Monolingual or Multilingual Instruction Tuning : Which Makes a Better Alpaca

Chen, Pinzhen

2024

Chen, P, Ji, S, Bogoychev, N, Kutuzov, A, Haddow, B & Heafield, K 2024, Monolingual or Multilingual Instruction Tuning : Which Makes a Better Alpaca. in Y Graham & M Purver (eds), Findings of the Association for Computational Linguistics : EACL 2024. Association for Computational Linguistics (ACL), Kerrville, pp. 1347-1356, Conference of the European Chapter of the Association for Computational Linguistics, St. Julians, Malta, 17/03/2024.

http://hdl.handle.net/10138/574139

Monolingual or Multilingual Instruction Tuning: Which Makes a Better Alpaca

Pinzhen Chen[1,*] **Shaoxiong Ji**[2,*] **Nikolay Bogoychev**[1]
Andrey Kutuzov[3] **Barry Haddow**[1] **Kenneth Heafield**[1]
[1]University of Edinburgh [2]University of Helsinki [3]University of Oslo
pchen3@ed.ac.uk shaoxiong.ji@helsinki.fi

Abstract

Foundational large language models (LLMs) can be instruction-tuned to perform open-domain question answering, facilitating applications like chat assistants. While such efforts are often carried out in a single language, we empirically analyze cost-efficient strategies for multilingual scenarios. Our study employs the Alpaca dataset and machine translations of it to form multilingual data, which is then used to tune LLMs through either low-rank adaptation or full-parameter training. Under a controlled computation budget, comparisons show that multilingual tuning is on par or better than tuning a model for each language. Furthermore, multilingual tuning with downsampled data can be as powerful and more robust. Our findings serve as a guide for expanding language support through instruction tuning.

1 Introduction

Language capacity has attracted much attention in pre-trained language models. Some pioneering works focused on a single language (Peters et al., 2018; Devlin et al., 2019), while later works aim to cover multiple languages (Conneau et al., 2020; Liu et al., 2020). In the recent blossom of open-source LLMs, English-centric ones include GPT-2, LLaMA, and Pythia (Radford et al., 2019; Touvron et al., 2023; Biderman et al., 2023), and multilingual ones are represented by BLOOM (Scao et al., 2022). Multilingual models seem attractive when considering operational costs, cross-lingual transfer, and low-resource languages (Artetxe and Schwenk, 2019; Wu and Dredze, 2020), yet English-centric models can possess good multilingual transferability (Ye et al., 2023).

Instruction tuning makes LLMs follow and respond to inputs (Sanh et al., 2022; Wei et al., 2022). With multilingual instruction data becoming feasible and available, this paper compares monolingual and multilingual instruction tuning applied to English-centric and multilingual LLMs to search for the optimal strategy to support multiple languages. Unlike prior works on multilingual multi-NLP-task tuning (Mishra et al., 2022; Muennighoff et al., 2023), we focus on open-ended question answering under language generation.

Our data setting combines two low-cost practices: self-instruct, which distils data from a powerful LLM (Wang et al., 2023; Taori et al., 2023) and the idea of leveraging machine translation to create multilingual datasets (Muennighoff et al., 2023). We fine-tune several decoder LLMs with either full-parameter fine-tuning (FFT) or low-rank adaptation (LoRA, Hu et al., 2022) with different language combinations. Our experiments feature a fixed computation budget to offer practical insights. It is shown that multilingual tuning is preferred to monolingual tuning for each language under LoRA, but the results are mixed under FFT. English-tuned LLMs are not well-versed in responding in other languages, whereas a downsampled multilingual tuning scheme proposed by us is more robust. Finally, we examine our model performance on unseen languages and various LLMs of roughly the same size.

2 Methodology

2.1 Instruction data

We use the Alpaca dataset as a seed to create a multilingual instruction-response dataset. We used the cleaned version with 52K instances[1] and machine-translated it into eight languages: Bulgarian, Czech, Chinese, German, Finnish, French, Russian, and Spanish, using open-source translation systems.[2]

[*]Equal contribution. Our code, training data, and test data will be at https://github.com/hplt-project/monolingual-multilingual-instruction-tuning.

[1]https://github.com/gururise/alpacadatacleaned

[2]https://github.com/browsermt/bergamot-translator

Findings of the Association for Computational Linguistics: EACL 2024, pages 1347–1356
March 17-22, 2024

2.2 Budget-controlled instruction tuning

For monolingual tuning, we tune LLMs for each language separately, whereas for multilingual tuning, we merge and shuffle the data in all languages. This allows for resource-controlled comparisons between monolingual and multilingual tuning, where a fixed (and equal for each language) computation budget is allocated to support all languages of interest. Experimental resource usage is described as follows:

1) Let C_{Alpaca} denote the cost of *monolingual* Alpaca fine-tuning for a single language, then it costs $N \times C_{Alpaca}$ to tune individual models to support N languages.
2) *Multilingual* instruction tuning will cost $N \times C_{Alpaca}$ too, as it trains on data available in all N languages in one go.

