COMMIT: Code-Mixing English-Centric Large Language Model for Multilingual Instruction Tuning

Jaeseong Lee[1], **Yeonjoon Jung**[2], **Seung-won Hwang**[12*]
[1]Computer Science and Engineering, [2]IPAI
Seoul National University
{tbvj5914,y970120,seungwonh}@snu.ac.kr

Abstract

Recently, instruction-tuned large language models (LLMs) are showing prominent performance on various tasks, such as question answering. However, the majority of instruction-tuned LLMs are English-centric, which hinders their application to low-resource language QA. In this paper, we propose COde-Mixed Multilingual Instruction Tuning (COMMIT) to adapt English-centric LLM to low-resource language QA. We point out two main causes of English-centricness: imbalance of unlabeled data, and English-centric instruction tuning datasets. To deviate from English-centric instruction tuning, we propose to specialize code-mixing for instruction tuning, which blocks code-mixing in English templates, to leverage the potential of its superiority. To overcome data imbalance, we perform cross-lingual alignment. The majority of cross-lingual alignment works focused on making representations similar, which is not desirable to decoder-based LLMs, such as LLaMA. Therefore, we propose code-mixed continual causal language modeling to align the decoder. COMMIT improves the exact match score of low-resourced language QA by up to 32x. Code is publicly available.

1 Introduction

Recently, large language models (LLMs) have shown prominent performance on various natural language processing tasks (Brown et al., 2020; OpenAI, 2023; Touvron et al., 2023), such as question answering (QA). Moreover, instruction-tuning (Wang et al., 2022b; Taori et al., 2023; Wang et al., 2023) further updates the LLMs to be more efficient.

However, the majority of instruction-tuned LLMs are English-centric. The reasons are two-fold: both the pretraining corpora and the instruction-tuning datasets are English-centric Therefore, the performance of QA with low-resourced languages is lacking.

Resolving two would boost performance, but it is not trivial. First, to alleviate the former problem, the imbalance in unlabeled data, a naïve approach would be pretraining the LLM again with balanced data, which is tremendously costly (Zeng et al., 2023). Alternatively, cross-lingual alignment (Wu and Dredze, 2020; Alqahtani et al., 2021) can be considered. These methods focus on making the representations of different languages similar, particularly on encoder-based architectures such as mBERT (Devlin et al., 2019) or XLM-R (Conneau et al., 2020). However, for decoder-based LLMs, such as LLaMA (Touvron et al., 2023), similar representation across languages may confuse what language should decoder generate, thus such an approach is undesirable. Second, to deviate from English instruction tuning datasets, machine translation could be considered. However, assuming high-quality machine translation for low-resource languages can be impractical. Moreover, it ignores cross-lingual transferability from high-resource languages.

To overcome such shortcomings, in this paper, we propose **CO**de-**M**ixed **M**ultilingual **I**nstruction **T**uning (COMMIT). First, to efficiently utilize the English instruction tuning dataset, we code-mix it using the provided lexicon. Since a dictionary is much more available than machine translation (Wang et al., 2022a), it is more practical to assume a dictionary. Furthermore, code-mixing can leverage cross-lingual alignment (Lin et al., 2020).

While promising, we notice more room for improvement than naïvely performing code-mixing to the all part of the data. Thus, we specialize code-mixing for instruction tuning. Inspired by the fact that the English prompt is more effective even in multilingual LLMs (Muennighoff et al., 2023), we keep the template in English to preserve its strength,

[*]Corresponding author

Findings of the Association for Computational Linguistics: NAACL 2024, pages 3130–3137
June 16-21, 2024

without allowing code-mixing.

Second, to alleviate unlabeled data imbalance, we perform cross-lingual alignment beforehand. To align, we propose continual causal language modeling with code-mixed corpus, relying on the cross-lingual alignment ability of the code-mixing (Qin et al., 2020; Lin et al., 2020).

