

Monolingual or Multilingual Instruction Tuning: Which Makes a Better Alpaca

Pinzhen Chen^{1,*} Shaoxiong Ji^{2,*} Nikolay Bogoychev¹
Barry Haddow¹ Kenneth Heafield¹

¹University of Edinburgh ²University of Helsinki

pinzhen.chen@ed.ac.uk, shaoxiong.ji@helsinki.fi, n.bogoych@ed.ac.uk

Abstract

Foundational large language models (LLMs) can be instruction-tuned to develop open-ended question-answering capability, facilitating applications such as the creation of AI assistants. While such efforts are often carried out in a single language, building on prior research, we empirically analyze cost-efficient approaches of monolingual and multilingual tuning, shedding light on the efficacy of LLMs in responding to queries across monolingual and multilingual contexts. Our study employs the Alpaca dataset and machine translations of it to form multilingual training data, which is then used to tune LLMs through low-rank adaptation and full-parameter training. Comparisons reveal that multilingual tuning is not crucial for an LLM’s English performance, but is key to its robustness in a multilingual environment. With a fixed budget, a multilingual instruction-tuned model, merely trained on downsampled data, can be as powerful as training monolingual models for each language. Our findings serve as a guide for expanding language support through instruction tuning with constrained computational resources.

1 Introduction

In the realm of natural language processing with large language models (LLMs), the language capacity of pre-trained models has attracted much research attention (Conneau et al., 2020). Intuitive language acquisition might favour learning a single language, exemplified by monolingual language models such as BERT (Devlin et al., 2019) and Pythia (Biderman et al., 2023). On the other hand, multilingual models, e.g. BLOOM (Scao et al., 2022), are pre-trained with texts in many languages, which seem attractive when multilingual capabilities are of interest in the downstream tasks, due to lower operational costs. Ye et al. (2023a) studied language versatility and specialization and revealed

that English-centric models such as LLaMA (Touvron et al., 2023) possess good multilingual transfer ability in natural language inference and reasoning tasks. While base LLMs only produce a completion of the input, recently proposed instruction tuning (Sanh et al., 2022; Wei et al., 2022a) is able to instate the open-ended question-answering capability in LLMs, i.e., generating responses aligned with human intention and solving tasks (Mishra et al., 2022; Muennighoff et al., 2022; Taori et al., 2023; Ye et al., 2023b).

Building on research that turns a base LLM into a chat model in an inexpensive way (Taori et al., 2023), this work compares monolingual and multilingual instruction fine-tuning, in order to explore cost-efficient strategies to adapt base LLMs to a multilingual chat environment. Specifically, we evaluate LLMs instruction-tuned on different data combinations and tested using open-domain question answering in monolingual and multilingual settings. Our methodology combines two low-cost practices: (1) the self-instruct paradigm which distills instruction-response data from a powerful LLM (Wang et al., 2023a) and (2) the idea of leveraging machine translation to create multilingual datasets (Muennighoff et al., 2022).

The goal of this work is to offer practical insights into monolingual and multilingual instruction tuning. Both full-parameter fine-tuning and low-rank adaptation (Hu et al., 2022) are experimented with. We study the result patterns from models having different sizes and compare monolingual tuning, multilingual tuning and the language transfer ability of English-tuned LLMs. We also propose a budget-aware multilingual training scheme that is demonstrated to be more robust. Finally, we examine our conclusions by generalizing to unseen languages and to LLMs from several families of roughly the same size.

*Equal contribution.

2 Methodology

2.1 Instruction tuning

Instruction tuning refers to continually training an LLM with NLP tasks formatted as natural language instructions and model responses treated as task outputs (Wei et al., 2022a). Further to solving structured natural language tasks, Taori et al. (2023) turned an English-centric LLM, LLaMA, into an open-ended chat model. It delivers GPT-like performance for English by training on distilled data from GPT itself (Wang et al., 2023b). Soon after, by translating the Alpaca dataset into other languages, counterparts emerged for more languages such as Chinese (Cui et al., 2023), German, and Portuguese.^{1,2} Those models achieve reasonable performance in the target language, with the machine-translated data acting as a source of both instruction tuning and language adaptation, enabling LLaMA to produce fluent non-English text. Our work uses the same seed data to exploit multilingual opportunities and cross-lingual properties with LLMs.

