Beyond Bilingual Transfer: Multilingual Code-Switching in Instruction Tuning

Shunta Asano* **Jeonghun Baek*** **Toshihiko Yamasaki**
The University of Tokyo
asano@cvm.t.u-tokyo.ac.jp

Abstract

Recent studies have shown that code-switching data (CSD), in which multiple languages are mixed within the same context, can improve cross-lingual transfer and multilingual alignment in large language models (LLMs). Recent studies on synthetic code-switching for LLMs have primarily focused on bilingual transfer between English and a target language, leaving instruction-tuning settings that jointly mix three or more languages within each example largely unexplored. In this work, we investigate multilingual code-switching instruction tuning across four languages: English, Japanese, Korean, and Chinese. We evaluate multilingual understanding on Belebele and MMLU. Our experiments show that simple sentence-level multilingual CSD consistently improves the four-language average accuracy across all bilingual, trilingual, and quadrilingual settings for Gemma-2-2B, indicating that multilingual code-switching can be effective beyond bilingual transfer settings.

1 Introduction

Recent large language models (LLMs), including GPT-4 (Achiam et al., 2023), Gemini (Comanici et al., 2025), Llama 3 (Grattafiori et al., 2024), and Qwen3 (Yang et al., 2025), have demonstrated strong multilingual capabilities across a wide range of languages. However, while English data is abundant in pretraining corpora, many non-English languages remain relatively low-resource, often leading to inconsistent performance across languages (Yoo et al., 2025; Wang et al., 2025).

To address this challenge, recent studies have explored code-switching data (CSD), where multiple languages are mixed within the same context, to improve cross-lingual transfer and multilingual alignment (Yoo et al., 2025; Wang et al., 2025). Prior approaches such as CSCL (Yoo et al., 2025) and SynCS (Wang et al., 2025) demonstrated that synthetic CSD effectively improves bilingual transfer between English and a target language. However, existing LLM studies primarily focus on bilingual settings involving only two languages, such as English-Korean or English-Chinese transfer. As a result, it remains unclear whether CSD remains effective during LLM instruction tuning when three or more languages are jointly mixed within the same training example.

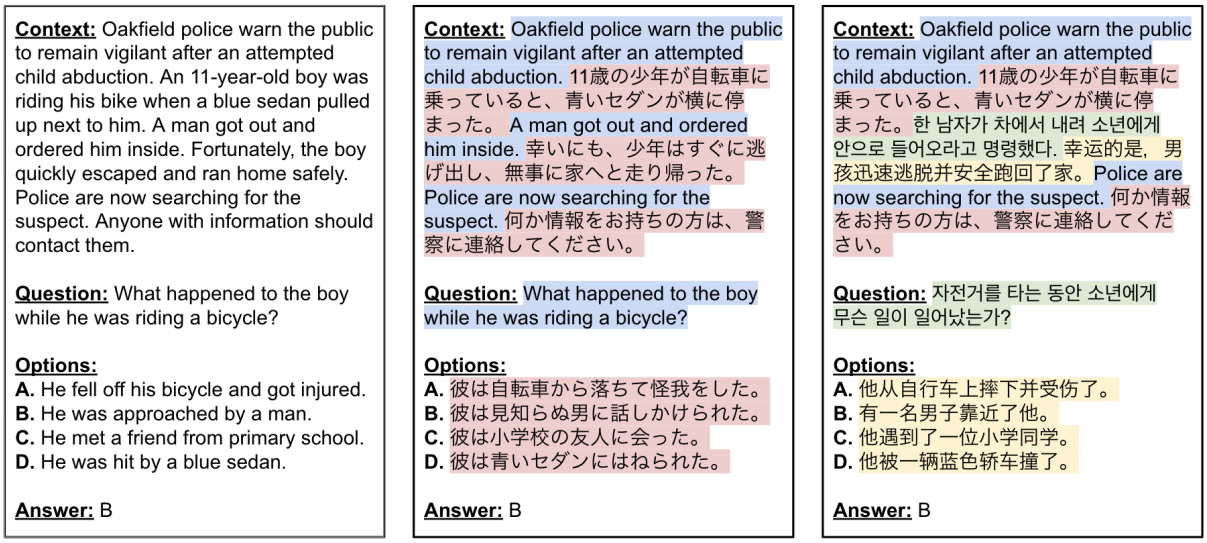
Moreover, prior work mainly investigates CSD during pretraining settings (Yoo et al., 2025; Wang et al., 2025). However, instruction tuning also plays an important role in multilingual capabilities (Wei et al., 2022; Ouyang et al., 2022; Chung et al., 2024). Despite this, multilingual CSD during instruction tuning remains underexplored.

In this work, we investigate multilingual code-switching instruction tuning across four languages: English, Japanese, Korean, and Chinese. We extend prior bilingual CSD approaches to multilingual settings and study whether CSD improves multilingual performance during instruction tuning. To evaluate multilingual understanding, we conduct experiments on the multilingual multiple-choice benchmarks Belebele (Bandarkar et al., 2024) and MMLU (Hendrycks et al., 2021).