Experiments on MLQA (Lewis et al., 2020), and XQuAD (Artetxe et al., 2020) show the effectiveness of COMMIT–it increases the exact match up to 32x. Our code is publicly available.[1]

2 Related Works

2.1 Large Language Models

LLMs, which are pre-trained with language modeling over a large corpus, contain world knowledge (Zhao et al., 2023). To generalize world knowledge over diverse tasks such as question answering, LLMs reduce the gap between the pre-training and downstream tasks. Specifically, diverse tasks are formulated as language modeling, under which LLMs are pre-trained (Raffel et al., 2020). Additionally, LLMs adopt a decoder-only transformer which is specialized for the language modeling task (Zhao et al., 2023; Touvron et al., 2023).

2.2 Instruction Tuning for Non-English

For better generalization on unseen tasks, LLMs are instruction-tuned, fine-tuning to follow natural language instruction of such tasks (Chung et al., 2022). To generate such data for non-English languages, the simplest approach would be human annotation (Zhang et al., 2023), which is expensive. An alternative approach is to translate the instruction tuning data (Cui et al., 2023; Muennighoff et al., 2023; Li et al., 2023a; Santilli and Rodolà, 2023; Holmström and Doostmohammadi, 2023; Chen et al., 2023a,b; Lai et al., 2023; Li et al., 2023b) or utilize machine translation data (Zhu et al., 2023a; Ranaldi et al., 2023), or generation with an LLM (Wei et al., 2023). However, for low-resourced languages, high-quality translation or generation may not be available. In contrast, we assume the existence of a dictionary, which is a much more practical assumption (Wang et al., 2022a). Our proposed COMMIT can generate an instruction-tuning dataset for the target language, only relying on a dictionary.

[1] https://github.com/thnkinbtfly/COMMIT

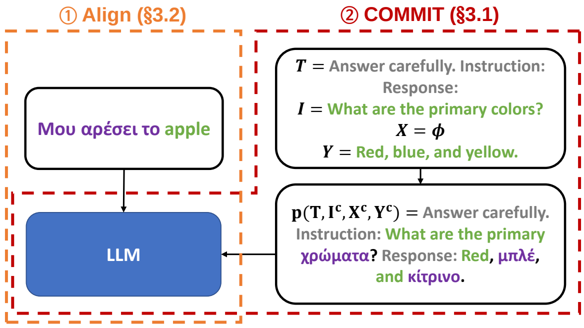

Figure 1: Overview of the proposed method, Align (§3.2) + COMMIT (§3.1). Grey represents the template, which is fixed, purple represents the target language, and green represents the replaceable English words.

3 Proposed Method

We assume that the given instruction tuning dataset is in English, and a dictionary is provided. This is a realistic scenario, considering the existing instruction tuning datasets (Taori et al., 2023; Wang et al., 2022b), and the availability of a dictionary (Wang et al., 2022a). We also assume that our English-centric LLM covers the majority of target language tokens, which is practical considering language contamination (Blevins and Zettlemoyer, 2022).

3.1 COMMIT: Specialized Code-Mixing for Instruction Tuning

We first formally define instruction tuning. For given instruction I, and input X, the model is expected to generate the specific output Y, with the aid of template T. X can be an empty string, while I must be a non-empty string, as exemplified in Figure 1. The model does language modeling with the sentence formulated as follows:

$$p(T, I, X); Y \tag{1}$$

where p is a function to put the words of I, X among T, and ; is the concatenation.

Recall that we take a practical assumption that T, I, X, Y are typically in English. Direct instruction tuning with the dataset would not efficiently transfer the knowledge to the target language. To efficiently utilize the English dataset for the target language, we may perform code-mixing. For $S \in \{T, I, X, Y\}$, let $S = [w_1, \cdots, w_n]$. For given dictionary $D = \{(w_i, t_i)\}$ between English and the target language, we generate code-mixed sentence S^c as follows:

$$x_i \sim B(\alpha) \tag{2}$$

$$c_i = \begin{cases} t_i & \text{if } x_i = 1, (w_i, t_i) \in D \\ w_i & \text{otherwise} \end{cases} \tag{3}$$

$$S^c = [c_1, \cdots, c_n] \tag{4}$$

where B is the bernoulli distribution, and α is the hyperparameter for it. The model may do language modeling with the sentence $p(T^c, I^c, X^c); Y^c$, which we call 'naïve code-mixing'.