We can fairly compare LLMs trained via 1) and 2) for any language. In addition, we propose to benchmark two budget-saving options which cost the same C_{Alpaca} as a monolingual Alpaca:

3) As a simple baseline, we use an *English-tuned* model to respond to all languages.
4) *Downsampled multilingual*: we randomly sample from the multilingual data in 2) to have the size of a monolingual dataset.

Our study covers two training paradigms: *low-rank adaptation* and *full-parameter fine-tuning*. Both fine-tune an LLM with the causal language modelling objective on the instruction-response data, with hyperparameters listed in Appendix A.1. Five LLMs are involved: Baichuan-2, BLOOM, LLaMA, OpenLLaMA, and Pythia, aiming to test with different language coverage in the base LLMs. Pythia, LLaMA, and OpenLLaMA are predominantly English, while Baichuan-2 and BLOOM are more versatile. A detailed description of the LLMs is in Appendix A.2.

2.3 Evaluation setup

Test data Our instruction-tuned LLMs are benchmarked on languages both *seen* and *unseen* during fine-tuning. We employ native speakers to manually translate 50 prompts sampled from OpenAssistant (Köpf et al., 2023) into eight languages: six seen during training and two unseen. The seen category includes English, French, Spanish, Bulgarian, Russian, and Chinese. Among the six, English is the highest-resourced, followed by French and Spanish which share the same script as English. Bulgarian and Russian are European languages but use a writing system distinct from English. Finally, Chinese is a high-resource distant language in a different script. For unseen tests, we pick Bengali and Norwegian. Bengali is distant from the above languages and uses a different script, whereas Norwegian is under-resourced but overlaps with English writing script to some extent.

LLM-as-a-judge To avoid expensive evaluation costs, we adopt LLM-as-a-judge (Zheng et al., 2023) to assign a score (1 to 3) to each instruction-response pair, and the final model score is the sum of its scores across all test instances. We use GPT-3.5 (`gpt-3.5-turbo-0613`) as the judge; it is queried with an instruction-response pair each time without model information or request history. We make modifications to Zheng et al. (2023)'s prompt to ask the LLM to consider that an answer should be in the same language as the question, which is often the expectation with AI assistants.[3] The exact wording is as Appendix B.1 Figure 6.

Language (in)consistency Our manual inspection suggests that GPT-3.5 does not always obey the language requirement imposed. An example in Appendix B.2 Table 2 shows a response in another language but scored highly. Hence, we run language identification and force-set a score to 0 if the response language is different from the query. We use the `fastText` framework (Joulin et al., 2017) with Burchell et al. (2023)'s checkpoint. The final response score can be framed as a product of GPT's quality score and a binary language identification outcome: $score = eval\_score \times lang\_id$. The aggregated test score thus ranges from 0 to 150.

Human-LLM agreement We pick 600 outputs from 12 models to cover multilingual and monolingual systems and invite human evaluators to score each sample with an instruction similar to the LLM-as-a-judge prompt as in Appendix B.3. Four languages—English, Spanish, Bulgarian, and Chinese—are human-evaluated, and we obtain very high system-level Pearson correlation coefficients of 0.9225, 0.9683, 0.9205, and 0.8685, respectively between GPT-3.5 and human. Details are in Table 3 in the appendix. This indicates the reliability of using LLM-as-a-judge to draw meaningful findings.

[3] There could be exceptions like text translation and code generation (Shaham et al., 2024).

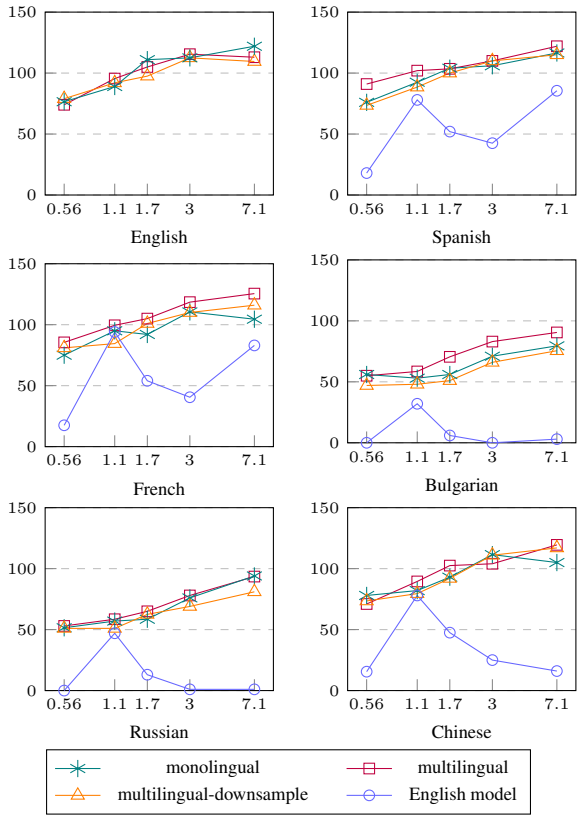

Figure 1: **LoRA** with **BLOOM** at different sizes. Caption: language; y-axis: score; x-axis: model size (B).