For Gemma-2-2B, multilingual CSD consistently outperforms simple multilingual concatenation in average performance across bilingual, trilingual, and quadrilingual instruction-tuning settings on Belebele, with improvements also observed in trilingual and quadrilingual settings on MMLU. These results suggest that multilingual code-switching remains effective even in more complex multilingual settings beyond pairwise bilingual transfer. Our contributions are twofold. First, we extend synthetic code-switching during LLM instruction tuning beyond bilingual language pairs by jointly mixing three or more languages within the same example. Second, we show that

*Equal contribution.

Findings of the Association for Computational Linguistics: IJCNLP-AACL 2026, pages 23–34
November 6-10, 2026 ©2026 Association for Computational Linguistics



(a) Monolingual instruction data (b) Bilingual CSD (c) Multilingual CSD

Figure 1: Illustration of (a) monolingual instruction data, (b) bilingual sentence-level code-switching data (CSD), and (c) multilingual sentence-level CSD used in our work. Sentence-level CSD is constructed by combining sentence units from parallel multilingual instruction examples. Despite its simple construction process, sentence-level CSD remains effective for multilingual instruction tuning. In CSD examples, different background colors indicate different languages.

simple sentence-level CSD is effective for multilingual instruction tuning across four languages.

2 Related Work

Recent studies have explored code-switching data (CSD) to improve multilingual alignment and cross-lingual transfer in large language models (LLMs) (Winata et al., 2023; Yoo et al., 2025; Wang et al., 2025). Prior work showed that multilingual language models can implicitly learn cross-lingual alignment through shared multilingual representations (Pires et al., 2019). Recent studies further investigated synthetic code-switching, where multiple languages are mixed within the same context, to improve multilingual transfer and alignment.

CSCL (Yoo et al., 2025) introduced curriculum learning with synthetic code-switching data and demonstrated improved bilingual transfer between English and a target language. Similarly, SynCS (Wang et al., 2025) investigated synthetic code-switching during multilingual pretraining and showed improved cross-lingual alignment across several languages. However, SynCS constructs code-switching data only from bilingual language pairs between English and each target language, rather than jointly mixing three or more languages within the same example.

Earlier work such as CoSDA-ML (Qin et al., 2020) explored multilingual code-switching data augmentation for fine-tuning mBERT on zero-shot cross-lingual NLP tasks. Our work focuses on the empirical evaluation of code-switching involving three or more languages within individual examples during LLM instruction tuning, a setting that remains underexplored. While prior studies include token-, phrase-, and sentence-level code-switching strategies, we focus on simple sentence-level mixing that is easy to construct yet remains effective for multilingual instruction tuning.

3 Multilingual Code-Switching

We first generate English instruction data, translate the generated instruction examples into multiple languages, and then construct multilingual sentence-level CSD for instruction tuning.

3.1 Multilingual Instruction Data Construction

We use Dolma v1.6 (Soldaini et al., 2024), a large-scale open pretraining corpus, as the source of English text data for instruction generation, following the setup of CSCL (Yoo et al., 2025). We sample 50K English texts from Dolma while excluding the code-related *The Stack* subset.

Using `gpt-4o-mini` (OpenAI, 2024), chosen for its favorable balance between cost and generation quality, we generate multiple-choice instruction

data from sampled English texts. For each instruction example, we generate a multiple-choice question-answering pair consisting of a context snippet selected from the text, a question, four answer options, and the correct answer. We generate 3–5 instruction examples for each text. The prompt used for instruction data generation is provided in Table B.

Through this process, we construct approximately 200K English instruction examples comprising roughly 25M tokens. The resulting instruction examples contain 127.6 tokens on average, with lengths ranging from 57 to 586 tokens. Figure 1(a) illustrates an example of the generated English instruction data.

We then translate the generated instruction examples into Japanese (JA), Korean (KO), and Chinese (ZH) using `gpt-4o-mini`, resulting in parallel multilingual instruction data across four languages, including English (EN). The prompt used for translation is provided in Table C.

3.2 Sentence-Level Code-Switching Data

Following prior work on synthetic code-switching data (CSD) (Yoo et al., 2025), we construct sentence-level multilingual CSD for instruction tuning. Unlike prior bilingual settings that mainly focus on transfer between English and a target language, our setting jointly mixes three or more languages within the same instruction example.

We generate sentence-level multilingual CSD by selecting and combining sentence units from parallel instruction examples in different languages. We construct each context by mixing sentences from different language versions of the same example. The question and answer options are also selected from different language versions of the same example. To maintain answer consistency and readability, all answer options within a single example are written in the same language.

To determine the language order, we cycle through all permutations of the selected languages across examples: $3! = 6$ permutations in the trilingual setting and $4! = 24$ permutations in the quadrilingual setting. Within each example, we repeatedly follow the assigned language order for the context sentences, the question, and the answer options, treating the entire set of answer options as a single unit and preserving the original sentence order.

Figures 1(b) and (c) show examples of bilingual and multilingual sentence-level CSD, respectively. Sentence-level CSD is simpler to construct than more fine-grained token-level code-switching while remaining effective for multilingual instruction tuning.

