-MMLU	HT-MMLU
Llama2-7B	native	38.4	37.6	34.4	33.8
	cohere	38.0	37.2	27.6	27.8
	google	36.4	35.9	30.4	29.5
Gemma-7B	native	55.9	54.9	48.7	48.0
	cohere	58.4	57.5	50.8	50.3
	google	56.1	55.6	49.7	48.8
Qwen1.5-4B	native	40.9	40.2	49.3	49.0
	cohere	39.9	39.0	44.5	45.2
	google	39.6	39.0	44.2	45.0
Qwen1.5-7B	native	50.3	49.6	52.9	53.0
	cohere	49.6	48.4	52.6	51.8
	google	50.4	49.3	51.8	51.3
Qwen1.5-14B	native	58.1	57.8	61.3	60.7
	cohere	55.8	55.1	57.9	57.7
	google	54.9	54.2	58.0	57.3

Table 3: Results for models trained on different data on MT-MMLU and HT-MMLU.

benchmarks and the native-translated gap is smaller compared with those on native or generative tasks. As shown in Appendix B Tables 19 and 21, the gaps even disappear under a lower learning rate.

Interestingly, gap patterns are consistent across the two translated MMLU tests: Llama2-7B on Chinese, Qwen1.4-4B on Chinese, and Qwen1.5-14B on both languages. These observations imply that both test sets are homogeneous and that (good) MT can match professional HT in expanding test

set language coverage. This also corroborates our early finding that missing language-specific knowledge can be a more differentiating factor.

4.5 Can we bridge the gap?

Despite Section 4.4 suggesting that it is more critical to have the domain and the knowledge of the native language in instructions, it is an unrealistic setting since it employs native data. This is difficult to obtain especially for under-served languages, so it is hard to avoid machine-translated data. We, therefore, investigate techniques that can apply better regularization during instruction tuning to reduce the negative impact of the translated data. This also represents an effort to pursue a more generalizable finding.

A lower learning rate Our first inspiration is drawn from Chirkova and Nikoulina (2024), whose experiments showed that English instruction-tuned models display remarkably different levels of cross-lingual transfer when only changing the learning rate—a smaller one leads to better instruction following in zero-shot languages. This means that it is possible to teach a base model a desired instruction-response style without even touching on the content or language. In this case, the undesirable properties in translated data could be mitigated. Following this, we run another set of experiments with the learning rate reduced from 10^{-4} to 10^{-6}.

Multilingualism Another exploration is multilingual instruction tuning, which could prevent a model from overfitting to a single language. In addition to Spanish, Russian, and Chinese which we evaluate, we also add another five languages—Arabic (ar), German (de), Finnish (fi), Irish (ga), and Hindi (hi)—into the multilingual pot. For the native multilingual data, we simply down-sample all languages in the Aya dataset to a size of 241 (the size of the German split in Aya, which is the smallest among the eight languages), leading to a total size of 1928. For the translated data in each language, we randomly select 241 instances from English and translate them (different data instances for different languages). This simulates a multilingual instruction set derived from translating English resources.

Setup For each of our previous data-model combinations, we now have two variants. Due to the space constraint, we only display results from larger models in the main text for the following benchmarks: TyDi QA, CMMLU, XQuAD, MSGM, MT-MMLU, and open-ended question answering. We **bold** the best native results and underline translated results if they are close to native—meaning that the gap can be closed. Moreover, exhaustive results for all models and all languages on all benchmarks are enclosed in Tables 10 to 25 in Appendix B.

Base Model	Data	$10^{-6}\leftarrow$ Mono	10^{-4} Mono	10^{-4} $\rightarrow$Multi
Llama2-7B	native	**33.4**	28.3	25.1
	cohere	33.3	28.8	23.4
	google	33.3	25.5	22.9
Gemma-7B	native	**37.7**	33.6	31.5
	cohere	38.1	34.4	31.4
	google	37.9	30.7	30.9
Qwen1.5-7B	native	22.4	**37.0**	**37.0**
	cohere	22.9	34.2	33.0
	google	22.7	27.2	27.1
Qwen1.5-14B	native	24.8	**34.4**	32.8
	cohere	24.6	33.0	29.3
	google	24.9	32.1	35.2

Table 4: Sometimes the gap can be closed on TyDi QA.

Base Model	Data	$10^{-6}\leftarrow$ Mono	10^{-4} Mono	10^{-4} $\rightarrow$Multi
Llama2-7B	native	31.8	**32.7**	**32.6**
	cohere	32.0	30.2	32.7
	google	32.0	31.2	32.1
Gemma-7B	native	49.9	48.7	**50.1**
	cohere	49.7	48.3	50.4
	google	49.8	46.4	50.7
Qwen1.5-7B	native	72.0	**72.3**	**72.6**
	cohere	71.9	68.4	71.4
	google	71.9	67.6	71.4
Qwen1.5-14B	native	77.7	**78.2**	**78.2**
	cohere	77.8	76.2	77.6
	google	77.8	75.8	77.2

Table 5: Sometimes the gap can be closed on CMMLU.

Native, structured benchmarks We make bold those scores that are higher than the rest for each model under all hyperparameter settings. We find that the pattern seems to be affected by the base model and the task. It can be seen that Llama2-7B and Gemma-7B enjoy a performance leap in two scenarios: 1) on TyDi QA with a lower learning rate; and 2) on CMMLU with multilingual instruction tuning. In both cases, the performance gap between native and translated data can be overcome. However, for Qwen1.5, while the results change as the training configuration changes, native data still is the best data condition to go with.

Base Model	Data	$10^{-6}\leftarrow$ Mono	10^{-4} Mono	10^{-4} →Multi
Llama2-7B	native	18.5	**30.3**	**31.0**
	cohere	18.0	20.8	21.6
	google	17.8	21.1	24.1
Gemma-7B	native	**17.8**	**17.4**	16.8
	cohere	17.3	14.8	16.3
	google	17.2	14.5	15.3
Qwen1.5-7B	native	30.7	**34.9**	**42.6**
	cohere	30.2	24.9	31.5
	google	29.9	22.0	27.7
Qwen1.5-14B	native	33.4	**36.5**	**45.6**
	cohere	33.6	28.5	30.8
	google	33.5	26.4	32.2

Table 6: There is always a large gap on XQuAD (EM).

Base Model	Data	$10^{-6}\leftarrow$ Mono	10^{-4} Mono	10^{-4} →Multi
Llama2-7B	native	9.5	9.3	10.8
	cohere	9.8	9.1	10.8
	google	9.7	8.1	12.0
Gemma-7B	native	38.9	32.5	37.1
	cohere	38.8	33.6	37.2
	google	39.5	36.8	36.4
Qwen1.5-7B	native	42.1	40.9	41.2
	cohere	41.5	38.1	40.8
	google	43.3	40.8	44.5
Qwen1.5-14B	native	55.9	52.8	55.2
	cohere	55.7	49.1	53.5
	google	55.7	52.1	56.4

Table 7: There is always no gap on MGSM

Translated, structured benchmarks Moving on to the translated test set results listed in Tables 6 to 8, we find that our previous findings still apply even when the learning rate is lowered or multilingual instruction tuning is applied. It can be seen that for the generative XQuAD, most of the time native instruction data maintains a huge advantage over the other two translated versions. Nonetheless, for MGSM and MT-MMLU, the difference between using translated and native data is not clear under most conditions. These also indicate that the stability of our results on translated structured tasks is not affected by the two hyperparameters.

Open-ended question answering with translated questions Finally, we compare monolingual and multilingual training on open-ended generation in Table 9. Despite some fluctuations, the native-translated gap cannot be mitigated when evaluated on open-ended generation with translated questions. This is consistent with patterns on XQuAD that generative benchmarks can more effectively differentiate the instruction tuning data source.