2.2 Language speciality and versatility

We interpret language speciality as an LLM’s concentration on yielding high-quality responses in a single language, and versatility as its ability to respond in multiple languages. These abilities could be affected by both its pre-training data and the instruction tuning data. Essentially, Alpaca attempts in the previous section opted for speciality, but we are interested in understanding the trade-off between the two. While a multilingual base LLM like BLOOM seems to be more versatile, models trained predominantly in English, such as LLaMA and Pythia, can also acquire and demonstrate knowledge of other languages, likely due to incidental dataset contamination (Blevins and Zettlemoyer, 2022; Briakou et al., 2023).

Previous work on multilingual instruction tuning through translation taps advances an LLM’s performance in various NLP tasks (Muennighoff et al., 2022). This work takes the same machine translation approach to create the instruction data. We take a further step to compare the quality of open-ended responses in the same language, but from LLMs tuned in different scenarios under controlled settings: English-only, individual languages, and a mix of languages.

¹<https://github.com/avocardio/Zicklein>

²<https://github.com/22-hours/cabrita>

2.3 Budget-constrained training

Since fine-tuning LLMs is expensive and technically unbounded given unlimited resources, we choose to constrain the computational budget in separate experiments to make data recipes comparable. Our experimental conditions are as follows:

- (1) Let C_{alpaca} denote the cost of monolingual Alpaca fine-tuning for a single language, then it will cost $X \times C_{alpaca}$ to tune individual models to support X languages of interest.
- (2) Multilingual instruction-tuning costs $X \times C_{alpaca}$ too, as it uses data available in all languages.

We are able to fairly compare the performance of an LLM tested on any language trained via (1) and (2). In addition, we propose to benchmark two practical budget-saving options:

- (3) As a naive baseline, we use an English-tuned model to respond to other languages. It has the same cost C_{alpaca} as (1).
- (4) Downsampled multilingual: we downsample the multilingual dataset in (2) to the size of a single monolingual dataset, with training cost C_{alpaca} as (1) too.

2.4 Base models

We conduct instruction fine-tuning on four base LLMs, aiming to test with different levels of language coverage. As discussed, Pythia and LLaMA are predominantly English while BLOOM is more versatile. These models are as follows.

- Pythia (Biderman et al., 2023): trained on the Pile dataset (Gao et al., 2020) containing nearly 300 billion tokens after global deduplication. The data is intended to be in English only. We experiment with the full range from 70M to 12B.
- LLaMA (Touvron et al., 2023): an LLM trained on 1.4 trillion tokens, mainly in English with some in European languages in Latin and Cyrillic scripts. It could also support other languages having byte-BPE tokenization. We report its 7B model’s performance.
- OpenLLaMA (Geng and Liu, 2023): an open-source reproduction of LLaMA, trained on the RedPajama dataset (Together Computer, 2023), which is similar to LLaMA’s data composition. Similarly, we use the 7B version.

- BLOOM (Scao et al., 2022): trained on the ROOTS dataset (Laurençon et al., 2022), which has 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.

3 Experimental Setup

3.1 Training Data

We utilize the 52K Alpaca dataset generated using OpenAI’s text-davinci-003 as our seed English data (Taori et al., 2023; Wang et al., 2023a). We use the publicly available cleaned version.³ The seed data is then translated by us into eight languages: Bulgarian (bg), Czech (cs), Chinese (zh), German (de), Finnish (fi), French (fr), Russian (ru), and Spanish (es), using various open-source models.⁴

For *monolingual* instruction-tuning, we tune a foundation LLM on each language data separately, whereas, for *multilingual* tuning, we combine and shuffle the data in all languages. This follows the constrained comparison between monolingual and multilingual fine-tuning discussed earlier, where a fixed computational budget is given to support all languages of interest.

As an alternative resource-limited comparison, we also investigate the effect of training on a *multilingual dataset downsampled* randomly to the size of a single language. In addition, we test a naive baseline using an *English* instruction-tuned model to decode all languages. Similarly, these two settings use equal computational resources.

3.2 Training Details

Our analysis covers two training scenarios: *low-rank adaptation* (LoRA, Hu et al., 2022) and *full-parameter fine-tuning*. Both LoRA and full-parameter fine-tuning start from the released LLM checkpoints and continually train the causal language modelling objective using the instruction dataset.