4 Experiments and Analysis

4.1 Implementation Details

Model Setup. We use the pretrained Gemma-2-2B (Gemma Team et al., 2024) as our main model. We additionally evaluate the pretrained Qwen2-1.5B (Yang et al., 2024) in trilingual and quadrilingual settings on Belebele, with results reported in Table A in the appendix. Our training setup is based on the supervised fine-tuning (SFT) implementation in Hugging Face's TRL (von Werra et al., 2020) library. We train for one epoch with a learning rate of 5×10^{-5}, batch size 32, and a cosine learning rate scheduler. We employ parameter-efficient fine-tuning using LoRA (Hu et al., 2022) with rank 16.

Computational Resources and Training Time. All experiments are conducted using four NVIDIA A100 GPUs with a batch size of 32. Each model is trained for one epoch, which requires approximately one hour.

Comparison Methods. We compare multilingual CSD with concatenation baselines without code-switching to analyze the effectiveness of multilingual code-switching during instruction tuning. We construct bilingual, trilingual, and quadrilingual versions of CSD that always include English: (EN-JA, EN-KO, EN-ZH), (EN-JA-KO, EN-JA-ZH, EN-KO-ZH), and (EN-JA-KO-ZH). For comparison, we also construct concatenation baselines without code-switching. For controlled comparison, all settings use the same number of training examples (200K); only the way in which languages are combined differs. For example, in the EN-JA-KO-ZH concatenation baseline, the first 50K samples are in English, followed by 50K samples each in Japanese, Korean, and Chinese.

Evaluation Data. For evaluation, we use Belebele (Bandarkar et al., 2024), a multilingual multiple-choice reading comprehension benchmark in which the same questions are translated into multiple languages by experts fluent in English and the target language. We use 900 parallel test samples across EN, JA, KO, and ZH, enabling direct multilingual comparison across the four languages. An example of the Belebele test data is shown in Figure C. We additionally evaluate

Setting	Method	EN	JA	KO	ZH	Avg.
Baseline	Gemma-2-2B	34.4	27.4	29.7	27.8	29.8
E+J	Concat	82.8 ± 2.8	69.9 ± 2.3	69.4 ± 1.9	75.5 ± 3.3	74.4 ± 2.6
	CSD	83.4 ± 0.9	72.6 ± 1.4	71.5 ± 0.4	77.4 ± 1.0	76.2 ± 0.9
	Δ	$+0.7 \pm 2.1$	$+2.7 \pm 1.2$	$+2.0 \pm 1.6$	$+1.9 \pm 2.4$	$+1.8 \pm 1.7$
E+K	Concat	81.4 ± 0.8	70.6 ± 1.2	70.6 ± 1.0	76.2 ± 1.1	74.7 ± 1.0
	CSD	83.6 ± 1.1	72.4 ± 0.9	72.1 ± 1.1	77.3 ± 1.4	76.4 ± 0.9
	Δ	$+2.3 \pm 1.8$	$+1.9 \pm 1.4$	$+1.5 \pm 1.8$	$+1.1 \pm 2.5$	$+1.7 \pm 1.8$
E+Z	Concat	83.7 ± 1.5	70.5 ± 1.2	71.2 ± 0.4	77.5 ± 1.4	75.7 ± 1.0
	CSD	84.3 ± 1.5	71.0 ± 0.9	71.1 ± 0.8	78.1 ± 0.8	76.2 ± 0.5
	Δ	$+0.5 \pm 0.4$	$+0.5 \pm 1.9$	0.0 ± 0.9	$+0.7 \pm 1.4$	$+0.4 \pm 0.9$

Table 1: Results of bilingual training settings on Belebele for Gemma-2-2B (mean $\pm$ standard deviation over three runs). Concat denotes multilingual concatenation without code-switching, and CSD denotes code-switching data. Δ denotes the performance difference between CSD and Concat (CSD $-$ Concat). Positive improvements are highlighted in **green**. E, J, K, and Z denote English, Japanese, Korean, and Chinese, respectively.