Base Model	Data	$10^{-6}\leftarrow$ Mono	10^{-4} Mono	10^{-4} →Multi
Llama2-7B	native	35.8	35.6	36.3
	cohere	35.8	33.4	34.1
	google	35.8	33.7	34.3
Gemma-7B	native	53.7	52.5	53.7
	cohere	53.8	54.4	55.6
	google	54.0	53.1	55.6
Qwen1.5-7B	native	50.3	50.2	50.6
	cohere	50.2	49.6	51.2
	google	50.1	49.8	50.9
Qwen1.5-14B	native	58.2	58.3	58.3
	cohere	58.3	56.3	58.5
	google	58.3	56.1	58.2

Table 8: There is always no gap on MT-MMLU.

Base Model	Data	Mono	Multi
Llama2-7B	native	**171.5**	121.7
	cohere	126.3	126.3
	google	125.7	131.0
Gemma-7B	native	**216.5**	164.7
	cohere	150.0	146.3
	google	157.3	147.3
Qwen1.5-7B	native	**187.5**	**189.3**
	cohere	137.2	138.0
	google	132.7	133.7
Qwen1.5-14B	native	**204.2**	**210.2**
	cohere	152.7	145.7
	google	140.0	140.7

Table 9: There is always a large gap on open-ended question answering (translated, GPT-4-Turbo judged).

5 Conclusion and Future Work

This work systematically analysed the effects of native and translated data on both the LLM instruction tuning and evaluation ends. The difference in data leads to result gaps on native test sets and generative benchmarks. We showed that knowledge mismatch is more likely to cause performance degradation rather than translation errors. With regularization, translated instruction data can potentially catch up with native data on structured benchmarks but not generative tasks.

Given our findings, we would like to call for prudent choices in multilingual LLM benchmarking. While the current work provides comprehensive empirical results and extrinsic evaluation, future work can consider investigating the knowledge in data intrinsically. More broadly, it is meaningful to coordinate efforts to develop large-scale native test sets that more accurately assess the breadth of languages and cultures LLMs aim to serve.

Limitations

This paper focused on providing empirical results as an extrinsic evaluation of data characteristics. It can benefit from having an intrinsic understanding of the distinction between native and translated data, e.g. the knowledge or language features missing in the translated data and how this is associated with errors in specific test questions.

Also, our work centred around instruction tuning, but we have very limited knowledge of the pre-training data for the LLMs we study. This work assumes that the base models are described accurately by respective makers and that the LLM pre-training data would not prevent us from making meaningful scientific conclusions.

Ethical Considerations

We consider our work to have minimal ethical risks. Like most papers on LLMs, it is difficult to make sure that the fine-tuned model is safe in all cases, but our models are not intended for the public. In terms of LLM evaluation, we believe this paper makes a positive contribution towards trustworthy and tailored evaluation for languages covered in large language models.

Acknowledgments

The work has received funding from UK Research and Innovation (UKRI) under the UK government's Horizon Europe funding guarantee [grant number 10052546].

Computations described in this research were supported by the Baskerville Tier 2 HPC service (https://www.baskerville.ac.uk/). Baskerville was funded by the EPSRC and UKRI through the World Class Labs scheme (EP/T022221/1) and the Digital Research Infrastructure programme (EP/W032244/1) and is operated by Advanced Research Computing at the University of Birmingham. We also acknowledge the Edinburgh International Data Facility (EIDF) and the Data-Driven Innovation Programme at the University of Edinburgh. The Cohere API credits were supported by a Cohere For AI Research Grant.

References

Kabir Ahuja, Harshita Diddee, Rishav Hada, Millicent Ochieng, Krithika Ramesh, Prachi Jain, Akshay Nambi, Tanuja Ganu, Sameer Segal, Mohamed Ahmed, Kalika Bali, and Sunayana Sitaram. 2023. MEGA: Multilingual evaluation of generative AI. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2020a. Translation artifacts in cross-lingual transfer learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*.

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020b. On the cross-lingual transferability of monolingual representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Mona Baker. 1996. *Corpus-based Translation Studies: The Challenges that Lie Ahead*. Benjamins Translation Library. John Benjamins Publishing Company.

Lucas Bandarkar, Davis Liang, Benjamin Muller, Mikel Artetxe, Satya Narayan Shukla, Donald Husa, Naman Goyal, Abhinandan Krishnan, Luke Zettlemoyer, and Madian Khabsa. 2024. The belebele benchmark: a parallel reading comprehension dataset in 122 language variants. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*.

José Luis González Cañete, Gabriel Chaperon, Rodrigo Fuentes, Jou-Hui Ho, Hojin Kang, and Jorge P'erez. 2023. Spanish pre-trained BERT model and evaluation data. *arXiv preprint*.

Linzheng Chai, Jian Yang, Tao Sun, Hongcheng Guo, Jiaheng Liu, Bing Wang, Xiannian Liang, Jiaqi Bai, Tongliang Li, Qiyao Peng, et al. 2024. xCoT: Cross-lingual instruction tuning for cross-lingual chain-of-thought reasoning. *arXiv preprint*.

Nuo Chen, Zinan Zheng, Ning Wu, Linjun Shou, Ming Gong, Yangqiu Song, Dongmei Zhang, and Jia Li. 2023a. Breaking language barriers in multilingual mathematical reasoning: Insights and observations. *arXiv preprint*.

Pinzhen Chen, Shaoxiong Ji, Nikolay Bogoychev, Andrey Kutuzov, Barry Haddow, and Kenneth Heafield. 2024. Monolingual or multilingual instruction tuning: Which makes a better Alpaca. In *Findings of the Association for Computational Linguistics: EACL 2024*.

Zhihong Chen, Shuo Yan, Juhao Liang, Feng Jiang, Xiangbo Wu, Fei Yu, Guiming Hardy Chen, Junying Chen, Hongbo Zhang, Li Jianquan, Wan Xiang, and Benyou Wang. 2023b. MultilingualSIFT: Multilingual supervised instruction fine-tuning. GitHub.

Nadezhda Chirkova and Vassilina Nikoulina. 2024. Zero-shot cross-lingual transfer in instruction tuning of large language models. In *Proceedings of the 17th International Natural Language Generation Conference*.

Jonathan H. Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. 2020. TyDi QA: A benchmark for information-seeking question answering in typologically diverse languages. *Transactions of the Association for Computational Linguistics*.

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

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. 2023. Free Dolly: Introducing the world's first truly open instruction-tuned LLM. Online Blog.

Yiming Cui, Ziqing Yang, and Xin Yao. 2023. Efficient and effective text encoding for Chinese LLaMA and Alpaca. *arXiv preprint*.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. QLoRA: Efficient finetuning of quantized LLMs. In *Thirty-seventh Conference on Neural Information Processing Systems*.

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

Stefan Daniel Dumitrescu, Petru Rebeja, Beata Lorincz, Mihaela Gaman, Andrei Avram, Mihai Ilie, Andrei Pruteanu, Adriana Stan, Lorena Rosia, Cristina Iacobescu, et al. 2021. LiRo: Benchmark and leaderboard for Romanian language tasks. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Julen Etxaniz, Gorka Azkune, Aitor Soroa, Oier Lopez de Lacalle, and Mikel Artetxe. 2024. BertaQA: How much do language models know about local culture? *arXiv preprint*.

Akhbardeh Farhad, Arkhangorodsky Arkady, Biesialska Magdalena, Bojar Ondřej, Chatterjee Rajen, Chaudhary Vishrav, Marta R Costa-jussa, España-Bonet Cristina, Fan Angela, Federmann Christian, et al. 2021. Findings of the 2021 conference on machine translation (WMT21). In *Proceedings of the Sixth Conference on Machine Translation*.