While promising, we conjecture mixing all English words would hinder the transfer of the knowledge learned in English-centric LLM. It is known that English prompts show superior performance than prompts in the target language, even in multilingual pretrained language models (Lin et al., 2022; Muennighoff et al., 2023; Huang et al., 2023). Inspired, by this phenomenon, we propose to keep the template of instruction tuning in English, to preserve the strength of English prompts. To this end, we let the model do language modeling with the following sentence:

$$p(T, I^c, X^c); Y^c \tag{5}$$

3.2 Aligning Before COMMIT

COMMIT may improve the performance of instruction tuning, however directly performing COMMIT may not fully leverage cross-lingual ability in the given English-centric language model. It is known that even the multilingual pretrained language models do not fully leverage cross-lingual ability, therefore cross-lingual alignment has been proposed (Kulshreshtha et al., 2020; Alqahtani et al., 2021). We shift our view to this aspect.

We need to carefully select the cross-lingual align method, since the majority of them focus on encoder-based models, making the representation similar. This is undesirable for decoder-based models, since it would confuse the decoder with what language should it generate.

To this end, we choose code-mixing (Qin et al., 2020; Lin et al., 2020) as a tool for cross-lingual alignment. Since it does not explicitly force the language model to make representation similar, such confusion would be reduced. Formally, before performing COMMIT, given the sentences of the corpus in target language C, we first construct the code-mixed corpus C^c, similarly to Eq. 4. Then we perform continual causal language modeling with the following objective:

lang (iso code)	lang family	# wiki	ling.sim
Greek (el)	Indo-European	209K	0.729
Thai (th)	Tai-Kadai	147K	0.712
Hindi (hi)	Indo-European	151K	0.683
Bengali (bn)	Indo-European	121K	0.680
Tamil (ta)	Dravidian	146K	0.620

Table 1: Languages used for the experiments in this paper. We report the size of the unlabeled dataset (# wiki), and linguistic similarity with the English.

$$L_{align} = -\frac{1}{N} \sum_i logP(c_i^c | c_{<i}^c) \tag{6}$$

where $C^c = [c_1^c, \cdots, c_N^c]$, $c_{<i}^c = [c_1^c, \cdots, c_{i-1}^c]$.

4 Experiments

4.1 Experimental Settings

We use LLaMA-7B (Touvron et al., 2023) as our representative English-centric large language model.

Tasks and Datasets For instruction tuning, we use the ALPACA dataset (Taori et al., 2023), and for continual causal language modeling, we utilize Wikipedia corpus.[2] For code-mixing, we use the MUSE dictionary (Lample et al., 2018).

We evaluate our model on the extended version of LM-EVALUATION-HARNESS (Gao et al., 2021).[3] We select the available QA datasets: MLQA (Lewis et al., 2020), and XQuAD (Artetxe et al., 2020). We also implement IndicQA (Doddapaneni et al., 2023), which additionally requests unanswerable question classification, differently from MLQA or XQuAD.

Language selection Among languages with given QA datasets and dictionaries, we choose languages with less than 250K Wikipedia articles, which are the five least-resourced languages: Greek (el), Hindi (hi), Thai (th), Tamil (ta), and Bengali (bn). These languages are not covered in the pretraining of LLaMA (Touvron et al., 2023). We describe the size of the unlabeled dataset, and linguistic similarity with English,[4] in Table 1.

[2]https://huggingface.co/datasets/graelo/wikipedia

[3]https://github.com/OpenGPTX/lm-evaluation-harness

[4]Following Ansell et al. (2021) we take the cosine similarity of URIEL feature vectors (Littell et al., 2017) to calculate the linguistic similarity between languages.