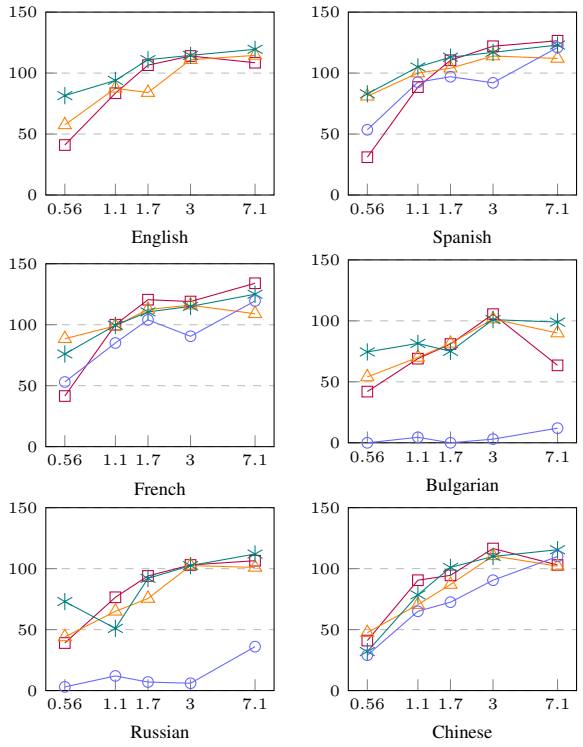

Figure 2: **FFT** with **BLOOM** at different sizes. Caption: language; y-axis: score; x-axis: model size (B). Same legend as Figure 1.

3 Performance and Discussions

3.1 Model sizes

Results from LoRA fine-tuning of BLOOM at different sizes are shown in Figure 1. At smaller sizes, multilingual (—□—) and monolingual (—✳—) instruction tuning attain similar performance, and at larger sizes, multilingual models are generally better except for English. We observe similar trends for Pythia, placed in Appendix C.1 Figure 8 due to space constraints. Moving on to full-parameter fine-tuning of BLOOM in Figure 2, we discover that at relatively small (<1.7B) or large sizes (7B), monolingual models are generally better than multilingual models for individual languages. These observations suggest that multilingualism works well with LoRA, but separate monolingual tuning might be better with FFT. Overall, the LLMs' performance is correlated with sizes regardless of the tuning technique as anticipated.

3.2 Budget-efficient tuning

To aid our exploration of resource-constrained instruction tuning, in the aforementioned Figures 1, 2, and 8 (in appendix C.1), we add the plots of two budget data conditions: using English-tuned models to respond to instructions in other languages (—○—), as well as instruction tuning with downsampled multilingual data (—△—).

When using a single English model for all languages, its efficacy depends on the intended language/script's closeness to English: Spanish and French can maintain reasonable scores, but Bulgarian, Russian, and Chinese record very low performance. The only exception is BLOOM FFT in Figure 2, where the model is not too behind when operating in Chinese. Interestingly, BLOOM with LoRA sees a performance spike at 1.1B for non-English. At this specific size, it displayed multilingual transferability from pre-training and learned to follow multilingual instructions despite being fine-tuned merely in English.

In contrast, while consuming the same computational resources, downsampled multilingual tuning is significantly more robust across all test languages. These models sometimes achieve on-par performance with monolingual tuning in individual languages. This means that to support several languages with limited resources, the best practice is to train on small multilingual data even created with machine translation instead of full English data. Nonetheless, if the budget permits, training with the full multilingual data is still slightly better.

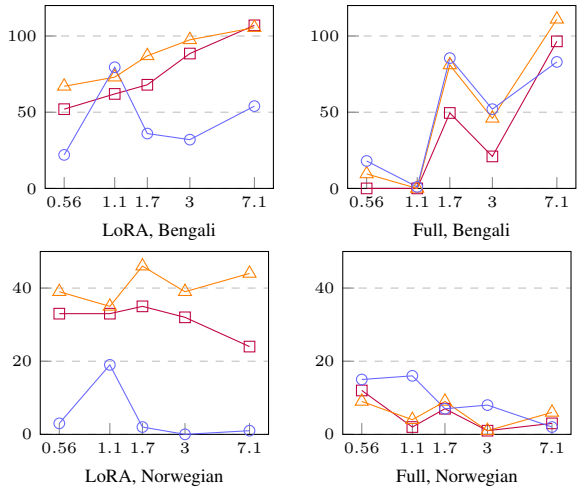

Figure 3: **LoRA** and **FFT** with **BLOOM** at different sizes and tested on **unseen** languages. Caption: training method and language; y-axis: score; x-axis: model size (B).