Leo Gao, Jonathan Tow, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Kyle McDonell, Niklas Muennighoff, et al. 2023. A framework for few-shot language model evaluation. Zenodo.

Martin Gellerstam. 1986. Translationese in Swedish novels translated from English. In *Translation studies in Scandinavia: Proceedings from the Scandinavian Symposium on Translation Theory II*. CWK Gleerup.

Aryo Pradipta Gema, Joshua Ong Jun Leang, Giwon Hong, Alessio Devoto, Alberto Carlo Maria Mancino, Rohit Saxena, Xuanli He, Yu Zhao, Xiaotang Du, Mohammad Reza Ghasemi Madani, et al. 2024. Are we done with MMLU? *arXiv preprint*.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. 2024. Gemma: Open models based on Gemini research and technology. *arXiv preprint*.

Sreyan Ghosh, Chandra Kiran Reddy Evuru, Sonal Kumar, Deepali Aneja, Zeyu Jin, Ramani Duraiswami, Dinesh Manocha, et al. 2024. A closer look at the limitations of instruction tuning. *arXiv preprint*.

Yvette Graham, Barry Haddow, and Philipp Koehn. 2020. Statistical power and translationese in machine translation evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*.

Yuling Gu, Oyvind Tafjord, Bailey Kuehl, Dany Haddad, Jesse Dodge, and Hannaneh Hajishirzi. 2024. Olmes: A standard for language model evaluations. *arXiv preprint*.

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*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Yuzhen Huang, Yuzhuo Bai, Zhihao Zhu, Junlei Zhang, Jinghan Zhang, Tangjun Su, Junteng Liu, Chuancheng Lv, Yikai Zhang, Yao Fu, et al. 2024. C-Eval: A multi-level multi-discipline Chinese evaluation suite for foundation models. *Advances in Neural Information Processing Systems*.

Hamish Ivison, Yizhong Wang, Valentina Pyatkin, Nathan Lambert, Matthew Peters, Pradeep Dasigi, Joel Jang, David Wadden, Noah A Smith, Iz Beltagy, et al. 2023. Camels in a changing climate: Enhancing LM adaptation with Tulu 2. *arXiv preprint*.

Shaoxiong Ji and Pinzhen Chen. 2024. Lucky 52: How many languages are needed to instruction fine-tune large language models? *arXiv preprint*.

Tannon Kew, Florian Schottmann, and Rico Sennrich. 2023. Turning english-centric LLMs into polyglots: How much multilinguality is needed? *arXiv preprint*.

Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi-Rui Tam, Keith Stevens, Abdullah Barhoum, Nguyen Minh Duc, Oliver Stanley, Richárd Nagyfi, et al. 2023. OpenAssistant conversations - democratizing large language model alignment. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Huiyuan Lai and Malvina Nissim. 2024. mCoT: Multilingual instruction tuning for reasoning consistency in language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*.

Viet Lai, Chien Nguyen, Nghia Ngo, Thuat Nguyen, Franck Dernoncourt, Ryan Rossi, and Thien Nguyen. 2023. Okapi: Instruction-tuned large language models in multiple languages with reinforcement learning from human feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*.

Samuel Läubli, Sheila Castilho, Graham Neubig, Rico Sennrich, Qinlan Shen, and Antonio Toral. 2020. A set of recommendations for assessing human-machine parity in language translation. *Journal of Artificial Intelligence Research*.

Haonan Li, Fajri Koto, Minghao Wu, Alham Fikri Aji, and Timothy Baldwin. 2023a. Bactrian-X: A multilingual replicable instruction-following model with low-rank adaptation. *arXiv preprint*.

Haonan Li, Yixuan Zhang, Fajri Koto, Yifei Yang, Hai Zhao, Yeyun Gong, Nan Duan, and Tim Baldwin. 2023b. CMMLU: Measuring massive multitask language understanding in Chinese. *arXiv preprint*.

Chaoqun Liu, Wenxuan Zhang, Yiran Zhao, Anh Tuan Luu, and Lidong Bing. 2024. Is translation all you need? a study on solving multilingual tasks with large language models. *arXiv preprint*.

Chenyang Lyu, Minghao Wu, and Alham Fikri Aji. 2024. Beyond probabilities: Unveiling the misalignment in evaluating large language models. *arXiv preprint*.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2022. Cross-task generalization via natural language crowdsourcing instructions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng-Xin Yong, Hailey Schoelkopf, et al. 2023. Crosslingual generalization through multitask finetuning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*.

Ratish Puduppully, Anoop Kunchukuttan, Raj Dabre, Ai Ti Aw, and Nancy Chen. 2023. DecoMT: Decomposed prompting for machine translation between related languages using large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*.

Qwen Team. 2024. Introducing Qwen1.5. Online Blog.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. 2022. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*.

Uri Shaham, Jonathan Herzig, Roee Aharoni, Idan Szpektor, Reut Tsarfaty, and Matan Eyal. 2024. Multilingual instruction tuning with just a pinch of multilinguality. *arXiv preprint*.

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

Jasdeep Singh, Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. 2019. XLDA: Cross-lingual data augmentation for natural language inference and question answering. *arXiv preprint*.

Shivalika Singh, Freddie Vargus, Daniel Dsouza, Börje F Karlsson, Abinaya Mahendiran, Wei-Yin Ko, Herumb Shandilya, Jay Patel, Deividas Mataciunas, Laura OMahony, et al. 2024. Aya dataset: An open-access collection for multilingual instruction tuning. *arXiv preprint*.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford Alpaca: An instruction-following LLaMA model. GitHub.

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

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. 2022. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

Xiangpeng Wei, Haoran Wei, Huan Lin, Tianhao Li, Pei Zhang, Xingzhang Ren, Mei Li, Yu Wan, Zhiwei Cao, Binbin Xie, et al. 2023. PolyLM: An open source polyglot large language model. *arXiv preprint*.

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. In *North American Chapter of the Association for Computational Linguistics*.

Wen Yang, Chong Li, Jiajun Zhang, and Chengqing Zong. 2023. BigTranslate: Augmenting large language models with multilingual translation capability over 100 languages. *arXiv preprint*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2024. Judging LLM-as-a-judge with MT-bench and Chatbot Arena. *Advances in Neural Information Processing Systems*.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, LILI YU, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. 2023. LIMA: Less is more for alignment. In *Thirty-seventh Conference on Neural Information Processing Systems*.

A Prompts

A.1 Command R translation prompt

We list the translation prompt we use to query Command R in Figure 6, which asks the LLM to translate a given text while preserving the formatting. The source language, target language, and text variables are replaced by their string values during prompting.

```
Please translate from ${source_lang} to ${target_lang}. Do your best to preserve the formatting. The following content should and should only be translated.

${text}
```

Figure 6: Prompt template for requesting a translation from Command R.

A.2 LLM-as-a-judge prompt

We list the LLM-as-a-judge prompt we use to query GPT-4-Turbo and Command-R+ in Figure 7, which requires the judge to give a brief explanation before scoring. The instruction and response variables are replaced by their string values during prompting.

```
Please act as an impartial judge and evaluate the quality of a response to a user instruction displayed below. Your evaluation should consider factors such as helpfulness, relevance, accuracy, depth, creativity, and level of detail. Begin your evaluation with a brief explanation. After that, please rate the response on a scale of 1 to 5 by strictly following this format: "[[rating]]". The rating must be enclosed by two square brackets, for example: "Rating: [[2]]".

[User Instruction]
${instruction}

[Response]
${response}
```

Figure 7: Prompt template for requesting a model response evaluation from GPT-4-Turbo or Command-R+.