	MLQA		XQuAD							
	hi EM	hi F1	hi EM	hi F1	th EM	th F1	el EM	el F1	EM avg	F1 avg
LLaMA	0.35	5.93	0.59	6.85	0.08	2.38	1.09	7.93	0.53	5.77
Alpaca	0.28	7.95	0.00	8.10	0.25	3.76	1.01	11.82	0.39	7.91
LLaMA+En prompt	0.79	7.21	1.09	7.28	0.08	2.80	3.45	10.83	1.35	7.03
Alpaca+En prompt	1.12	9.78	1.34	10.48	1.34	4.86	3.36	14.98	1.79	10.03
COMMIT+En prompt	2.56	7.89	3.87	8.99	2.18	3.89	7.39	15.18	4.00	8.99
COMMIT	4.35	9.26	6.22	10.41	4.37	7.30	**9.92**	**18.12**	6.22	11.27
Align+COMMIT	**6.04**	**14.77**	**7.56**	**14.72**	**8.15**	**13.84**	8.57	16.19	**7.58**	**14.88**

Table 2: Exact match and F1 score of COMMIT and comparisons. Best scores are emphasized with bold.

	MLQA	XQUAD			
	hi	hi	th	el	avg
COMMIT	4.35	6.22	4.37	9.92	6.22
CLM+COMMIT	4.19	4.96	7.39	6.22	5.69
Align+COMMIT	6.04	7.56	8.15	8.57	**7.58**

Table 3: Exact match score of aligning with code-mix, or simply consuming data with CLM, before COMMIT.

Implementation Details To perform instruction tuning, we largely follow the setting from Alpaca (Taori et al., 2023).[5] We use learning rate of 2e-5; sequence length of 512; warmup for 3% of total steps; and train for 3 epochs. We use α of 0.9 for code-mixing.[6] We perform continual causal language modeling with similar hyperparameters, except that we train for 10K steps. We use α of 0.5 for code-mixing. COMMIT is performed on TPUv3-8, taking less than 8 hours in total. The code is based on EasyLM (Geng, 2023), implemented with JAX (Bradbury et al., 2018).

We evaluate the LLMs with a batch size of 2, in a zero-shot manner. Evaluation is conducted on RTX3090, which takes less than an hour.

Baselines We compare COMMIT with the following baselines. a) **LLaMA:** The baseline LLM; b) **Alpaca:** The baseline instruction-tuned LLM; c) **LLaMA/Alpaca+En Prompt:** We try English prompt instead of prompt in the target language, since they are known to perform better (Lin et al., 2022; Huang et al., 2023); d) **naïve codemix:** We use naïve code-mix, described in §3.1; e) **Machine Translation:** We use Google Translate API to translate the instruction tuning dataset.

[5] `https://github.com/tatsu-lab/stanford_alpaca`

[6] We probed {0.8,0.9,1.0} since large code-mix ratio is preferred in language adaptation (Wang et al., 2022a), and selected based on MLQA val EM score.

4.2 Experimental Results

Superiority of COMMIT COMMIT outperforms the baselines (Table 2). For example, XQuAD th EM of Align+COMMIT is more than 32x larger than LLaMA or Alpaca. Using English prompts does improve the performance, however, COMMIT even outperforms this tough baseline. For example, XQuAD th EM score or MLQA hi EM score of COMMIT is about 6x larger than the baselines with English prompts.

Overall, the average scores of Align+COMMIT is the best among the comparisons (Table 2).The exception of a lowered score of Greek (el) can be explained by the linguistic similarity with English (Table 1). Since Greek is showing the maximum similarity, the LLM is already aligned well; additional alignment may harm the language model. Note that the similarity score does not perfectly correlate with the performance gain (e.g. th vs hi), however combined with linguistic genealogy, we can roughly explain the trend. We leave the improving the quality of the similarity metric as a future work.

English prompt is not needed Surprisingly, COMMIT favors target language prompts over English prompts (Table 2), which implies COMMIT effectively adapted the model to the target language. This favor is more desirable for real-world use cases, which is different from the known fact that LLMs favor English prompts (Lin et al., 2022; Huang et al., 2023).

Efficiency of aligning beforehand One may question whether the improvement simply comes from an increase in data. Table 3 discloses that simply consuming the target language corpus with causal language modeling (CLM) even lowers the average score, ruining the language model. In contrast, our approach efficiently utilizes the corpus, improving the performance.