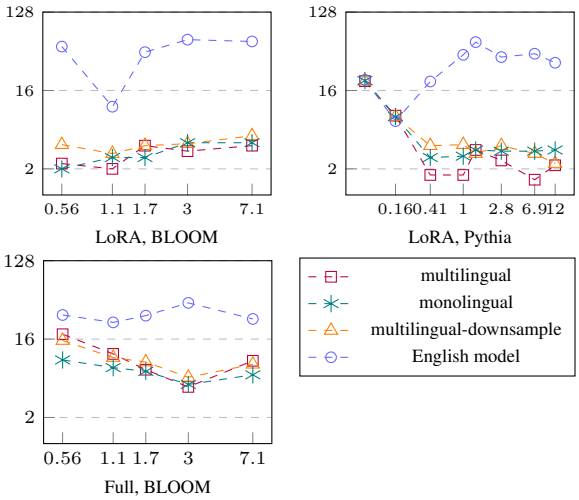

Figure 4: Evaluation **score change** before and after language identification, **averaged** over six seen test languages, at different LLM sizes. Caption: training method and base model; y-axis: score difference (log scale); x-axis: model size (B).

3.3 Unseen languages

Further in Figure 3, we look at BLOOM models which underwent LoRA or FFT but were subsequently instructed in unseen languages at test time. English-tuned LLMs behave distinctly with LoRA and FFT. With the former, they are nowhere near multilingual tuned models, but with the latter, we see close or even better results. It might imply that FFT can even lift performance for languages not present in the instruction data. However, FFT results on Norwegian could be an outlier given its comparably low scores. Considering multilingual instruction tuning, we notice a pattern opposed to that on languages seen during training—learning on the downsampled data is superior to ingesting the full mixed data. We conclude that it is important to not overfit to instruction languages if unseen languages are expected in downstream tasks.

3.4 Language robustness

We review each model and data recipe's scores before and after adding language identification, to isolate an LLM's language robustness from its "inherent quality" (regardless of the response language). We compute the *differences* in GPT evaluation scores before and after applying language identification. A (big) difference suggests that a model produces reasonable answers in an undesired language. In Figure 4, we report the *average* of the score differences across all six test languages seen during tuning. English-only models are the least robust—their score differences are way above other techniques. With LoRA, full multilingual tuning records the smallest performance drop; with FFT, monolingual tuning is preferred. The insights from language robustness are corroborated by our early findings in Section 3.1: superior results are obtained when using multilingual tuning with LoRA and monolingual tuning with full-parameter tuning. Nonetheless, monolingual and multilingual tuning are not too far apart; specifically for BLOOM with LoRA, language robustness does not improve as the model gets larger.

3.5 Model families

Finally, we experiment with base LLMs from different families of around 7 billion parameters. In Figure 5, we plot the evaluation scores for multilingual, downsampled multilingual, and monolingual LoRA tuning for six languages. Generally, LLaMA and OpenLLaMA have better performance than BLOOM and Pythia potentially because they have pre-training data that is an order of magnitude larger. Also Bulgarian, Russian, and Chinese see lower scores than English, again presumably due to the language distribution in the pre-training data.

Delving into the comparison between monolingual and multilingual instruction tuning, we find that out of 30 cases across six languages and five LLMs, monolingual tuning is ahead in just two cases: LLaMA tested in Russian and Chinese. The cost-efficient downsampled multilingual tuning leads in four cases: two in French and two in Russian. In other situations, multilingual training is on par if not better. The outcome of tuning several similar-sized LLMs confirms that multilingual tuning is favourable using LoRA.