B Comprehensive Results

We list a breakdown of the results for each model and each language on various benchmarks in this appendix section. These are:

- Table 10: TyDi QA Russian, F1
- Table 11: CMMLU, accuracy
- Table 12: XQuAD, 10^{-4}, exact match
- Table 13: XQuAD, 10^{-6}, exact match
- Table 14: XQuAD, 10^{-4}, "include"
- Table 15: XQuAD, 10^{-6}, "include"
- Table 16: MGSM, 10^{-4}, exact token match
- Table 17: MGSM, 10^{-6}, exact token match
- Table 18: MT-MMLU, 10^{-4}, accuracy
- Table 19: MT-MMLU, 10^{-6}, accuracy
- Table 20: HT-MMLU, 10^{-4}, accuracy
- Table 21: HT-MMLU, 10^{-6}, accuracy
- Table 22: translated questions, GPT-4 judge
- Table 23: translated questions, Cmd R+ judge
- Table 24: native questions, GPT-4 judge
- Table 25: native questions, Cmd R+ judge

Base Model	Data	$1e{-}4$			$1e{-}6$	
		Mono	Multi	RTT	Mono	Multi
Llama2-7B	native	28.3	25.1	–	33.4	33.4
	cohere	28.8	23.4	25.9	33.3	33.4
	google	25.5	22.9	25.7	33.3	33.4
Gemma-2B	native	28.5	24.9	–	27.7	27.6
	cohere	28.2	28.8	27.7	27.7	27.8
	google	28.3	28.8	28.1	28.0	27.5
Gemma-7B	native	33.6	31.5	–	37.7	38.1
	cohere	34.4	31.4	29.4	38.1	39.0
	google	30.7	30.9	33.3	37.9	38.7
Qwen1.5-0.5B	native	22.5	23.9	–	16.6	17.1
	cohere	19.8	20.0	23.9	16.9	17.7
	google	17.5	19.8	23.0	16.8	17.1
Qwen1.5-1.8B	native	20.1	27.6	–	19.5	19.7
	cohere	20.2	18.1	26.4	19.7	19.6
	google	18.1	19.3	23.6	19.8	19.8
Qwen1.5-4B	native	21.9	28.3	–	18.1	18.2
	cohere	20.0	22.5	23.0	17.9	18.2
	google	20.1	22.5	22.4	17.9	18.2
Qwen1.5-7B	native	37.0	37.0	–	22.4	23.8
	cohere	34.2	33.0	35.5	22.9	23.9
	google	27.2	27.1	34.9	22.7	23.9
Qwen1.5-14B	native	34.4	32.8	–	24.8	25.1
	cohere	33.0	29.3	30.4	24.6	25.1
	google	32.1	35.2	30.7	24.9	25.9

Table 10: Results for each model on TyDiQA Russian (F1, %).

Base Model	Data	$1e{-}4$			$1e{-}6$	
		Mono	Multi	RTT	Mono	Multi
Llama2-7B	native	32.7	32.6	-	31.8	31.9
	cohere	30.2	32.7	31.6	32.0	31.6
	google	31.2	32.1	32.2	32.0	31.8
Gemma-2B	native	31.8	31.5	-	31.2	31.0
	cohere	29.4	31.2	30.4	31.2	31.0
	google	30.7	31.8	30.2	31.3	31.2
Gemma-7B	native	48.7	50.1	-	49.9	49.7
	cohere	48.3	50.4	48.6	49.7	49.9
	google	46.4	50.7	48.4	49.8	50.1
Qwen1.5-0.5B	native	44.1	43.9	-	42.2	42.1
	cohere	41.6	42.3	41.0	42.1	42.2
	google	42.8	42.7	43.0	42.0	42.2
Qwen1.5-1.8B	native	55.9	56.6	-	56.7	56.7
	cohere	53.8	56.6	54.7	56.6	56.4
	google	52.9	57.0	55.3	56.9	56.6
Qwen1.5-4B	native	65.3	66.4	-	66.0	65.8
	cohere	59.3	65.2	63.7	65.7	65.7
	google	59.8	65.2	64.6	66.0	65.8
Qwen1.5-7B	native	72.3	72.6	-	72.0	71.9
	cohere	68.4	71.4	68.9	71.9	71.8
	google	67.6	71.4	70.5	71.9	72.0
Qwen1.5-14B	native	78.2	78.2	-	77.7	77.6
	cohere	76.2	77.6	77.5	77.8	77.7
	google	75.8	77.2	77.4	77.8	77.7

Table 11: Results for each model on CMMLU (accuracy, %).

Base model	Data	Monolingual				Multilingual			
		es	ru	zh	*average*	es	ru	zh	*average*
Llama2-7B	native	30.7	18.7	41.3	30.3	28.0	21.8	43.0	31.0
	cohere	17.2	15.3	29.8	20.8	18.4	7.6	38.8	21.6
	google	20.8	14.3	28.2	21.1	22.9	10.3	39.2	24.1
Gemma-2B	native	11.3	5.9	15.0	10.7	11.1	6.0	14.1	10.4
	cohere	10.8	6.0	11.6	9.5	11.2	5.0	9.2	8.5
	google	10.4	5.8	13.4	9.9	9.9	4.8	5.4	6.7
Gemma-7B	native	12.4	10.3	29.4	17.4	13.0	10.8	26.7	16.8
	cohere	12.7	10.1	21.5	14.8	12.9	9.0	27.1	16.3
	google	12.4	8.8	22.3	14.5	13.2	8.5	24.1	15.3
Qwen1.5-0.5B	native	26.5	7.6	20.6	18.2	18.1	10.2	21.7	16.6
	cohere	9.8	6.6	16.1	10.8	11.3	7.4	15.5	11.4
	google	10.8	6.5	17.2	11.5	12.7	7.7	16.1	12.2
Qwen1.5-1.8B	native	28.7	10.9	23.3	21.0	28.7	20.7	31.9	27.1
	cohere	11.7	5.9	14.7	10.8	14.3	5.5	21.9	13.9
	google	10.3	5.8	11.6	9.2	15.9	6.3	25.5	15.9
Qwen1.5-4B	native	33.1	24.9	37.7	31.9	36.5	31.4	52.6	40.2
	cohere	29.6	19.3	25.4	24.8	31.9	18.4	39.4	29.9
	google	19.7	18.7	23.9	20.8	35.1	20.3	40.2	31.8
Qwen1.5-7B	native	43.9	30.4	30.3	34.9	39.9	35.4	52.5	42.6
	cohere	27.6	25.5	21.6	24.9	36.3	22.9	35.4	31.5
	google	23.6	21.0	21.5	22.0	32.1	19.9	31.0	27.7
Qwen1.5-14B	native	44.7	34.2	30.7	36.5	49.7	38.7	48.3	45.6
	cohere	35.3	28.4	21.9	28.5	39.9	24.0	28.6	30.8
	google	28.3	26.8	24.2	26.4	42.1	26.6	27.7	32.2

Table 12: All model and all language results on XQuAD (10^{-4}, exact match, %).