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 and use a rank of 8

³<https://huggingface.co/datasets/yahma/alpaca-cleaned>

⁴https://github.com/browsermt/bergamot-translator/tree/alpaca_translator

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
Full-parameter	epochs	5
	learning rate	$2e^{-5}$
	global batch size	256
	epochs	3

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

throughout. We set a fixed training budget of 5 epochs and select the best checkpoint based on validation cross-entropy. For full-parameter fine-tuning, we follow the strategy of Alpaca by training for 3 epochs with a warmup ratio of 0.03.

We utilize 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. Some hyperparameters differ between the two training strategies but all are kept consistent for different data conditions as detailed in Table 1.

3.3 Test Data

To evaluate our fine-tuned LLMs’ instruction-following capability and response quality, models are benchmarked on test samples in languages both seen and unseen during the instruction-tuning time. We aim to cover languages in different scripts and families so we can draw generic conclusions across base LLMs with various degrees of multilingualism, from versatile BLOOM to English-focused Pythia. We employ native speakers to manually translate 50 prompts sampled from OpenAssistant (Köpf et al., 2023) into languages of our interest.⁵

The *seen* category includes six languages: English, Spanish, French, Bulgarian, Russian, and Chinese. Among the six, English is the highest-resourced, and Spanish and French are high-resource and share the same script as English. Bulgarian and Russian are European languages, but use a writing system distinct from English; between these two, Bulgarian could be more challenging because most of the Cyrillic data on the Internet is written in Russian. Finally, Chinese is a high-resource but distant language in a totally different script.

⁵<https://github.com/LAION-AI/Open-Assistant>

For *unseen* tests, we pick Bengali and Norwegian. Norwegian is under-resourced and overlaps with English writing script to some extent, whereas Bengali perhaps appears more often in the LLMs’ pre-training data, but operates on a completely different vocabulary.

3.4 LLM Evaluation

We adopt LLM-as-a-judge (Zheng et al., 2023) to score each instruction-model response pair directly, and the final model evaluation scores are obtained by adding up a model’s total scores on all test samples. Such single response grading brings benefits two-fold: 1) we can alleviate the judge model’s position preference; 2) it requires significantly fewer evaluation requests as opposed to comparing responses in a pairwise manner. We used GPT-3.5 as the judge.⁶ It is queried with a question and a model response each time in a new session, without model information or request history.

We use a prompt template close to that designed by Zheng et al. (2023), except that, as we are dealing with multilingual scenarios, we make the LLM consider that the instruction and the response should be in the same language. Although we particularly care about the final score given by the judge, we still ask for a brief explanation, as this forms a chain-of-thought process (Wei et al., 2022b) that might improve the judge’s scoring accuracy. The exact wording is list as Figure 1, where `{instruction}` and `{response}` are replaced with questions and model responses.