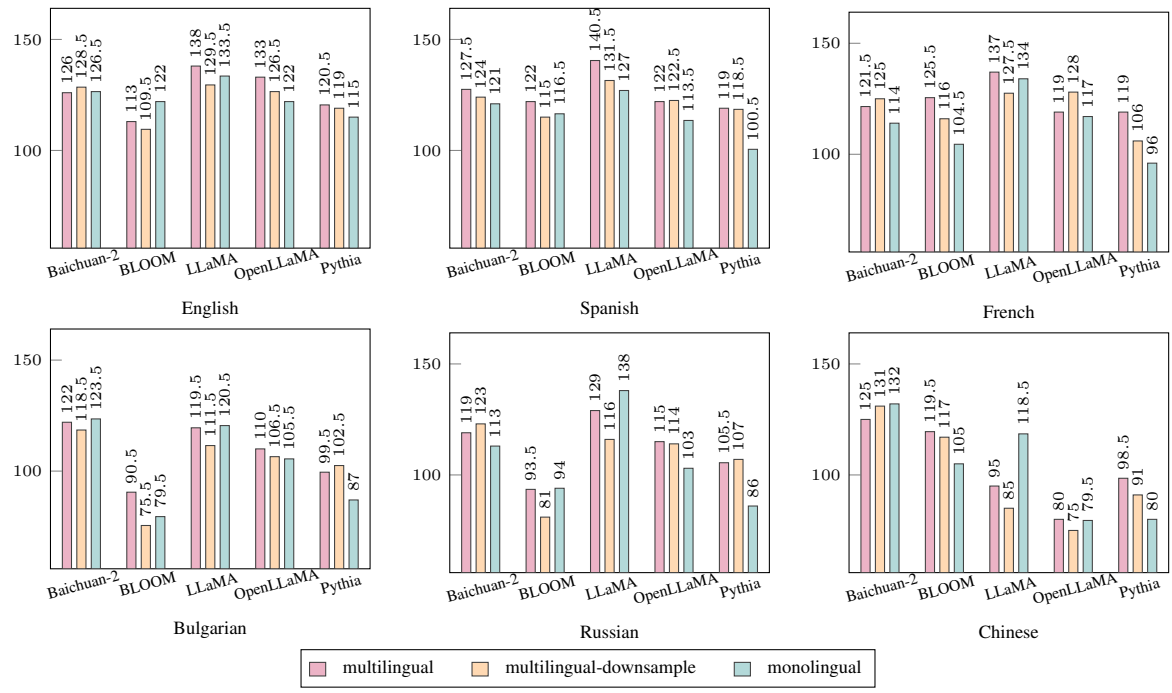

Figure 5: LoRA fine-tuning on different 7B LLMs. Caption: language generated; y-axis: score; x-axis: model family.

4 Related Work

Many large language models appeared recently: the closed-source GPT model family (Radford et al., 2019; Brown et al., 2020; Ouyang et al., 2022); open-source English-centric models like LLaMA (Touvron et al., 2023), OpenLLaMA (Geng and Liu, 2023), and Pythia (Biderman et al., 2023); open-source multilingual models like mT5 (Xue et al., 2021) and BLOOM (Scao et al., 2022). These models have exhibited different degrees of language versatility.

LLM pre-training data is usually skewed towards English. One way to improve an LLM's coverage of non-English languages is through continued pre-training (Cui et al., 2023, inter alia). Another rich body of literature looks into multilingualism in instruction tuning, which is used to adjust base models to respond to input (Mishra et al., 2022; Sanh et al., 2022; Wei et al., 2022; Longpre et al., 2023). It trains an LLM by providing downstream tasks' input and output in a specific format. Early research created a multilingual instruction dataset using machine translation and showed that multilingual tuning gained higher performance than English-only fine-tuning (Muennighoff et al., 2023). They also found that low-cost translated instructions are superior to human-written non-English prompts on multiple language understanding tasks.

Lately, multiple contemporaneous papers delving into multilingual instruction tuning have been made public on arXiv—some appeared before our work and some after. This reflects the importance and interest in widening LLMs' language support. Li et al. (2023a) created an instruction dataset with instructions translated from English but responses generated by an LLM. When tuned with LoRA, their monolingual models outperform multilingual ones on language understanding tasks. Wei et al. (2023) created a multilingual counterpart of Alpaca using self-instruct. It has also been showcased that translation instructions improve cross-lingual capabilities (Li et al., 2023b; Zhang et al., 2023; Ranaldi et al., 2023) and research explored more cross-lingual task data and multilingual tuning (Zhu et al., 2023). Moreover, researchers have unveiled that fine-tuning on a modest number of languages—approximately three—seems to effectively instigate cross-lingual transfer in downstream tasks (Kew et al., 2023; Shaham et al., 2024).

5 Conclusion

This paper presents a study of instruction tuning of large language models in different language contexts. Our study in a resource-controlled setting suggests that multilingual tuning offers more benefits compared to monolingual tuning. We find that multilingual tuning on a downsampled dataset achieves better robustness on unseen languages.

Limitations

The LLMs we studied have primarily 7B and at most 13B parameters and the multilingual training only spanned nine languages. Scaling to larger models and more languages would be interesting. The best checkpoint for our instruction fine-tuning is selected based on validation cross-entropy, but there is no guarantee that this leads to the best performance on the downstream task.

To manage the budget for human translation and evaluation, we consider eight languages (six seen and two unseen languages during instruction tuning) to translate and sample 50 instances for evaluation. The training data for non-English languages are obtained via machine translation, which introduces errors, affects response fluency, and might alter the nature of some tasks such as grammatical error correction and code generation.

Ethics Statement

The dataset we translated and generated does not contain private or sensitive information. Similar to other research on large language models, there is no definitive way for us to prevent the instruction-tuned models from generating inappropriate content. However, we see minimal such risks associated with our project, as neither our models nor generated contents are intended for public consumption. Human evaluators did not report inappropriate content generated by the models.