Base model	Data	Monolingual				Multilingual			
		es	ru	zh	*average*	es	ru	zh	*average*
Llama2-7B	native	12.9	7.3	35.3	18.5	12.4	7.1	31.9	17.1
	cohere	12.9	7.1	33.9	18.0	12.4	7.3	32.3	17.3
	google	12.9	7.1	33.3	17.8	12.4	7.2	32.2	17.3
Gemma-2B	native	10.0	6.1	4.8	7.0	10.0	6.2	4.9	7.0
	cohere	10.0	5.9	4.6	6.8	9.9	6.1	4.8	6.9
	google	10.0	6.1	4.7	6.9	10.0	6.0	4.9	6.9
Gemma-7B	native	12.5	9.1	31.7	17.8	12.5	9.1	31.2	17.6
	cohere	12.4	9.1	30.6	17.3	12.4	9.3	30.3	17.3
	google	12.3	9.1	30.3	17.2	12.9	9.1	30.3	17.4
Qwen1.5-0.5B	native	12.1	6.2	12.8	10.4	10.5	6.5	11.2	9.4
	cohere	10.8	6.2	11.8	9.6	10.2	6.5	10.6	9.1
	google	11.1	6.2	12.0	9.8	9.9	6.2	11.0	9.0
Qwen1.5-1.8B	native	15.6	5.1	20.4	13.7	14.1	5.1	16.0	11.7
	cohere	15.0	5.2	18.7	13.0	14.1	5.0	16.3	11.8
	google	14.9	5.0	18.2	12.7	14.0	5.0	16.5	11.8
Qwen1.5-4B	native	24.4	14.3	44.9	27.8	21.0	14.9	37.8	24.6
	cohere	23.3	14.2	43.4	26.9	20.9	14.7	37.6	24.4
	google	22.7	14.3	43.5	26.8	20.8	15.0	37.6	24.5
Qwen1.5-7B	native	31.3	19.0	41.8	30.7	23.4	19.9	33.2	25.5
	cohere	32.8	18.8	39.0	30.2	23.2	19.4	33.3	25.3
	google	31.9	19.1	38.7	29.9	22.8	19.6	32.9	25.1
Qwen1.5-14B	native	37.7	26.9	35.7	33.4	37.2	27.6	28.1	31.0
	cohere	38.3	27.1	35.3	33.6	36.7	27.5	27.3	30.5
	google	37.6	27.0	35.9	33.5	36.8	27.3	27.4	30.5

Table 13: Results for each model and each language on XQuAD (10^{-6}, exact match, %).

Base model	Data	Monolingual				Multilingual			
		es	ru	zh	*average*	es	ru	zh	*average*
Llama2-7B	native	51.7	31.8	66.0	49.8	55.6	30.8	61.9	49.4
	cohere	23.5	26.3	43.4	31.1	30.3	16.9	55.3	34.2
	google	43.7	27.1	43.8	38.2	46.1	28.4	58.7	44.4
Gemma-2B	native	17.2	18.8	50.3	28.8	19.6	13.9	44.7	26.1
	cohere	26.3	20.3	36.6	27.7	36.1	17.8	47.1	33.7
	google	33.7	22.6	39.7	32.0	39.2	22.8	52.8	38.3
Gemma-7B	native	14.7	24.7	61.8	33.7	15.7	17.5	67.2	33.5
	cohere	13.9	20.3	37.5	23.9	13.4	12.3	44.6	23.4
	google	16.5	29.0	48.5	31.3	23.7	19.3	48.2	30.4
Qwen1.5-0.5B	native	35.7	19.7	54.5	36.6	29.7	20.4	46.1	32.1
	cohere	19.7	23.8	33.3	25.6	25.5	16.7	45.2	29.1
	google	26.9	24.5	37.9	29.8	28.8	16.9	46.2	30.6
Qwen1.5-1.8B	native	44.9	25.8	64.4	45.0	45.0	30.6	56.4	44.0
	cohere	24.0	20.8	40.6	28.5	34.2	20.0	56.2	36.8
	google	35.5	19.5	46.8	33.9	35.4	23.2	59.1	39.2
Qwen1.5-4B	native	57.8	35.5	69.0	54.1	57.2	41.8	72.3	57.1
	cohere	40.7	34.4	49.7	41.6	51.6	28.1	63.9	47.8
	google	38.9	33.4	58.5	43.6	51.9	31.5	67.5	50.3
Qwen1.5-7B	native	58.6	44.0	68.4	57.0	65.0	45.5	72.6	61.0
	cohere	48.9	33.5	44.6	42.4	43.5	31.7	61.6	45.6
	google	44.2	37.4	54.7	45.4	44.9	33.9	62.5	47.1
Qwen1.5-14B	native	61.1	43.5	69.6	58.1	65.0	51.0	69.7	61.9
	cohere	48.3	35.6	43.3	42.4	43.1	30.1	56.7	43.3
	google	48.9	34.0	55.2	46.1	48.5	30.8	57.8	45.7

Table 14: Results for each model and each language on XQuAD (10^{-4}, "include", %).

Base model	Data	Monolingual				Multilingual			
		es	ru	zh	*average*	es	ru	zh	*average*
Llama2-7B	native	15.3	11.7	53.0	26.7	14.2	11.8	52.6	26.2
	cohere	15.0	11.6	53.4	26.7	14.0	11.8	53.2	26.4
	google	15.0	11.6	53.1	26.6	14.3	11.8	52.7	26.2
Gemma-2B	native	37.6	19.2	45.5	34.1	36.5	19.3	45.2	33.7
	cohere	37.8	19.7	45.4	34.3	36.8	19.2	45.0	33.7
	google	38.4	19.6	45.0	34.3	36.5	19.2	45.5	33.7
Gemma-7B	native	14.2	11.8	50.7	25.6	15.9	11.6	52.4	26.6
	cohere	13.5	11.0	45.4	23.3	16.0	10.3	51.3	25.9
	google	12.4	11.3	45.7	23.1	16.6	10.4	50.9	26.0
Qwen1.5-0.5B	native	41.1	31.1	53.9	42.0	43.9	30.8	55.4	43.4
	cohere	43.2	32.2	54.9	43.4	44.5	31.9	55.7	44.1
	google	42.8	31.8	55.5	43.3	44.1	31.6	55.7	43.8
Qwen1.5-1.8B	native	39.4	24.4	59.5	41.1	42.4	24.4	60.4	42.4
	cohere	41.1	24.8	59.8	41.9	42.6	24.3	61.0	42.6
	google	40.6	24.4	59.7	41.6	42.2	24.5	60.8	42.5
Qwen1.5-4B	native	59.4	37.7	70.5	55.9	62.1	37.3	71.0	56.8
	cohere	59.7	37.2	70.7	55.9	62.4	37.4	71.3	57.0
	google	60.3	36.6	70.6	55.8	62.4	37.8	70.8	57.0
Qwen1.5-7B	native	63.4	44.9	70.6	59.6	64.9	43.8	72.9	60.5
	cohere	62.4	45.2	70.3	59.3	65.8	44.0	73.6	61.1
	google	63.4	45.1	70.3	59.6	65.2	44.5	73.0	60.9
Qwen1.5-14B	native	57.1	43.4	70.8	57.1	59.7	43.3	72.6	58.5
	cohere	56.1	43.8	70.3	56.7	59.8	43.4	72.9	58.7
	google	56.6	43.4	70.2	56.8	59.0	43.1	72.4	58.2

Table 15: Results for each model and each language on XQuAD (10^{-6}, "include", %).