Our early manual inspection of LLM-as-a-judge scoring suggests that GPT-3.5 does not always obey the same language requirement imposed in the prompt. We visualize an example in Appendix A Table 2, where the LLM response is in a different language from the query, but it is scored highly by GPT-3.5. Hence, we run language identification and force-set the score to 0 if a response is in a language different from the instruction, with the results presented to examine the language consistency of LLM responses in Section 4.4. We use the `fastText` framework (Joulin et al., 2017) with a recent checkpoint from Burchell et al. (2023). Formally, the final score of a response given a question can be represented as a product of GPT’s quality score and a binary language identification $score = eval_score \times language_id$. It ranges between 0 and 150.

```

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 1: Template for requesting a response evaluation.

4 Performance and Discussions

4.1 Model sizes

For LoRA fine-tuning of LLMs at different sizes, we observe similar trends for Pythia and BLOOM, plotted in Figure 2 and Figure 3 respectively. At smaller sizes, multilingual (—□—) and monolingual (—*—) instruction-tuning attain similar performance, and at larger sizes, multilingual models are generally better, except for English.

Moving on to full-parameter fine-tuning with BLOOM in Figure 4, we discover that at relatively small (<1.7B) or large sizes (7B), monolingual models are generally better than multilingual models for individual languages. This indicates that speciality is preferred to versatility when given ample training during instruction tuning.

These observations suggest that **multilingualism works well with LoRA, but with full-parameter, separate monolingual tuning might be better**. Overall, as anticipated, tuned LLMs’ performance is positively correlated with their sizes regardless of the tuning technique.

4.2 Budgeting