Setting	Method	EN	JA	KO	ZH	Avg.
Baseline	Gemma-2-2B	34.4	27.4	29.7	27.8	29.8
E+J+K	Concat	80.6 ± 3.5	70.3 ± 2.4	69.2 ± 2.2	73.8 ± 3.3	73.5 ± 2.8
	CSD	84.0 ± 1.3	73.3 ± 1.4	72.5 ± 0.8	78.0 ± 1.2	77.0 ± 1.1
	Δ	$+3.4 \pm 2.2$	$+3.0 \pm 1.4$	$+3.2 \pm 1.4$	$+4.2 \pm 2.2$	$+3.5 \pm 1.8$
E+J+Z	Concat	81.9 ± 2.0	70.3 ± 2.7	69.8 ± 1.9	77.1 ± 1.4	74.8 ± 1.9
	CSD	83.4 ± 1.1	72.7 ± 0.6	72.9 ± 0.5	78.5 ± 1.2	76.9 ± 0.4
	Δ	$+1.5 \pm 3.0$	$+2.4 \pm 2.5$	$+3.2 \pm 2.2$	$+1.4 \pm 2.4$	$+2.1 \pm 2.3$
E+K+Z	Concat	79.0 ± 1.8	67.6 ± 1.6	67.1 ± 1.2	73.4 ± 0.8	71.8 ± 1.3
	CSD	83.9 ± 0.2	72.9 ± 0.3	72.4 ± 0.9	78.5 ± 0.3	76.9 ± 0.2
	Δ	$+4.9 \pm 1.7$	$+5.3 \pm 1.7$	$+5.3 \pm 2.1$	$+5.1 \pm 0.7$	$+5.2 \pm 1.5$
E+J+K+Z	Concat	80.1 ± 1.0	70.1 ± 1.4	68.7 ± 0.7	75.3 ± 1.1	73.5 ± 0.4
	CSD	85.0 ± 1.0	74.7 ± 0.2	73.0 ± 0.4	79.7 ± 1.2	78.1 ± 0.2
	Δ	$+4.9 \pm 1.4$	$+4.7 \pm 1.5$	$+4.3 \pm 0.4$	$+4.4 \pm 1.3$	$+4.6 \pm 0.5$

Table 2: Results of multilingual training settings on Belebele. E, J, K, and Z denote English, Japanese, Korean, and Chinese, respectively.

our method on MMLU (Hendrycks et al., 2021) across EN, JA, KO, and ZH. We use `cais/mmlu`[1] for English and the professionally translated versions from `openai/MMMLU`[2] for Japanese, Korean, and Chinese. We use the full test set of 14,042 questions per language.

Evaluation Metric. For each fine-tuned setting, we perform three runs with different random seeds. We compute accuracy (%) on the full evaluation set for each run and report the mean and standard deviation across the three runs. Avg. denotes the arithmetic mean of accuracy across the four evaluation languages. Following the EleutherAI LM Evaluation Harness (Biderman et al., 2024; Gao et al., 2024), we adopt likelihood-based multiple-choice evaluation instead of free-form generation for more stable multilingual evaluation. We compute the conditional likelihood of each answer choice and select the choice with the highest likelihood as the prediction.

4.2 Results on Bilingual Settings

Table 1 presents the results of bilingual training settings on Belebele. Overall, CSD consistently improves average performance compared to concatenation baselines across all bilingual settings.

English, Japanese, and Chinese improve across all bilingual settings. Korean improves in the E+J

[1] https://huggingface.co/datasets/cais/mmlu
[2] https://huggingface.co/datasets/openai/MMMLU

Setting	Method	EN	JA	KO	ZH	Avg.
Baseline	Gemma-2-2B	29.2	24.3	24.4	24.0	25.5
E+J+K	Concat	50.8 ± 2.0	39.5 ± 1.3	37.8 ± 0.7	41.0 ± 1.8	42.3 ± 1.5
	CSD	52.2 ± 0.4	40.5 ± 0.2	38.9 ± 0.2	42.3 ± 0.3	43.5 ± 0.3
	Δ	$+1.4 \pm 1.5$	$+1.0 \pm 1.2$	$+1.1 \pm 0.6$	$+1.2 \pm 1.5$	$+1.2 \pm 1.2$
E+J+Z	Concat	52.0 ± 0.6	40.3 ± 0.4	38.5 ± 0.3	42.1 ± 0.3	43.2 ± 0.2
	CSD	52.8 ± 0.5	40.8 ± 0.5	39.6 ± 0.2	43.0 ± 0.3	44.1 ± 0.2
	Δ	$+0.8 \pm 0.3$	$+0.4 \pm 0.7$	$+1.1 \pm 0.2$	$+0.9 \pm 0.1$	$+0.8 \pm 0.2$
E+K+Z	Concat	49.7 ± 0.6	38.3 ± 1.0	37.3 ± 0.9	40.3 ± 0.9	41.4 ± 0.9
	CSD	52.2 ± 0.4	40.4 ± 0.1	39.3 ± 0.2	42.4 ± 0.4	43.6 ± 0.2
	Δ	$+2.5 \pm 0.4$	$+2.1 \pm 0.9$	$+1.9 \pm 1.0$	$+2.1 \pm 0.5$	$+2.2 \pm 0.7$
E+J+K+Z	Concat	50.5 ± 0.8	39.6 ± 0.9	37.8 ± 0.9	41.2 ± 1.1	42.3 ± 0.9
	CSD	52.7 ± 0.2	40.9 ± 0.4	39.4 ± 0.3	42.8 ± 0.3	44.0 ± 0.2
	Δ	$+2.2 \pm 0.7$	$+1.3 \pm 0.7$	$+1.7 \pm 0.7$	$+1.7 \pm 0.9$	$+1.7 \pm 0.7$

Table 3: Results of multilingual training settings on MMLU.

and E+K settings and remains nearly unchanged in the E+Z setting. These results suggest that code-switching is more effective than simple concatenation for multilingual transfer.