Acknowledgements

This paper stemmed from a hackathon project organized by the High Performance Language Technologies (HPLT) consortium.[4] We are grateful to Alicia Núñez Alcover, David Samuel, Joona Kytöniemi, Jörg Tiedemann, Lucas Charpentier, Petter Mæhlum, Sampo Pyysalo, Sunit Bhattacharya, and Zhicheng Guo for project discussions, test data translation, and evaluation setup.

The work has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No 101070350, from UK Research and Innovation (UKRI) under the UK government's Horizon Europe funding guarantee [grant number 10052546], as well as from the European Research Council (ERC) under the EU's Horizon 2020 research and innovation program (agreement № 771113).

Computation in this work was performed on LUMI, Karolina, and Baskerville. We acknowledge CSC-IT Center for Science, Finland for awarding this project access to the LUMI supercomputer, owned by the EuroHPC Joint Undertaking, hosted by CSC (Finland) and the LUMI consortium through Finnish extreme scale call (project LumiNMT). Karolina was supported by the Ministry of Education, Youth and Sports of the Czech Republic through the e-INFRA CZ (ID:90254). The Baskerville Tier 2 HPC 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.

[4] https://hplt-project.org

References

Mikel Artetxe and Holger Schwenk. 2019. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the Association for Computational Linguistics*, 7:597–610.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*.

Laurie Burchell, Alexandra Birch, Nikolay Bogoychev, and Kenneth Heafield. 2023. An open dataset and model for language identification. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*.

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*.

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

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)*.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. 2020. The pile: An 800GB dataset of diverse text for language modeling. *arXiv preprint*.

Xinyang Geng and Hao Liu. 2023. OpenLLaMA: An open reproduction of LLaMA. GitHub repository.

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*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*.

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. *arXiv preprint*.

Hugo Laurençon, Lucile Saulnier, Thomas Wang, Christopher Akiki, Albert Villanova del Moral, Teven Le Scao, Leandro Von Werra, Chenghao Mou, Eduardo González Ponferrada, Huu Nguyen, et al. 2022. The bigscience ROOTS corpus: A 1.6TB composite multilingual dataset. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

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*.

Jiahuan Li, Hao Zhou, Shujian Huang, Shanbo Chen, and Jiajun Chen. 2023b. Eliciting the translation ability of large language models via multilingual finetuning with translation instructions. *arXiv preprint*.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, et al. 2023. The Flan collection: Designing data and methods for effective instruction tuning. In *Proceedings of the 40th International Conference on Machine Learning*.

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 (Volume 1: Long Papers)*.

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 (Volume 1: Long Papers)*.

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*.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*.

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

Leonardo Ranaldi, Giulia Pucci, and Andre Freitas. 2023. Empowering cross-lingual abilities of instruction-tuned large language models by translation-following demonstrations. *arXiv preprint*.

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

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. 2022. BLOOM: A 176B-parameter open-access multilingual language model. *arXiv preprint*.

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*.

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 repository.

Together Computer. 2023. RedPajama: An open source recipe to reproduce LLaMA training dataset. GitHub repository.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. LLaMA: Open and efficient foundation language models. *arXiv preprint*.

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)*.

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 *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*.

Shijie Wu and Mark Dredze. 2020. Are all languages created equal in multilingual BERT? In *Proceedings of the 5th Workshop on Representation Learning for NLP*.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Aiyuan Yang, Bin Xiao, Bingning Wang, Borong Zhang, Chao Yin, Chenxu Lv, Da Pan, Dian Wang, Dong Yan, Fan Yang, et al. 2023. Baichuan 2: Open large-scale language models. *arXiv preprint*.

Jiacheng Ye, Xijia Tao, and Lingpeng Kong. 2023. Language versatilists vs. specialists: An empirical revisiting on multilingual transfer ability. *arXiv preprint*.

Shaolei Zhang, Qingkai Fang, Zhuocheng Zhang, Zhengrui Ma, Yan Zhou, Langlin Huang, Mengyu Bu, Shangtong Gui, Yunji Chen, Xilin Chen, et al. 2023. BayLing: Bridging cross-lingual alignment and instruction following through interactive translation for large language models. *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. 2023. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Wenhao Zhu, Yunzhe Lv, Qingxiu Dong, Fei Yuan, Jingjing Xu, Shujian Huang, Lingpeng Kong, Jiajun Chen, and Lei Li. 2023. Extrapolating large language models to non-english by aligning languages. *arXiv preprint*.