To aid our exploration on resource-constrained instruction tuning, in the above-mentioned Figures 2, 3, and 4, we add the comparison 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 (—△—).

Regarding using a single English model for all languages, we observe that its performance is af-

⁶`gpt-3.5-turbo-0613`

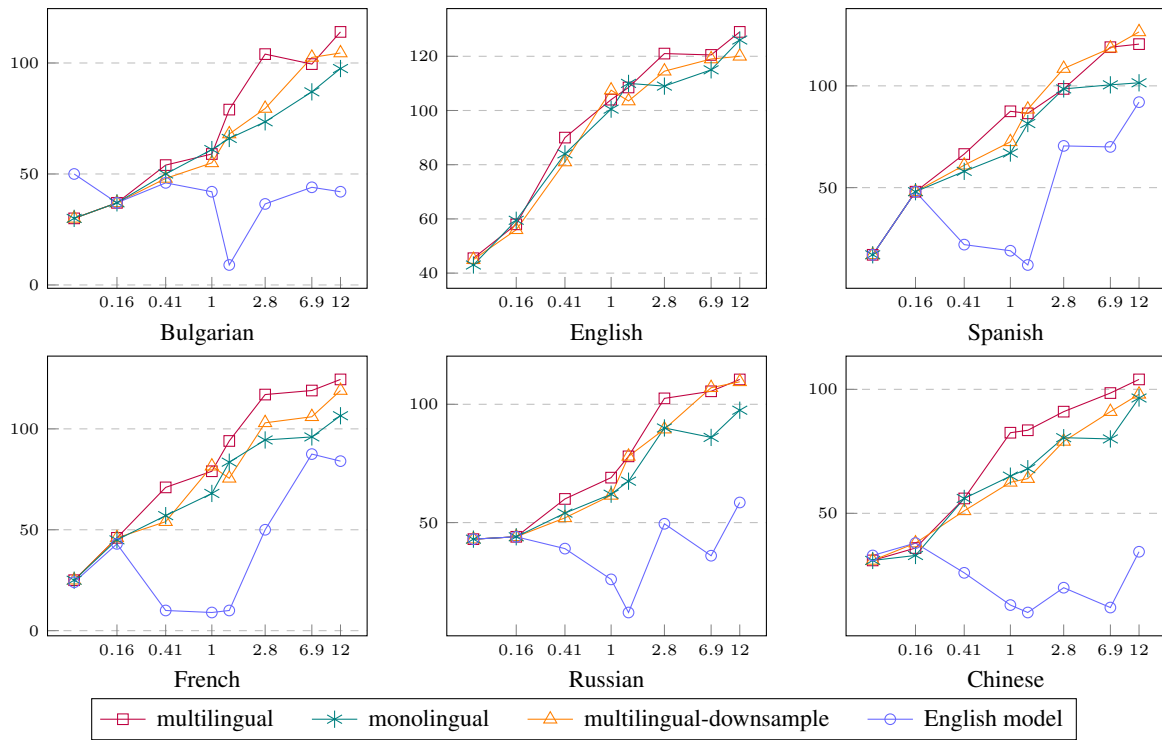


Figure 2: **LoRA** fine-tuning on **Pythia** at different sizes. Caption: language generated; y-axis: evaluation score; x-axis: model size (B) on a logarithmic scale.

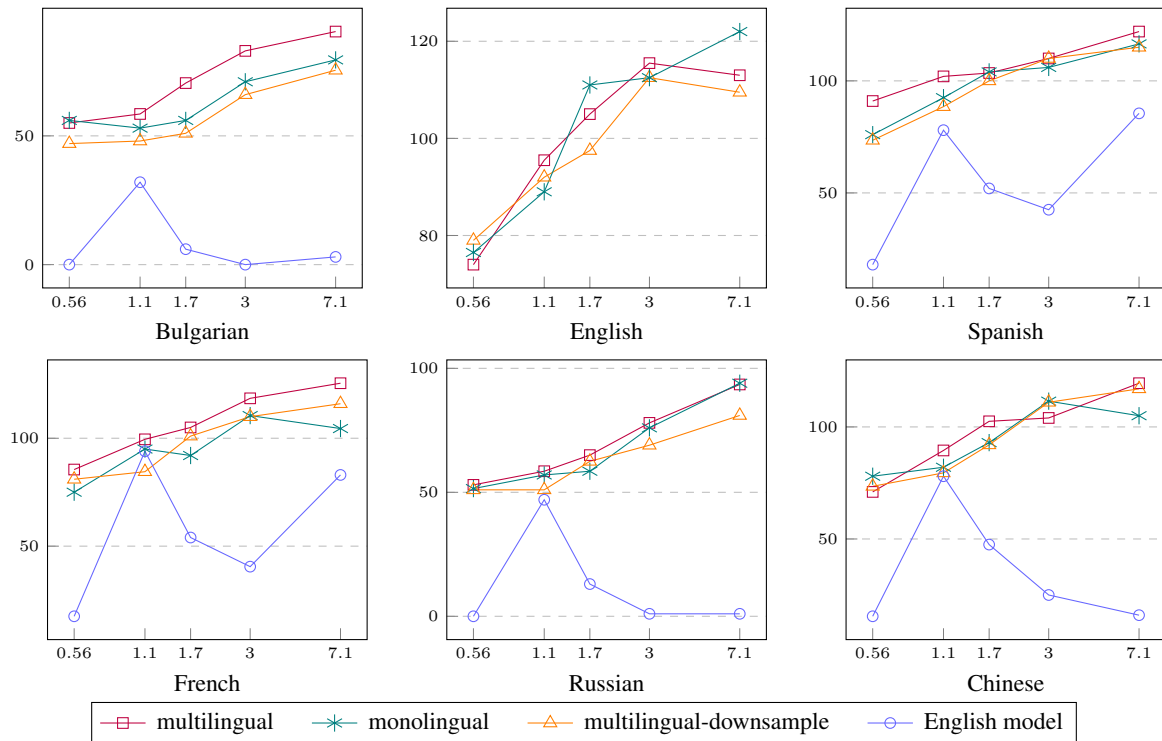


Figure 3: **LoRA** fine-tuning on **BLOOM** at different sizes. Caption: language generated; y-axis: evaluation score; x-axis: model size (B) on a logarithmic scale.

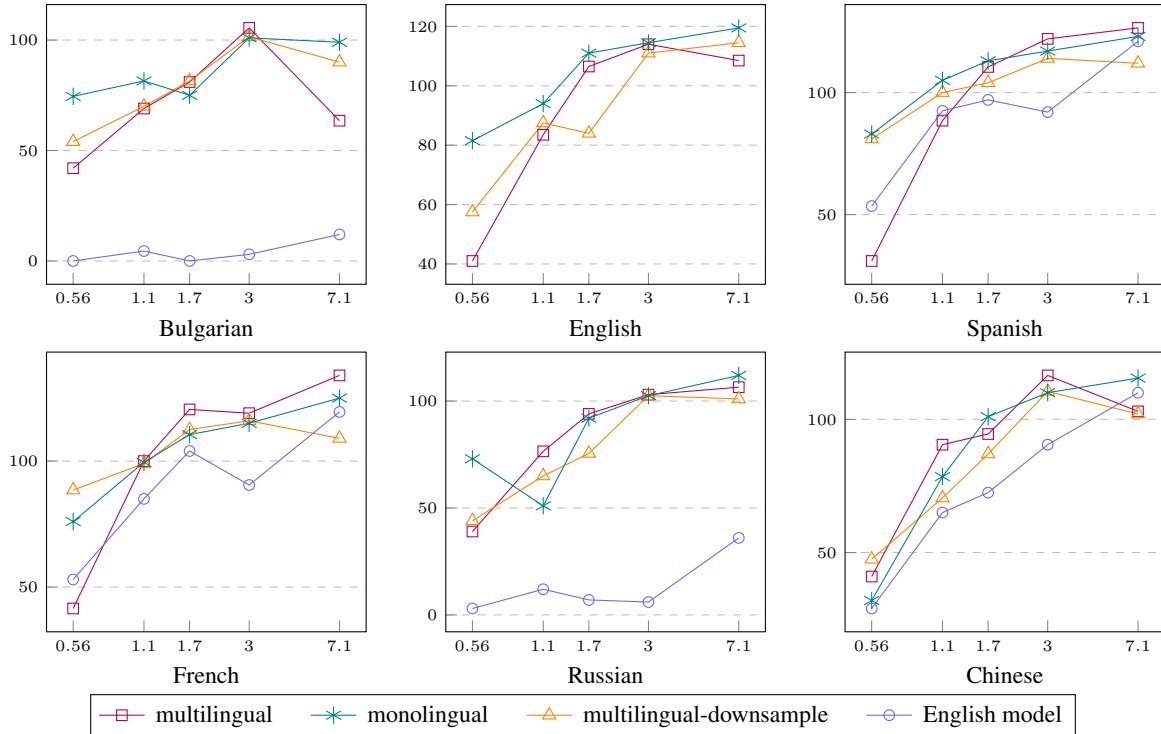


Figure 4: **Full-parameter** fine-tuning on **BLOOM** at different sizes. Caption: language generated; y-axis: evaluation score; x-axis: model size (B) on a logarithmic scale.

ected by the intended language’s closeness to English: Spanish and French can maintain reasonable scores, but Bulgarian, Russian, and Chinese are seen to have very low performance. The only exception is BLOOM full-parameter tuning shown in Figure 4, where the English model is not too behind other methods when operating in Chinese.

In Figure 3, an interesting observation on the English models tuned with LoRA is the performance peak at 1.1B for non-English tests, whilst the checkpoint itself does not stand out in the English test. At this particular size, the LoRA-tuned model learned to follow instructions without losing too much multilingual transfer ability from pre-training, despite being instruction-tuned in English.

On the other hand, with the same computational budget, models trained on downsampled multilingual data are significantly more robust across all test languages. They 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 created with machine translation instead of the full English data**. Nonetheless, if budget permits, training with the full multilingual data is still slightly better in most cases.