	MLQA	XQUAD			
	hi	hi	th	el	avg
Alpaca (no code-mix)	0.28	0.00	0.25	1.01	0.39
naïve code-mix	3.90	5.21	2.10	8.99	5.05
COMMIT	4.35	6.22	4.37	9.92	**6.22**

Table 4: Exact match score of specialized code-mix of COMMIT, naïve code-mix, and no code-mixing.

	MLQA	XQuAD			
	hi	hi	th	el	avg
Machine Translation	5.19	2.52	8.57	6.39	5.67
Align+COMMIT	6.04	7.56	8.15	8.57	**7.58**

Table 5: Exact match score of COMMIT and instruction tuning with machine translation.

Effectiveness of specialized code-mix Our specialization of code-mixing for instruction tuning is effective (Table 4). While naïve code-mixing improves the performance over not performing it, COMMIT outperforms naïve code-mixing.

Outperforming Machine Translation COMMIT outperforms MT baseline (Table 5). This may look counter-intuitive, but consistent observation was made (Ranaldi et al., 2023), benefiting from cross-lingual alignment during instruction-tuning. Based on this observation, we re-emphasize our contribution: Our proposed code-mixing, by using only a dictionary, enables cross-lingual alignment (Lin et al., 2020) during the instruction tuning, even outperforming compute-intensive MT-instruction-tuning.

Observation consistent on IndicQA When we extend our evaluation to include classification of unanswerable questions, utilizing IndicQA, the observations are consistent (Table 6). Align+COMMIT outperforms the baselines, COMMIT, and machine translation.

	ta	bn	avg
LLaMA	18.51	15.83	17.17
Alpaca	20.62	16.00	18.31
LLaMA+En prompt	19.96	15.94	17.95
Alpaca+En prompt	19.24	15.71	17.47
Machine Translation	22.67	17.87	20.27
COMMIT	22.28	17.92	20.10
Align+COMMIT	**24.45**	**20.25**	**22.35**

Table 6: Exact match score of COMMIT and comparisons on IndicQA.

5 Conclusion

We studied adapting English-centric LLM to low-resource language QA. We proposed Align+COMMIT, aligning and then performing a specialized code-mixing method for instruction tuning. Experiments show that each component contributes to improving the performance.

6 Limitation

In this work, we followed the most common way to code-mix the data (Qin et al., 2020; Lin et al., 2020). Considering context or morphology during code-mixing would be beneficial (Feng et al., 2022; Zhu et al., 2023b).

However, considering context or morphology is not necessary to claim the strength of our proposed method, as COMMIT outperforms machine translation, a solution scarcely violates such context or morphology. We would probe better code-mixing strategy (Feng et al., 2022; Zhu et al., 2023b) or optimization techniques such as LoRA (Hu et al., 2022) as future work.

Acknowledgements

This research was partially supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2024-2020-0-01789) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation). This work was also partially supported by IITP grant funded by MSIT (No.2022-0-00077, AI Technology Development for Commonsense Extraction, Reasoning, and Inference from Heterogeneous Data and No.2021-0-01343-004, Artificial Intelligence Graduate School Program (Seoul National University)]. We would also like to thank Google's TPU Research Cloud (TRC) program for providing Cloud TPUs.

References

Sawsan Alqahtani, Garima Lalwani, Yi Zhang, Salvatore Romeo, and Saab Mansour. 2021. Using Optimal Transport as Alignment Objective for fine-tuning Multilingual Contextualized Embeddings. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3904–3919, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Alan Ansell, Edoardo Maria Ponti, Jonas Pfeiffer, Sebastian Ruder, Goran Glavaš, Ivan Vulić, and Anna Korhonen. 2021. MAD-G: Multilingual Adapter Generation for Efficient Cross-Lingual Transfer. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4762–4781, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020. On the Cross-lingual Transferability of Monolingual Representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4623–4637, Online. Association for Computational Linguistics.

Terra Blevins and Luke Zettlemoyer. 2022. Language Contamination Helps Explains the Cross-lingual Capabilities of English Pretrained Models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3563–3574, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. 2018. JAX: Composable transformations of Python+NumPy programs.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Pinzhen Chen, Shaoxiong Ji, Nikolay Bogoychev, Barry Haddow, and Kenneth Heafield. 2023a. Monolingual or Multilingual Instruction Tuning: Which Makes a Better Alpaca.