These trends are consistent with prior studies on bilingual CSD. In particular, prior work such as CSCL and SynCS also reported improved multilingual transfer in EN+KO and EN+ZH settings, respectively. However, unlike prior work that mainly investigated bilingual CSD during large-scale pretraining settings (e.g., 1B tokens in CSCL and a 60B-English/600M-Chinese-token pretraining corpus with up to 2B additional synthetic code-switching tokens), our results show that bilingual CSD is also effective during instruction tuning with substantially smaller training data (approximately 25M tokens in the English source data).

4.3 Results on Multilingual Settings

Tables 2 and 3 present the results of trilingual and quadrilingual training settings on Belebele and MMLU, respectively. Compared to bilingual settings, multilingual CSD remains effective even when three or more languages are mixed within the same instruction example.

On Belebele, the quadrilingual setting (E+J+K+Z) achieves the highest average accuracy of 78.1%, improving over its concatenation baseline by 4.6 percentage points. Japanese improves by 4.7 percentage points in this setting. The largest average improvement occurs in the trilingual E+K+Z setting (+5.2 percentage points). These results suggest that the benefits of code-switching can extend beyond bilingual settings, even when three or more languages are jointly mixed within the same training example.

On MMLU, CSD improves average accuracy across all trilingual and quadrilingual settings, with gains ranging from 0.8 to 2.2 percentage points. The E+K+Z setting shows the largest average improvement, while the quadrilingual setting gains 1.7 percentage points. These results indicate that increasing the number of training languages does not necessarily yield larger improvements.

5 Conclusion

In this work, we investigated multilingual code-switching data (CSD) for multilingual instruction tuning across English, Japanese, Korean, and Chinese. Our experiments with Gemma-2-2B showed that multilingual sentence-level CSD consistently improves average performance across the four evaluation languages in bilingual, trilingual, and quadrilingual settings on Belebele, and in trilingual and quadrilingual settings on MMLU. The results further suggest that multilingual code-switching remains effective beyond bilingual transfer settings.

Additional experiments with Qwen2-1.5B (Appendix A) show smaller and less consistent gains, suggesting that the effectiveness of multilingual CSD depends on the underlying model and language combination. In addition, our findings show that simple sentence-level code-switching can provide consistent multilingual improvements during instruction tuning. We hope these findings will inform future work on multilingual instruction tuning and multilingual code-switching research.

Limitations

Our experiments focus on sentence-level code-switching data (CSD) across English and three East Asian languages (Chinese, Japanese, and Korean). Although we evaluate on both Belebele and MMLU, our evaluation is limited to multiple-choice benchmarks. Whether the observed improvements extend to open-ended multilingual generation remains an open question. Future work should investigate more fine-grained token- or phrase-level CSD, larger-scale training datasets, and more diverse multilingual settings covering broader language families and lower-resource languages.

Acknowledgments

This work used the UTokyo Azure computing service (Nakamura et al., 2025).

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and others. 2023. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*.

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 *ACL*, pages 749–775.

Stella Biderman, Hailey Schoelkopf, Lintang Sutawika, Leo Gao, Jonathan Tow, Baber Abbasi, Alham Fikri Aji, Pawan Sasanka Ammanamanchi, Sidney Black, Jordan Clive, and others. 2024. Lessons from the trenches on reproducible evaluation of language models. *arXiv preprint arXiv:2405.14782*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, and others. 2024. Scaling instruction-finetuned language models. *JMLR*, 25(70):1–53.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, and others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, and 5 others. 2024. The language model evaluation harness.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, and others. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and others. 2024. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *ICLR*.

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 *ICLR*.

Makoto Nakamura, Ryo Nakamura, Kazumi Arima, Kaoru Sasaki, Yoko Yamakata, Kenichiro Ito, Manabu Nishijima, Hiroshi Miyazaki, Tsutomu Ishizaki, Junji Tamatsukuri, Tomohiro Amemiya, Shigeru Chiba, and Kenjiro Taura. 2025. UTokyo Azure: Deployment of a cloud computing service at the university of tokyo. *Journal for Academic Computing and Networking*, 29(1):180–186.

OpenAI. 2024. GPT-4o mini: Advancing cost-efficient intelligence. https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/. Accessed: 2026-09-22.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and others. 2022. Training language models to follow instructions with human feedback. In *NeurIPS*, volume 35, pages 27730–27744.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In *ACL*, pages 4996–5001.

Libo Qin, Minheng Ni, Yue Zhang, and Wanxiang Che. 2020. CoSDA-ML: Multi-lingual code-switching data augmentation for zero-shot cross-lingual NLP. In *IJCAI*, pages 3853–3860.

Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, and others. 2024. Dolma: An open corpus of three trillion tokens for language model pretraining research. In *ACL*, pages 15725–15788.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Gallouédec. 2020. TRL: Transformers Reinforcement Learning. Accessed: 2026-09-22.

Zhijun Wang, Jiahuan Li, Hao Zhou, Rongxiang Weng, Jingang Wang, Xin Huang, Xue Han, Junlan Feng, Chao Deng, and Shujian Huang. 2025. Investigating and scaling up code-switching for multilingual language model pre-training. In *ACL Findings*, pages 11032–11046.