4.3 Unseen languages

Further in Figure 5, we include the performance of BLOOM models which underwent LoRA or full-parameter fine-tuning in various data conditions, but were subsequently used to respond in unseen languages, Bengali and Norwegian, at inference time.

Regarding the English-tuned LLMs, we observe different behaviours for LoRA and full-parameter fine-tuning. With the former, English models are nowhere near the performance of other multilingual tuned models, but with the latter, we see close or even better performance with English-only fine-tuning. It is thus implied that full-parameter instruction tuning can even lift performance for languages not present in the instruction dataset. However, we note that the results of full-parameter tuning on Norwegian could be considered an outlier given its comparably low scores.

Considering multilingual instruction tuning, we notice a pattern opposed to that on test languages seen during training, that learning on the downsampled data is superior to ingesting the full mixed data. Generally, we conclude that **it is important to not overfit to instruction languages if, in downstream tasks, queries in unseen languages are expected**.

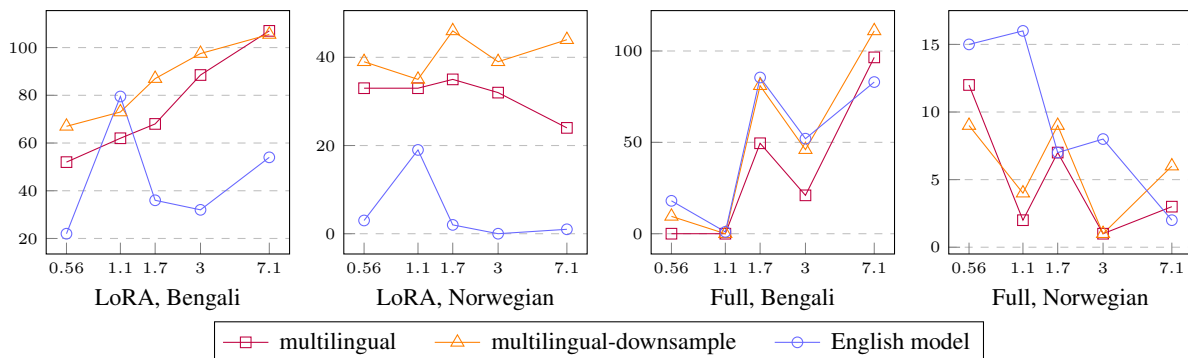


Figure 5: **LoRA** and **full-parameter** fine-tuning with **BLOOM** at different sizes, and tested on **unseen** languages. Caption: tuning method and language; y-axis: evaluation score; x-axis: model size (B) on a logarithmic scale.

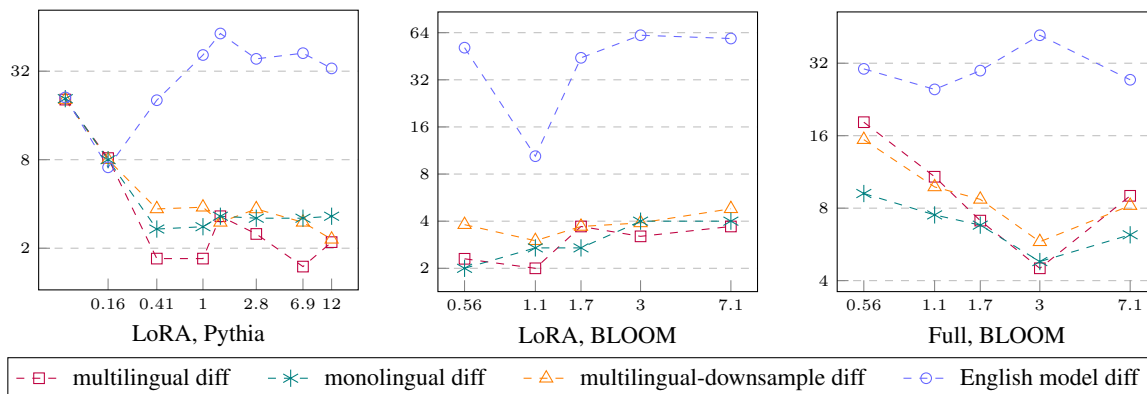


Figure 6: Evaluation **score difference** before and after language identification, averaged over test languages, for **LoRA** and **full-parameter** fine-tuning with **BLOOM** and **Pythia** at different sizes. Caption: tuning method and base model; y-axis: evaluation score difference on a logarithmic scale; x-axis: model size (B) on a logarithmic scale.

4.4 Language consistency