Base model	Data	Monolingual				Multilingual			
		es	ru	zh	*average*	es	ru	zh	*average*
Llama2-7B	native	8.4	10.0	9.6	9.3	12.0	10.4	10.0	10.8
	cohere	5.2	11.2	10.8	9.1	10.0	12.4	10.0	10.8
	google	8.4	9.2	6.8	8.1	12.0	10.8	13.2	12.0
Gemma-2B	native	11.6	12.8	8.8	11.1	13.2	10.8	9.2	11.1
	cohere	11.6	12.4	15.6	13.2	11.6	10.8	10.8	11.1
	google	12.8	11.2	8.8	10.9	10.0	9.2	11.2	10.1
Gemma-7B	native	30.0	48.8	18.8	32.5	36.4	47.2	27.6	37.1
	cohere	34.4	46.8	19.6	33.6	36.4	44.0	31.2	37.2
	google	30.0	48.8	31.6	36.8	37.6	42.8	28.8	36.4
Qwen1.5-0.5B	native	2.8	2.0	4.8	3.2	2.0	1.6	10.4	4.7
	cohere	1.6	2.4	8.0	4.0	3.6	2.4	8.8	4.9
	google	2.0	2.0	9.2	4.4	2.4	3.6	7.6	4.5
Qwen1.5-1.8B	native	6.0	6.4	15.6	9.3	9.6	6.0	19.6	11.7
	cohere	8.4	5.6	14.8	9.6	6.8	7.6	15.6	10.0
	google	6.4	5.6	14.8	8.9	6.0	5.2	21.2	10.8
Qwen1.5-4B	native	18.0	24.0	34.4	25.5	22.0	26.0	40.4	29.5
	cohere	20.0	21.6	16.0	19.2	21.6	24.8	42.0	29.5
	google	17.6	22.4	35.6	25.2	21.2	21.6	38.0	26.9
Qwen1.5-7B	native	40.4	40.8	41.6	40.9	34.8	40.0	48.8	41.2
	cohere	37.2	40.4	36.8	38.1	37.2	39.6	45.6	40.8
	google	42.4	40.8	39.2	40.8	42.8	42.4	48.4	44.5
Qwen1.5-14B	native	44.8	60.0	53.6	52.8	49.2	59.6	56.8	55.2
	cohere	32.8	63.6	50.8	49.1	49.2	59.6	51.6	53.5
	google	42.4	60.0	54.0	52.1	50.4	63.2	55.6	56.4

Table 16: Results for each model and each language on MGSM (10^{-4}, exact token match, %).

Base model	Data	Monolingual				Multilingual			
		es	ru	zh	*average*	es	ru	zh	*average*
Llama2-7B	native	9.6	10.0	8.8	9.5	10.0	9.6	8.8	9.5
	cohere	10.4	11.2	8.0	9.9	10.0	10.4	9.2	9.9
	google	9.6	9.6	10.0	9.7	10.4	9.6	9.2	9.7
Gemma-2B	native	13.2	10.8	10.4	11.5	12.4	11.2	12.4	12.0
	cohere	12.4	10.4	12.0	11.6	12.4	11.6	12.8	12.3
	google	13.6	11.6	13.2	12.8	14.4	11.6	11.6	12.5
Gemma-7B	native	34.8	44.8	37.2	38.9	36.8	46.4	36.4	39.9
	cohere	35.6	44.4	36.4	38.8	34.8	45.2	38.0	39.3
	google	35.2	46.8	36.4	39.5	37.2	44.0	36.8	39.3
Qwen1.5-0.5B	native	2.4	2.8	8.8	4.7	2.4	2.4	8.4	4.4
	cohere	2.8	2.0	10.8	5.2	3.2	2.4	8.8	4.8
	google	2.8	1.6	6.8	3.7	2.4	2.4	8.8	4.5
Qwen1.5-1.8B	native	7.2	6.0	20.8	11.3	7.2	6.8	24.4	12.8
	cohere	8.0	6.8	20.4	11.7	5.6	6.0	23.2	11.6
	google	6.0	5.6	20.8	10.8	6.8	6.0	24.4	12.4
Qwen1.5-4B	native	21.6	28.0	40.0	29.9	20.8	28.0	40.0	29.6
	cohere	21.2	27.6	40.0	29.6	22.0	28.8	38.8	29.9
	google	22.8	28.8	41.6	31.1	21.2	28.0	38.0	29.1
Qwen1.5-7B	native	38.0	41.6	46.8	42.1	35.2	42.4	53.2	43.6
	cohere	36.0	41.2	47.2	41.5	34.8	40.0	48.8	41.2
	google	38.4	43.2	48.4	43.3	34.0	43.6	50.8	42.8
Qwen1.5-14B	native	46.0	63.6	58.0	55.9	47.2	62.8	58.0	56.0
	cohere	47.6	61.6	58.0	55.7	46.8	63.6	57.6	56.0
	google	46.8	62.4	58.0	55.7	48.0	61.6	57.6	55.7

Table 17: Results for each model and each language on MGSM (10^{-6}, exact token match, %).

Base model	Data	Monolingual				Multilingual			
		es	ru	zh	*average*	es	ru	zh	*average*
Llama2-7B	native	38.4	33.9	34.4	35.6	39.6	35.3	33.9	36.3
	cohere	38.0	34.6	27.6	33.4	37.8	31.6	32.9	34.1
	google	36.4	34.4	30.4	33.7	37.6	31.9	33.5	34.3
Gemma-2B	native	33.3	30.8	30.5	31.5	32.4	30.4	30.7	31.2
	cohere	30.8	30.3	29.6	30.2	33.7	31.9	32.8	32.8
	google	31.8	30.0	32.0	31.3	34.0	31.0	32.8	32.6
Gemma-7B	native	55.9	53.0	48.7	52.5	57.3	53.0	50.7	53.7
	cohere	58.4	53.8	50.8	54.4	58.7	55.3	52.9	55.6
	google	56.1	53.6	49.7	53.1	58.8	55.5	52.5	55.6
Qwen1.5-0.5B	native	30.2	27.1	35.3	30.9	29.4	26.5	35.2	30.4
	cohere	30.1	26.5	35.5	30.7	28.7	26.8	34.8	30.1
	google	32.4	26.2	36.9	31.8	29.5	26.6	34.6	30.2
Qwen1.5-1.8B	native	34.0	33.5	40.8	36.1	36.0	32.9	42.1	37.0
	cohere	35.4	32.2	40.9	36.2	36.3	32.2	42.0	36.8
	google	36.7	31.9	39.9	36.2	36.8	32.8	42.0	37.2
Qwen1.5-4B	native	40.9	38.2	49.3	42.8	45.5	38.7	49.6	44.6
	cohere	39.9	39.4	44.5	41.3	43.9	37.8	49.4	43.7
	google	39.6	38.9	44.2	40.9	43.3	36.2	49.0	42.8
Qwen1.5-7B	native	50.3	47.2	52.9	50.2	51.0	46.7	54.0	50.6
	cohere	49.6	46.7	52.6	49.6	52.0	47.3	54.3	51.2
	google	50.4	47.2	51.8	49.8	51.9	46.7	54.0	50.9
Qwen1.5-14B	native	58.1	55.4	61.3	58.3	58.6	54.9	61.5	58.3
	cohere	55.8	55.2	57.9	56.3	59.3	55.3	61.1	58.5
	google	54.9	55.5	58.0	56.1	59.1	55.2	60.4	58.2

Table 18: Results for each model and each language on MT-MMLU (10^{-4}, accuracy, %).