Jason Wei, Maarten Bosma, Vincent Y 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 *ICLR*.

Genta Indra Winata, Alham Fikri Aji, Zheng-Xin Yong, and Thamar Solorio. 2023. The decades progress on code-switching research in NLP: A systematic survey on trends and challenges. In *ACL Findings*, pages 2936–2978.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, and 43 others. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Haneul Yoo, Cheonbok Park, Sangdoo Yun, Alice Oh, and Hwaran Lee. 2025. Code-switching curriculum learning for multilingual transfer in LLMs. In *ACL Findings*, pages 7816–7836.

A Additional Results on Qwen2-1.5B

Table A reports the results of trilingual and quadrilingual instruction tuning on Belebele for Qwen2-1.5B. For Qwen2-1.5B, CSD improves average accuracy on Belebele in three of the four multilingual settings, with the largest gain of 0.8 percentage points in the quadrilingual setting. The E+J+Z setting shows a slight decrease of 0.1 percentage points. These results suggest that the benefits extend beyond Gemma-2-2B, although their magnitude and consistency depend on the model and language combination.

B Data Examples

B.1 Instruction Data Generation

Figure A shows an example of instruction data generation from a Dolma (Soldaini et al., 2024) text sample. Given a raw text passage, `gpt-4o-mini` (OpenAI, 2024) generates a context snippet selected from the text, a question, candidate options, and the corresponding correct answer in a multiple-choice question-answering format. This process enables the automatic construction of instruction-style data from unlabeled text resources.

B.2 Manual Translation Quality Check

Before creating code-switching and concatenation datasets, we manually inspected 20 translations by comparing them with their English sources. We checked whether each translation preserves the source meaning and whether the correct answer remains valid given the translated context, question, and answer options. This process serves as a qualitative sanity check rather than a quantitative assessment of translation quality.

B.3 Multilingual Code-Switching Data

Figure B illustrates examples of multilingual sentence-level CSD used in our work. Each example jointly mixes English, Japanese, Korean, and Chinese within the same instruction example while preserving the original question-answer structure. These examples demonstrate how multilingual instruction examples are constructed from parallel multilingual data.

B.4 Belebele Test Data

Figure C presents an example from the Belebele benchmark (Bandarkar et al., 2024) used as test data in our experiments. Belebele is a multilingual reading comprehension benchmark consisting of parallel question-answer pairs across multiple languages. The figure shows one example of the same question presented in the four languages, demonstrating the parallel structure of the benchmark and enabling direct multilingual comparison.

C Qualitative Analysis

Figure D shows examples where multilingual CSD correctly predicts the answer while the concatenation baseline fails. These examples suggest that multilingual code-switching can improve multilingual understanding by exposing the model to more diverse multilingual contexts during instruction tuning.

Figure E presents failure cases where multilingual CSD still produces an incorrect prediction. Despite the improvements brought by multilingual

Setting	Method	EN	JA	KO	ZH	Avg.
Baseline	Qwen2-1.5B	29.1	31.0	28.6	30.7	29.9
E+J+K	Concat	81.5 ± 1.1	67.1 ± 0.9	68.6 ± 0.4	78.1 ± 0.8	73.8 ± 0.6
	CSD	81.7 ± 0.7	68.6 ± 0.6	68.6 ± 0.6	78.5 ± 0.8	74.3 ± 0.7
	Δ	$+0.2 \pm 0.5$	$+1.5 \pm 0.3$	$+0.0 \pm 0.5$	$+0.4 \pm 0.2$	$+0.5 \pm 0.1$
E+J+Z	Concat	81.8 ± 0.6	68.2 ± 0.6	68.2 ± 0.2	79.5 ± 0.3	74.4 ± 0.3
	CSD	81.9 ± 0.9	68.4 ± 0.8	68.1 ± 1.0	79.0 ± 0.6	74.4 ± 0.7
	Δ	$+0.1 \pm 0.4$	$+0.2 \pm 1.1$	-0.1 ± 0.9	-0.6 ± 0.6	-0.1 ± 0.6
E+K+Z	Concat	82.0 ± 1.0	67.3 ± 0.8	67.4 ± 0.8	78.9 ± 0.8	73.9 ± 0.7
	CSD	82.0 ± 0.2	68.1 ± 0.8	68.4 ± 0.8	79.2 ± 0.2	74.4 ± 0.5
	Δ	$+0.0 \pm 0.8$	$+0.7 \pm 0.5$	$+1.0 \pm 1.4$	$+0.3 \pm 1.0$	$+0.5 \pm 0.8$
E+J+K+Z	Concat	81.6 ± 0.2	68.0 ± 0.3	67.8 ± 0.5	78.6 ± 0.5	74.0 ± 0.1
	CSD	81.8 ± 0.7	69.3 ± 1.7	68.7 ± 1.0	79.4 ± 0.4	74.8 ± 0.8
	Δ	$+0.2 \pm 0.6$	$+1.3 \pm 1.9$	$+0.9 \pm 0.6$	$+0.8 \pm 0.3$	$+0.8 \pm 0.7$

Table A: Results of multilingual training settings on Belebele for Qwen2-1.5B (mean ± standard deviation over three runs).