In practical use cases, an LLM response should be in the same language as the question, unless otherwise instructed. However, as noted in Section 3.4, our manual inspection reveals that in some cases the instruction-tuned LLM outputs in English regardless of the query language. We therefore review each model and data recipe’s scores before and after adding the language identification, to isolate the impact of an LLM’s language robustness from its responses’ inherent “quality” (regardless of the language).

We compute the *differences* in GPT evaluation scores before and after applying the language identification module: $\Delta_{score} = eval_score - eval_score \times language_id$. This is done for both full-parameter and LoRA tuning on BLOOM and Pythia. A score difference can be interpreted as how much a response is penalized due to being in the wrong language. A large difference implies that the model produced a reasonable answer in an undesired language. We report the *average* of the

score differences across all six test languages seen during the instruction-tuning time. These score differences are displayed in Figure 6 with dashed lines to distinguish from absolute model scores represented using solid lines in previous figures.

We find English-only models to be the least robust, as their score differences are significantly above other tuning techniques, across both BLOOM and Pythia. Focusing on LoRA training, we see that full multilingual tuning records the smallest performance drop, whereas when full-parameter fine-tuning is concerned, monolingual tuning has a smaller dip than multilingual tuning. The insights on language robustness are corroborated by our early findings on overall performance in Section 4.1: **superior results are obtained when using multilingual tuning with LoRA and monolingual tuning with full-parameter tuning**. On the other hand, all three tuning techniques are not too far apart; specifically for BLOOM LoRA tuning, the language consistency does not improve as the base model gets larger in size.

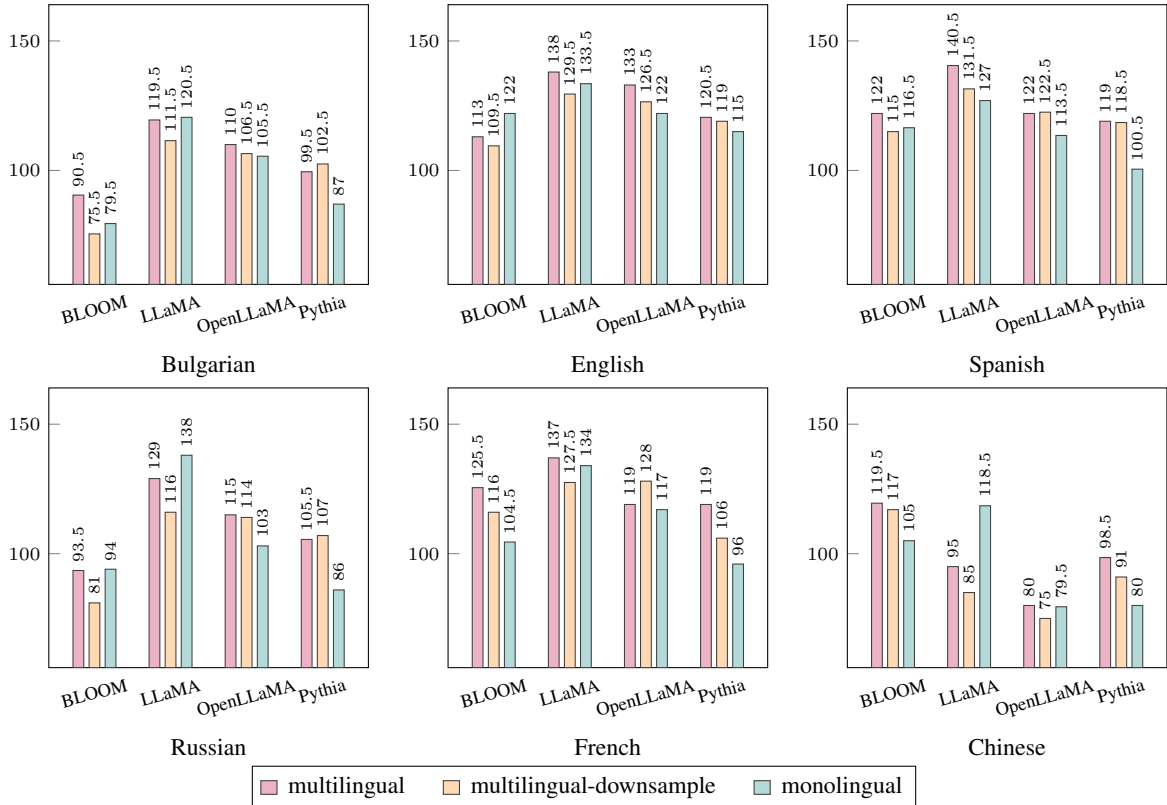