Base model	Data	Monolingual				Multilingual			
		es	ru	zh	*average*	es	ru	zh	*average*
Llama2-7B	native	39.1	34.5	33.9	35.8	38.9	34.7	33.7	35.8
	cohere	39.1	34.5	33.7	35.8	39.0	34.5	33.5	35.7
	google	39.1	34.5	33.9	35.8	38.9	34.3	33.6	35.6
Gemma-2B	native	31.8	30.8	31.7	31.4	31.7	30.6	31.6	31.3
	cohere	31.8	30.8	31.7	31.4	32.0	30.6	31.6	31.4
	google	32.0	30.8	31.7	31.5	31.8	30.7	31.7	31.4
Gemma-7B	native	56.5	53.6	51.0	53.7	56.7	53.7	51.5	54.0
	cohere	56.5	53.6	51.2	53.8	56.8	54.1	51.3	54.1
	google	56.2	53.9	51.7	54.0	57.3	54.0	51.2	54.1
Qwen1.5-0.5B	native	28.0	25.7	34.8	29.5	27.9	25.8	34.6	29.4
	cohere	27.9	25.9	34.5	29.5	28.2	25.9	34.5	29.5
	google	28.0	25.9	34.5	29.5	28.1	25.8	34.6	29.5
Qwen1.5-1.8B	native	36.1	31.8	41.4	36.4	35.9	31.8	41.6	36.5
	cohere	36.0	31.7	41.5	36.4	36.0	31.7	41.5	36.4
	google	36.2	31.8	41.3	36.4	36.1	31.7	41.3	36.4
Qwen1.5-4B	native	45.1	39.0	49.3	44.5	44.9	39.0	49.4	44.4
	cohere	45.0	38.9	49.3	44.4	44.7	39.1	49.3	44.4
	google	45.1	38.9	49.4	44.5	44.8	38.8	49.4	44.3
Qwen1.5-7B	native	51.3	46.4	53.3	50.3	51.0	46.3	53.1	50.1
	cohere	51.2	46.4	53.1	50.2	51.1	46.4	53.2	50.2
	google	51.2	46.3	52.9	50.1	51.0	46.2	53.2	50.2
Qwen1.5-14B	native	58.7	55.1	60.8	58.2	58.6	55.2	61.0	58.3
	cohere	58.7	55.1	61.1	58.3	58.6	55.1	61.0	58.2
	google	58.7	55.1	61.0	58.3	58.7	55.1	61.1	58.3

Table 19: Results for each model and each language on MT-MMLU (10^{-6}, accuracy, %).

Base Model	Data	Monolingual			Multilingual		
		es	zh	*average*	es	zh	*average*
Llama2-7B	native	37.6	33.8	35.7	39.0	33.5	36.3
	cohere	37.2	27.8	32.5	36.9	32.1	34.5
	google	35.9	29.5	32.7	37.1	32.5	34.8
Gemma-2B	native	33.3	31.1	32.2	32.3	31.0	31.7
	cohere	30.2	29.4	29.8	33.3	32.4	32.9
	google	31.1	31.8	31.4	33.6	33.0	33.3
Gemma-7B	native	54.9	48.0	51.5	56.2	50.4	53.3
	cohere	57.5	50.3	53.9	57.8	53.1	55.4
	google	55.6	48.8	52.2	57.7	52.7	55.2
Qwen1.5-0.5B	native	30.4	35.6	33.0	29.4	35.2	32.3
	cohere	29.7	34.7	32.2	29.0	34.4	31.7
	google	31.5	36.8	34.1	29.3	34.4	31.8
Qwen1.5-1.8B	native	33.2	40.7	37.0	35.8	42.6	39.2
	cohere	35.2	40.7	38.0	36.7	42.0	39.3
	google	36.3	40.0	38.1	37.1	42.3	39.7
Qwen1.5-4B	native	40.2	49.0	44.6	44.4	49.9	47.2
	cohere	39.0	45.2	42.1	43.2	49.2	46.2
	google	39.0	45.0	42.0	42.2	49.2	45.7
Qwen1.5-7B	native	49.6	53.0	51.3	50.4	53.4	51.9
	cohere	48.4	51.8	50.1	50.5	54.3	52.4
	google	49.3	51.3	50.3	50.5	53.3	51.9
Qwen1.5-14B	native	57.8	60.7	59.2	58.6	61.4	60.0
	cohere	55.1	57.7	56.4	58.7	61.3	60.0
	google	54.2	57.3	55.7	58.3	61.2	59.7

Table 20: Results for each model and each language on HT-MMLU (10^{-4}, accuracy, %).

Base Model	Data	Monolingual			Multilingual		
		es	zh	*average*	es	zh	*average*
Llama-2-7B	native	38.3	33.3	35.8	38.1	33.4	35.8
	cohere	38.2	33.1	35.6	38.1	33.5	35.8
	google	38.1	33.4	35.8	38.2	33.3	35.8
Gemma-2B	native	31.1	31.0	31.0	30.8	31.2	31.0
	cohere	31.2	31.3	31.2	31.2	31.4	31.3
	google	31.2	31.0	31.1	31.2	31.2	31.2
Gemma-7B	native	56.0	49.9	53.0	55.9	50.0	52.9
	cohere	55.5	50.3	52.9	55.8	49.6	52.7
	google	55.5	50.6	53.1	55.8	50.1	53.0
Qwen1.5-0.5B	native	28.5	34.6	31.5	28.4	34.6	31.5
	cohere	28.5	34.7	31.6	28.4	34.7	31.5
	google	28.4	34.5	31.4	28.4	34.6	31.5
Qwen1.5-1.8B	native	35.7	41.6	38.7	35.7	41.5	38.6
	cohere	35.7	41.2	38.4	35.7	41.2	38.5
	google	35.7	41.2	38.5	35.5	41.2	38.4
Qwen1.5-4B	native	43.7	49.6	46.7	43.9	49.5	46.7
	cohere	43.9	49.4	46.6	43.7	49.4	46.5
	google	43.8	49.6	46.7	43.7	49.4	46.5
Qwen1.5-7B	native	50.4	52.7	51.6	50.5	52.7	51.6
	cohere	50.5	52.7	51.6	50.5	52.8	51.6
	google	50.6	52.8	51.7	50.4	52.7	51.6
Qwen1.5-14B	native	58.3	61.0	59.6	58.5	61.1	59.8
	cohere	58.5	61.2	59.9	58.6	61.3	59.9
	google	58.4	61.1	59.8	58.5	61.2	59.9

Table 21: Results for each model and each language on HT-MMLU (10^{-6}, accuracy, %).

Base model	Data	Monolingual				Multilingual			
		es	ru	zh	*average*	es	ru	zh	*average*
Llama2-7B	native	203.0	180.0	131.5	171.5	134.0	129.0	102.0	121.7
	cohere	144.0	131.0	104.0	126.3	134.0	130.0	115.0	126.3
	google	140.0	122.0	115.0	125.7	148.0	125.0	120.0	131.0
Gemma-2B	native	161.5	151.0	125.0	145.8	122.0	112.0	109.0	114.3
	cohere	125.0	110.0	115.0	116.7	155.5	138.0	115.0	136.2
	google	110.0	118.0	119.0	115.7	126.0	116.0	130.0	124.0
Gemma-7B	native	224.5	230.0	195.0	216.5	172.0	161.0	161.0	164.7
	cohere	146.0	156.0	148.0	150.0	146.0	144.0	149.0	146.3
	google	168.0	157.0	147.0	157.3	151.0	144.0	147.0	147.3
Qwen1.5-0.5B	native	89.0	76.0	141.0	102.0	93.5	77.0	133.0	101.2
	cohere	70.0	60.0	99.0	76.3	78.0	61.0	107.0	82.0
	google	75.0	61.0	109.0	81.7	72.0	58.0	88.0	72.7
Qwen1.5-1.8B	native	119.5	112.0	148.0	126.5	105.0	104.0	162.0	123.7
	cohere	88.0	88.0	126.0	100.7	86.0	87.0	123.0	98.7
	google	87.0	80.0	108.0	91.7	101.5	82.0	115.0	99.5
Qwen1.5-4B	native	187.0	149.5	190.0	175.5	186.5	151.0	199.0	178.8
	cohere	107.0	108.0	137.0	117.3	123.0	109.0	156.0	129.3
	google	113.0	116.0	119.0	116.0	121.0	108.0	145.0	124.7
Qwen1.5-7B	native	196.0	180.5	186.0	187.5	188.0	178.0	202.0	189.3
	cohere	150.5	118.0	143.0	137.2	139.0	116.0	159.0	138.0
	google	134.0	121.0	143.0	132.7	132.0	111.0	158.0	133.7
Qwen1.5-14B	native	204.0	205.5	203.0	204.2	205.0	209.5	216.0	210.2
	cohere	142.0	151.0	165.0	152.7	150.0	129.0	158.0	145.7
	google	137.0	132.0	151.0	140.0	141.0	134.0	147.0	140.7

Table 22: Results for each model and each language on open-ended translated questions (GPT-4-Turbo judged).