A Experimental Setup Details

A.1 Hyperparameters

Table 1 shows the hyperparameter configurations of LoRA and full-parameter fine-tuning. LoRA is a parameter-efficient training method where, for a big matrix, only low-rank matrices are trained and patched to it. In our case, we apply it to the attention matrices (key, query, value) and use rank 8, dropout 0.05, and scaling factor 16 throughout. We use a batch size of 128, set a fixed training budget of 5 epochs with a learning rate of $3e^{-4}$, and select the best checkpoint based on validation cross-entropy. For full-parameter fine-tuning, we follow the configurations of Alpaca by training for 3 epochs with a learning rate of $2e^{-5}$, a warm-up ratio of 0.03, and a batch size of 256.

Since we use a range of models of different sizes, we estimate computation time based on 7-billion parameter models which are the second largest we fine-tuned. LoRA tuning takes 15-20 hours on 4 GeForce RTX 3090 GPUs, using CPU memory offloading and distributed training. Full-parameter fine-tuning is performed on 4 AMD MI250x GPUs (treated as 8 GPUs with 64G memory each at runtime) with model parallelism, and it requires around 24 hours to finish. Given the high computational cost of model fine-tuning, we conducted all fine-tuning experiments once. We use a range of different GPUs, but through gradient accumulation, we maintain the same global batch size for each tuning technique: 128 for LoRA and 256 for full-parameter fine-tuning.

A.2 Description of LLMs

Due to the space constraint, we place a detailed description of LLMs used in our research here. All the models used in this study are publicly available and free to use for academic purposes.

Baichuan-2 (Yang et al., 2023) is a multilingual LLM trained on 2.6 trillion tokens. While the data composition is not transparent in its technical report, the LLM weights are open-source and it

Method	Hyperparameter	Value
LoRA	LoRA modules	query, key, value
	rank	8
	scaling factor	16
	dropout	0.05
	learning rate	$3e^{-4}$
	global batch size	128
	epochs	5
FFT	learning rate	$2e^{-5}$
	global batch size	256
	epochs	3

Table 1: Hyperparameter configurations of LoRA and full-parameter fine-tuning

performs strongly on tasks in English and Chinese. We use its 7B checkpoint.

BLOOM (Scao et al., 2022) is trained on the ROOTS dataset (Laurençon et al., 2022) containing 350 billion tokens in 46 natural languages spanning 9 language families and 12 programming languages. The LLM has English, Chinese, French, and Spanish as the major components. We use the checkpoints from 560M to 7.1B for experiments.

LLaMA (Touvron et al., 2023) has been trained on data mainly in English with some in European languages in Latin and Cyrillic scripts. It could also support other languages with byte-BPE tokenization. We use its 7B model which has seen 1 trillion tokens.

OpenLLaMA (Geng and Liu, 2023) is an open-source reproduction of LLaMA, trained on the RedPajama dataset (Together Computer, 2023), which is close to LLaMA's data composition. Similarly, we use the 7B version.

Pythia (Biderman et al., 2023) is trained on the Pile (Gao et al., 2020) which has nearly 300 billion tokens and is intended to be English-dominant. We experiment with the full range from 70M to 12B.

B Evaluation Details

B.1 Prompt template for LLM-as-a-judge

Our LLM-as-a-judge prompt template is as Figure 6, similar to Zheng et al. (2023)'s with an additional requirement on the response language.

B.2 An example of language inconsistency

Table 2 lists an example where the response is "correct" in content but does not match the language of the query. In this case, the judge, GPT-3.5 does not penalize this language inconsistency despite being asked to do so in the evaluation prompt. This makes extra language identification essential.

```
Please act as an impartial judge and evaluate
the quality of the response provided by an AI
assistant to the user instruction displayed
below. Your evaluation should consider factors
such as helpfulness, relevance, accuracy, depth,
creativity, and level of detail. It is also
required that the response is in the same language
as the instruction. Begin your evaluation with a
short explanation. Be as objective as possible.
After providing your explanation, please rate
the response on a scale of 1 to 3 by strictly
following this format:"[[rating]]", for example:
"Rating: [[2]]".

[User instruction]
${instruction}

[AI assistant's response]
${response}
```

Figure 6: Prompt template for requesting a response evaluation from GPT-3.5, where `${instruction}` and `${response}` are replaced with questions and model responses.

Instruction: 你好！你能帮我计划婚礼吗? *(English: Hello! Can you help me plan my wedding?)*
Response: Sure, I would be happy to help you plan your wedding. Can you tell me a little bit more about your wedding plans and what you are looking for in a wedding planner?
GPT-3.5 explanation: The AI assistant's response shows a good understanding of the user's request and offers assistance. It asks the user for more information about their wedding plans, which shows that it is trying to gather relevant details to provide appropriate help. The response is polite and friendly, which contributes to a positive user experience.
GPT-3.5 score: 3

Table 2: An example of instruction-response language inconsistency, where the question is in Chinese but the response is in English, which GPT-3.5 fails to account for in its evaluation.