Wei-Lin Chen, Cheng-Kuang Wu, and Hsin-Hsi Chen. 2023b. Traditional-chinese alpaca: Models and datasets.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling Instruction-Finetuned Language Models.

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

Yiming Cui, Ziqing Yang, and Xin Yao. 2023. Efficient and Effective Text Encoding for Chinese LLaMA and Alpaca.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Sumanth Doddapaneni, Rahul Aralikatte, Gowtham Ramesh, Shreya Goyal, Mitesh M. Khapra, Anoop Kunchukuttan, and Pratyush Kumar. 2023. Towards Leaving No Indic Language Behind: Building Monolingual Corpora, Benchmark and Models for Indic Languages. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12402–12426, Toronto, Canada. Association for Computational Linguistics.

Yukun Feng, Feng Li, and Philipp Koehn. 2022. Toward the Limitation of Code-Switching in Cross-Lingual Transfer. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5966–5971, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Leo Gao, Jonathan Tow, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Kyle McDonell, Niklas Muennighoff, Jason Phang, Laria Reynolds, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2021. A framework for few-shot language model evaluation. Zenodo.

Xinyang Geng. 2023. EasyLM: A simple and scalable training framework for large language models.

Oskar Holmström and Ehsan Doostmohammadi. 2023. Making Instruction Finetuning Accessible to Non-English Languages: A Case Study on Swedish Models. In *Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 634–642, Tórshavn, Faroe Islands. University of Tartu Library.

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*.

Haoyang Huang, Tianyi Tang, Dongdong Zhang, Xin Zhao, Ting Song, Yan Xia, and Furu Wei. 2023. Not all languages are created equal in LLMs: Improving multilingual capability by cross-lingual-thought prompting. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 12365–12394, Singapore. Association for Computational Linguistics.

Saurabh Kulshreshtha, Jose Luis Redondo Garcia, and Ching-Yun Chang. 2020. Cross-lingual Alignment Methods for Multilingual BERT: A Comparative Study. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 933–942, Online. 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*, pages 318–327, Singapore. Association for Computational Linguistics.

Guillaume Lample, Alexis Conneau, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018. Word translation without parallel data. In *International Conference on Learning Representations*.

Patrick Lewis, Barlas Oguz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. 2020. MLQA: Evaluating Cross-lingual Extractive Question Answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7315–7330, Online. Association for Computational Linguistics.

Haonan Li, Fajri Koto, Minghao Wu, Alham Fikri Aji, and Timothy Baldwin. 2023a. Bactrian-X: Multilingual Replicable Instruction-Following Models with Low-Rank Adaptation.

Lei Li, Yuwei Yin, Shicheng Li, Liang Chen, Peiyi Wang, Shuhuai Ren, Mukai Li, Yazheng Yang, Jingjing Xu, Xu Sun, Lingpeng Kong, and Qi Liu. 2023b. M$\hat{3}$IT: A Large-Scale Dataset towards Multi-Modal Multilingual Instruction Tuning.

Xi Victoria Lin, Todor Mihaylov, Mikel Artetxe, Tianlu Wang, Shuohui Chen, Daniel Simig, Myle Ott, Naman Goyal, Shruti Bhosale, Jingfei Du, Ramakanth Pasunuru, Sam Shleifer, Punit Singh Koura, Vishrav Chaudhary, Brian O'Horo, Jeff Wang, Luke Zettlemoyer, Zornitsa Kozareva, Mona Diab, Veselin Stoyanov, and Xian Li. 2022. Few-shot Learning with Multilingual Generative Language Models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9019–9052, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Zehui Lin, Xiao Pan, Mingxuan Wang, Xipeng Qiu, Jiangtao Feng, Hao Zhou, and Lei Li. 2020. Pre-training Multilingual Neural Machine Translation by Leveraging Alignment Information. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2649–2663, Online. Association for Computational Linguistics.