Base model	Data	Monolingual				Multilingual			
		es	ru	zh	*average*	es	ru	zh	*average*
Llama2-7B	native	184.0	167.0	129.0	160.0	180.0	168.0	138.0	162.0
	cohere	174.5	162.0	147.0	161.2	183.0	166.0	150.0	166.3
	google	179.0	165.0	148.0	164.0	178.0	162.0	137.0	159.0
Gemma-2B	native	168.0	155.0	126.0	149.7	170.0	150.0	149.0	156.3
	cohere	166.0	154.0	152.0	157.3	166.0	160.0	151.0	159.0
	google	162.0	159.0	144.0	155.0	168.0	153.0	157.0	159.3
Gemma-7B	native	194.0	190.0	152.0	178.7	190.0	181.0	166.0	179.0
	cohere	171.0	174.0	168.0	171.0	178.0	161.0	161.0	166.7
	google	186.0	172.0	170.0	176.0	181.0	172.0	164.0	172.3
Qwen1.5-0.5B	native	131.0	96.0	131.0	119.3	121.0	99.0	138.0	119.3
	cohere	117.0	98.0	143.0	119.3	126.0	98.0	145.0	123.0
	google	133.0	104.0	133.0	123.3	127.0	102.0	129.0	119.3
Qwen1.5-1.8B	native	143.0	131.0	140.0	138.0	133.0	115.0	135.0	127.7
	cohere	147.0	128.0	152.0	142.3	145.0	121.0	159.0	141.7
	google	163.0	125.0	145.0	144.3	148.0	124.0	152.0	141.3
Qwen1.5-4B	native	179.0	160.0	155.0	164.7	169.0	149.0	170.0	162.7
	cohere	156.0	156.0	165.0	159.0	171.0	153.0	167.0	163.7
	google	168.0	155.0	151.0	158.0	157.0	147.0	158.0	154.0
Qwen1.5-7B	native	181.0	158.0	144.0	161.0	177.0	156.0	166.0	166.3
	cohere	178.0	159.0	154.0	163.7	183.0	148.0	160.0	163.7
	google	174.0	153.0	161.0	162.7	172.0	141.0	168.0	160.3
Qwen1.5-14B	native	183.0	172.0	156.0	170.3	182.0	173.0	169.0	174.7
	cohere	172.0	163.0	149.0	161.3	179.0	158.0	155.0	164.0
	google	172.0	159.0	155.0	162.0	172.0	155.0	160.0	162.3

Table 23: Results for each model and each language on open-ended translated questions (Command R+ judged).

Base model	Data	Monolingual				Multilingual			
		es	ru	zh	*average*	es	ru	zh	*average*
Llama2-7B	native	195.0	172.5	140.0	169.2	175.0	181.0	140.0	165.3
	cohere	173.0	172.0	149.0	164.7	158.0	152.0	135.0	148.3
	google	176.5	163.5	150.5	163.5	173.0	164.0	161.0	166.0
Gemma-2B	native	156.0	129.5	130.0	138.5	174.0	149.0	136.5	153.2
	cohere	175.0	144.0	160.0	159.7	160.0	152.5	155.0	155.8
	google	157.0	129.5	140.0	142.2	166.0	148.5	141.0	151.8
Gemma-7B	native	220.0	222.0	177.0	206.3	208.0	199.0	180.0	195.7
	cohere	197.0	209.5	207.0	204.5	206.0	175.5	194.0	191.8
	google	204.0	190.5	200.0	198.2	187.5	198.0	211.0	198.8
Qwen1.5-0.5B	native	88.0	88.0	157.0	111.0	91.0	76.0	162.0	109.7
	cohere	91.5	91.0	145.5	109.3	78.0	70.0	156.0	101.3
	google	93.0	66.0	142.0	100.3	91.0	72.0	157.0	106.7
Qwen1.5-1.8B	native	121.0	107.0	174.0	134.0	118.0	88.0	156.0	120.7
	cohere	116.0	102.0	177.0	131.7	121.0	108.0	193.5	140.8
	google	114.0	96.0	153.0	121.0	136.5	101.0	196.5	144.7
Qwen1.5-4B	native	162.0	126.0	186.0	158.0	163.0	129.0	208.0	166.7
	cohere	152.0	141.0	173.0	155.3	168.5	140.5	208.0	172.3
	google	146.0	137.0	191.0	158.0	161.0	133.0	213.0	169.0
Qwen1.5-7B	native	191.0	179.0	176.0	182.0	201.0	139.0	186.0	175.3
	cohere	194.0	175.5	205.5	191.7	185.0	165.0	213.0	187.7
	google	178.0	164.0	183.0	175.0	188.0	153.0	184.0	175.0
Qwen1.5-14B	native	206.0	144.0	197.0	182.3	204.5	184.0	219.0	202.5
	cohere	194.0	206.5	216.0	205.5	211.0	187.5	236.0	211.5
	google	177.0	167.0	222.5	188.8	197.0	182.0	212.0	197.0

Table 24: Results for each model and each language on open-ended native questions (GPT-4-Turbo judged).

Base model	Data	Monolingual				Multilingual			
		es	ru	zh	*average*	es	ru	zh	*average*
Llama2-7B	native	194.0	165.0	131.0	163.3	184.0	172.0	144.0	166.7
	cohere	173.0	160.0	145.0	159.3	184.0	163.0	149.0	165.3
	google	184.0	161.0	153.0	166.0	179.0	158.0	142.0	159.7
Gemma-2B	native	168.0	151.0	128.0	149.0	179.0	152.0	155.0	162.0
	cohere	176.0	152.0	155.0	161.0	178.0	155.0	151.0	161.3
	google	178.0	153.0	140.0	157.0	179.0	162.0	151.0	164.0
Gemma-7B	native	201.5	187.0	148.0	178.8	202.0	181.0	169.0	184.0
	cohere	183.0	172.0	162.0	172.3	180.0	162.0	168.0	170.0
	google	190.0	180.0	174.0	181.3	182.0	175.0	170.0	175.7
Qwen1.5-0.5B	native	133.0	103.0	153.0	129.7	136.0	96.0	150.0	127.3
	cohere	136.0	98.0	145.0	126.3	129.0	99.0	151.0	126.3
	google	130.0	97.0	145.0	124.0	125.0	102.0	157.0	128.0
Qwen1.5-1.8B	native	160.0	138.0	155.0	151.0	152.0	112.0	145.0	136.3
	cohere	151.0	121.0	170.0	147.3	149.0	118.0	167.0	144.7
	google	156.0	128.0	166.0	150.0	161.0	125.0	163.0	149.7
Qwen1.5-4B	native	180.0	152.0	161.0	164.3	181.0	149.0	171.0	167.0
	cohere	173.0	154.0	180.0	169.0	173.0	154.0	176.0	167.7
	google	168.0	149.0	171.0	162.7	182.0	147.0	177.0	168.7
Qwen1.5-7B	native	184.0	164.0	154.0	167.3	193.0	153.0	159.0	168.3
	cohere	184.0	154.0	160.0	166.0	184.0	146.0	173.0	167.7
	google	181.0	155.0	166.0	167.3	182.0	149.0	164.0	165.0
Qwen1.5-14B	native	191.0	160.0	152.0	167.7	192.0	167.0	169.0	176.0
	cohere	189.0	173.0	172.0	178.0	190.0	164.0	173.0	175.7
	google	184.0	163.0	178.0	175.0	188.0	160.0	165.0	171.0

Table 25: Results for each model and each language on open-ended native questions (Command R+ judged).