B.3 Human evaluation and human-LLM agreement

We invited human evaluators who are fluent or native in the language of the instructions and responses to score in total outputs from 12 models fine-tuned with LoRA. We attach the instruction given to human evaluators in Figure 7. The systems' responses for the same instruction are shuffled but grouped together to provide a context of the overall quality. The human evaluators are asked to assign each response a score. We list the model details, as well as their aggregated GPT and human evaluation scores in Table 3.

	LLM	Size (B)	English		Spanish		Bulgarian		Chinese	
			GPT-3.5	human	GPT-3.5	human	GPT-3.5	human	GPT-3.5	human
Multi-lingual	BLOOM	1.1	95.5	93.0	102.0	98.0	58.5	54.5	89.5	97.5
	BLOOM	3	115.5	105.0	110.0	103.5	83.0	59.0	104.0	102.0
	BLOOM	7.1	113.0	119.5	122.0	116.5	90.5	67.0	119.5	117.5
	LLaMA	7	138.0	131.5	140.5	123.0	119.5	112.0	95.0	89.0
	OpenLLaMA	7	133.0	130.0	122.0	112.5	110.0	89.0	80.0	67.5
	Pythia	6.9	120.5	117.0	119.0	107.5	99.5	75.0	98.5	87.5
Mono-lingual	BLOOM	1.1	89.0	81.0	92.5	86.0	53.0	49.0	82.0	75.5
	BLOOM	3	112.5	103.5	106.0	99.5	71.0	64.0	111.5	96.0
	BLOOM	7.1	122.0	111.5	116.5	111.5	79.5	73.5	105.0	106.0
	LLaMA	7	133.5	121.0	127.0	115.0	120.5	117.5	118.5	96.5
	OpenLLaMA	7	122.0	124.0	113.5	108.0	105.5	87.0	79.5	66.5
	Pythia	6.9	115.0	116.0	100.5	97.5	87.0	72.5	80.0	72.0
Pearson correlation coefficient			**0.9225**		**0.9683**		**0.9205**		**0.8685**	

Table 3: Human evaluation scores and their system-level correlation with GPT-3.5 scores. Models are fine-tuned with LoRA.

```
Please evaluate the quality of the responses provided by AI assistants to the questions in your respective tab. Most questions are open-ended, meaning there is no strictly correct or best answer. Please make a judgment based on your perspective of quality. You could consider factors such as helpfulness, relevance, accuracy, depth, creativity, and level of detail. It is also required that the response is in the same language as the question unless otherwise specified by the instruction itself. Please rate the response on a scale of 0 to 3. If you feel indecisive, you can use an increment of 0.5. You can give a score of 0 for "incorrect language, not readable, content cannot be understood"; give a score of 1 for "a relatively bad response"; give a score of 2 for "a medium response"; give a score of 3 for "a relatively good response".
```

Figure 7: Instructions for human evaluators.

C Result Details

C.1 Experiments on Pythia with LoRA

Apart from LoRA fine-tuning on BLOOM models, we conduct the same investigation on Pythia models at different sizes. We observe that multilingual tuning does not lose to monolingual tuning in any language, similar to what we find about BLOOM in Section 3.1. The plots for the six languages are included as Figure 8.

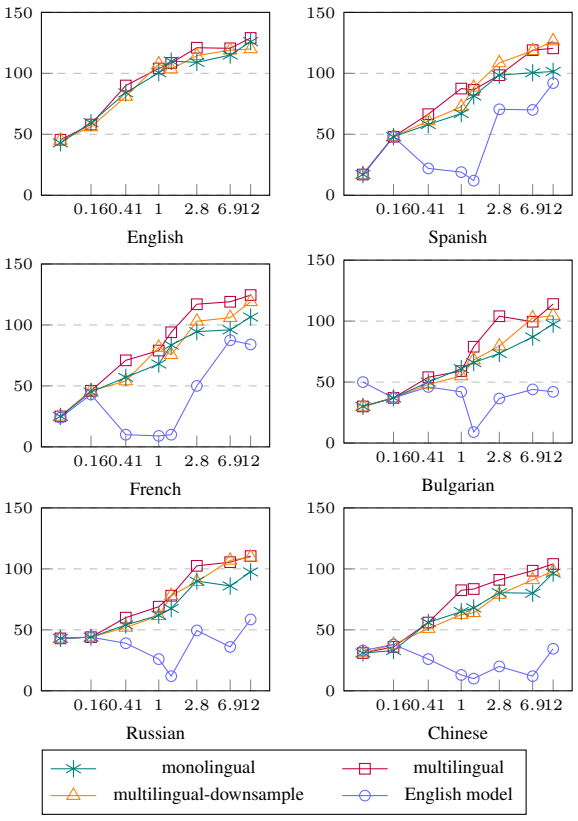

Figure 8: **LoRA** fine-tuning on **Pythia**. Caption: language generated; y-axis: score; x-axis: model size (B) on a logarithmic scale.