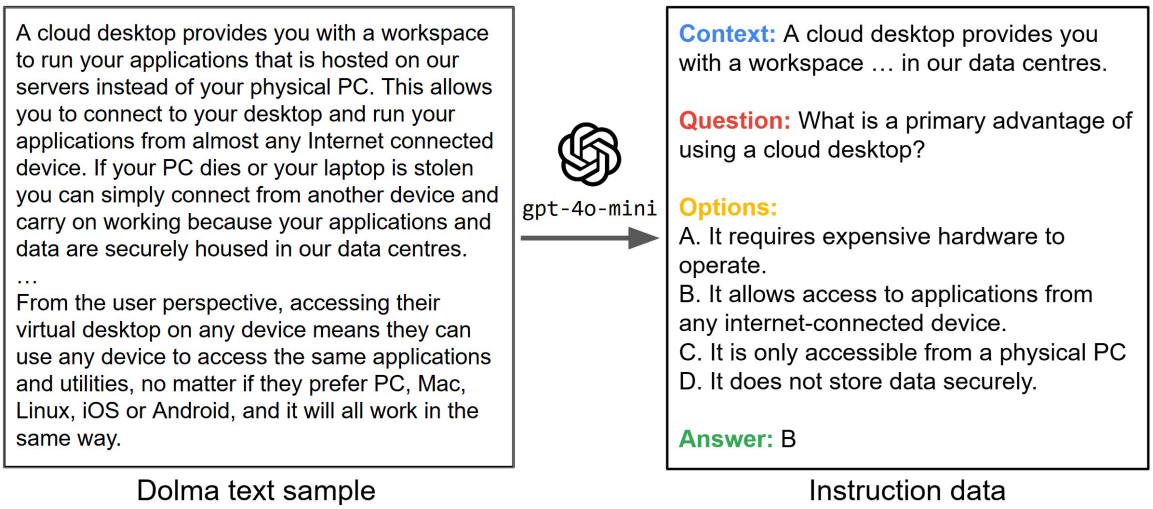


Figure A: Example of instruction data generation from a Dolma text sample using GPT-4o-mini.

CSD, such cases indicate that multilingual instruction tuning remains challenging, particularly for complex multilingual reasoning scenarios.

D Prompts

We use prompts for two stages: (1) instruction data generation from English Dolma data and (2) translation of the generated instruction data into Japanese, Korean, and Chinese.

First, instruction data is generated from English Dolma data using the prompt shown in Table B. The generated English instruction data is then translated into Japanese, Korean, and Chinese using the translation prompt shown in Table C.

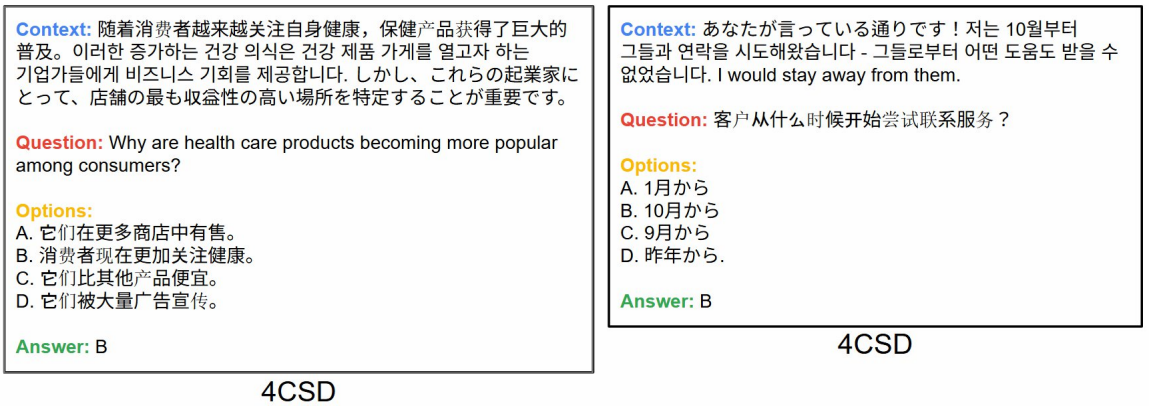


Figure B: Examples of multilingual sentence-level code-switching data (CSD) used in our work. The examples jointly mix English, Japanese, Korean, and Chinese within the same instruction example while preserving the original question-answer structure.



Figure C: Example of a parallel test sample from the Belebele benchmark. The same reading comprehension example is provided in English, Japanese, Korean, and Chinese.

Context: The major organ of the circulatory system is the heart, which pumps the blood. Blood goes away from the heart in tubes called arteries and comes back to the heart in tubes called veins. The smallest tubes are called capillaries.

Question: Which part of the circulatory system brings blood toward the heart?