Patrick Littell, David R. Mortensen, Ke Lin, Katherine Kairis, Carlisle Turner, and Lori Levin. 2017. URIEL and lang2vec: Representing languages as typological, geographical, and phylogenetic vectors. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 8–14, Valencia, Spain. 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, Xiangru Tang, Dragomir Radev, Alham Fikri Aji, Khalid Almubarak, Samuel Albanie, Zaid Alyafeai, Albert Webson, Edward Raff, and Colin Raffel. 2023. Crosslingual Generalization through Multitask Finetuning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15991–16111, Toronto, Canada. Association for Computational Linguistics.

OpenAI. 2023. GPT-4 Technical Report.

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 *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 3853–3860. International Joint Conferences on Artificial Intelligence Organization.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Leonardo Ranaldi, Giulia Pucci, and Andre Freitas. 2023. Empowering Cross-lingual Abilities of Instruction-tuned Large Language Models by Translation-following demonstrations.

Andrea Santilli and Emanuele Rodolà. 2023. Camoscio: An Italian Instruction-tuned LLaMA.

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.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and Efficient Foundation Language Models.

Xinyi Wang, Sebastian Ruder, and Graham Neubig. 2022a. Expanding Pretrained Models to Thousands More Languages via Lexicon-based Adaptation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 863–877, Dublin, Ireland. Association for Computational Linguistics.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-Instruct: Aligning Language Models with Self-Generated Instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13484–13508, Toronto, Canada. Association for Computational Linguistics.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Kuntal Kumar Pal, Maitreya Patel, Mehrad Moradshahi, Mihir Parmar, Mirali Purohit, Neeraj Varshney, Phani Rohitha Kaza, Pulkit Verma, Ravsehaj Singh Puri, Rushang Karia, Savan Doshi, Shailaja Keyur Sampat, Siddhartha Mishra, Sujan Reddy A, Sumanta Patro, Tanay Dixit, and Xudong Shen. 2022b. Super-NaturalInstructions: Generalization via Declarative Instructions on 1600+ NLP Tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5085–5109, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Xiangpeng Wei, Haoran Wei, Huan Lin, Tianhao Li, Pei Zhang, Xingzhang Ren, Mei Li, Yu Wan, Zhiwei Cao, Binbin Xie, Tianxiang Hu, Shangjie Li, Binyuan Hui, Bowen Yu, Dayiheng Liu, Baosong Yang, Fei Huang, and Jun Xie. 2023. PolyLM: An Open Source Polyglot Large Language Model.

Shijie Wu and Mark Dredze. 2020. Do Explicit Alignments Robustly Improve Multilingual Encoders? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4471–4482, Online. Association for Computational Linguistics.

Qingcheng Zeng, Lucas Garay, Peilin Zhou, Dading Chong, Yining Hua, Jiageng Wu, Yikang Pan, Han Zhou, Rob Voigt, and Jie Yang. 2023. GreenPLM: Cross-Lingual Transfer of Monolingual Pre-Trained Language Models at Almost No Cost. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, pages 6290–6298, Macau, SAR China. International Joint Conferences on Artificial Intelligence Organization.

Shaolei Zhang, Qingkai Fang, Zhuocheng Zhang, Zhengrui Ma, Yan Zhou, Langlin Huang, Mengyu Bu, Shangtong Gui, Yunji Chen, Xilin Chen, and Yang Feng. 2023. BayLing: Bridging Cross-lingual Alignment and Instruction Following through Interactive Translation for Large Language Models.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. A Survey of Large Language Models.

Wenhao Zhu, Yunzhe Lv, Qingxiu Dong, Fei Yuan, Jingjing Xu, Shujian Huang, Lingpeng Kong, Jiajun Chen, and Lei Li. 2023a. Extrapolating Large Language Models to Non-English by Aligning Languages.

Zhihong Zhu, Xuxin Cheng, Zhiqi Huang, Dongsheng Chen, and Yuexian Zou. 2023b. Enhancing code-switching for cross-lingual SLU: A unified view of semantic and grammatical coherence. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7849–7856, Singapore. Association for Computational Linguistics.