Options:
A. Arteries
B. Capillaries
C. Organs
D. Veins

Answer: D
CSD4 Prediction: D
Concat4 Prediction: A

Success case

Context: Around 60 cases of malfunctioning iPods overheating have been reported, causing a total of six fires and leaving four people with minor burns. Japan's Ministry of Economy, Trade and Industry (METI) said that it had been aware of 27 accidents related to the devices. Last week, METI announced that Apple had informed it of 34 additional overheating incidents, which the company called "non-serious." The ministry responded by calling Apple's postponement of the report "truly regrettable."

Question: Who said that the overheating incidents were not very serious?

Options:
A. METI
B. The iPod owners involved
C. Japan's fire department
D. Apple

Answer: D
CSD4 Prediction: D
Concat4 Prediction: A

Success case

Figure D: Examples where multilingual CSD correctly predicts the answer while the concatenation baseline fails. CSD4 and Concat4 denote the quadrilingual multilingual CSD and concatenation settings, respectively.

Context: The Luno had 120–160 cubic metres of fuel aboard when it broke down and high winds and waves pushed it into the breakwater. Helicopters rescued the twelve crewmembers and the only injury was a broken nose. The 100-metre ship was en route to pick up its usual fertiliser cargo and initially officials feared the vessel could spill a load.

Question: According to the passage, which of the following statements about the Luno's breakdown is not true?

Options:
A. The breakdown occurred prior to picking up cargo
B. The ship had 100 meters of fuel aboard at the time of the breakdown
C. Out of twelve crew members, one suffered an injury
D. The ship was pushed into the breakwater

Answer: B
CSD4 Prediction: A
Concat4 Prediction: B

Failure case

Context: TMRI is based on a physics phenomenon called nuclear magnetic resonance (NMR), which was discovered in the 1930s by Felix Bloch (working at Stanford University) and Edward Purcell (from Harvard University). In this resonance, magnetic field and radio waves cause atoms to give off tiny radio signals. In the year 1970, Raymond Damadian, a medical doctor and research scientist, discovered the basis for using magnetic resonance imaging as a tool for medical diagnosis. Four years later a patent was granted, which was the world's first patent issued in the field of MRI. In 1977, Dr. Damadian completed the construction of the first "whole-body" MRI scanner, which he called the "Indomitable".

Question: In what year was the first patent granted for medical imaging resonance?

Options:
A. 1977
B. 1974
C. 1930
D. 1970

Answer: B
CSD4 Prediction: D
Concat4 Prediction: D

Failure case

Figure E: Failure cases where multilingual CSD still produces an incorrect prediction. Despite the improvements brought by multilingual CSD, complex multilingual reasoning and contextual understanding remain challenging.

System message

You are creating high-quality multiple-choice reading comprehension QA data for instruction tuning of large language models.
Requirements:
- Generate 3 to 5 diverse QA pairs from the provided raw text.
- For each QA pair, select a relevant context snippet from the raw text.
- Each QA pair should be independently understandable using its own context.
- Different QA pairs should cover different parts or ideas from the raw text when possible.
- Context snippets should typically be around 3 sentences long.
- Include enough information to answer the question naturally.

Question requirements:
- Questions must require reading and understanding the context.
- Do NOT use external knowledge.
- Avoid trivial or duplicate questions.
- Avoid questions answerable from common world knowledge alone.
- Prefer natural reading comprehension questions.
- Avoid copying sentences directly from the context.

Multiple-choice requirements:
- Each question must have exactly 4 options: A, B, C, D.
- Only one option should be correct.
- Incorrect options should be plausible and context-related.
- Avoid obviously incorrect or joke options.
- Keep option styles and lengths reasonably similar.

Return ONLY valid JSON.

Format: [{ "context": "...", "question": "...", "options": { "A": "...", "B": "...", "C": "...", "D": "..." }, "answer": "<CORRECT_OPTION>" }]

User prompt

Generate 3 to 5 diverse multiple-choice reading comprehension QA pairs from the following raw text.
Raw text:
{{TEXT}}

Table B: Prompt used for instruction data generation.

System message

You are a professional translator.

Task:
Translate English multiple-choice reading comprehension QA data into natural and fluent {language}.

Requirements:
- Translate the values of "context", "question", and all option texts.
- Do NOT translate JSON keys.
- Do NOT translate the answer option letter.
- Keep the JSON structure exactly unchanged.
- Do NOT remove, add, or alter newline characters ("\n").
- Preserve all option keys exactly as A, B, C, D.
- Do NOT reorder, remove, merge, or split options.
- Do NOT reorder, remove, merge, or split QA pairs.
- The output array length MUST exactly match the input array length.
- Ensure that the translated options remain consistent with the correct answer letter.
- Preserve factual meaning, numbers, dates, URLs, names, and entities.
- Do NOT use external knowledge.
- Return ONLY valid JSON.

Output format: [{ "context": "...", "question": "...", "options": { "A": "...", "B": "...", "C": "...", "D": "..." }, "answer": "<CORRECT_OPTION>" }]

User prompt

Translate the following English multiple-choice QA data into {language}.
Input JSON:
{{QA_JSON}}

Table C: Prompt used for translating instruction data into Japanese, Korean, and Chinese.