Figure 7: LoRA fine-tuning on 7B LLMs from different families. Caption: language generated; y-axis: evaluation score; x-axis: model family.

4.5 Model families

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

Delving into the comparison between monolingual and multilingual instruction tuning, we find that out of 24 cases across six languages and four LLMs, monolingual tuning is ahead in merely two cases: LLaMA tested in Russian and Chinese. Downsampled multilingual tuning is better than full multilingual training in two cases: Pythia tested in Bulgarian and OpenLLaMA tested in French, and it is on par in three other scenarios. Nonetheless, it has seen just a fraction of the cost of full multilingual training. The outcome of tuning LLMs from several families confirms that **multilingualism performs better with LoRA fine-tuning**.

5 Related Work

Recent years have witnessed rapid development in large language models, for example, the renowned closed-source GPT family (Brown et al., 2020) as well as open-source models like LLaMA (Touvron et al., 2023) and OpenLLaMA (Geng and Liu, 2023). In addition to English-centric models, multilingual language models have also been designed such that multiple languages can be dealt with by a single LLM, reducing operational costs. These models such as mT5 (Xue et al., 2021) and BLOOM (Scao et al., 2022) have effectively demonstrated multilingual understanding ability.

While foundational LLMs are trained to complete input texts, a new paradigm named instruction tuning can adjust such models to respond in a question-answering style (Wei et al., 2022a; Sanh et al., 2022). It continually trains an LLM by formatting a specific task as a natural language query and the task output as a text response. Longpre et al. (2023) investigated the factors of effective instruction tuning such as tasks and methods. Combining the capabilities of multilingual models with instruction fine-tuning opens up new opportunities for instruction following and content gener-

ation in multilingual scenarios. Li et al. (2023) showcased that multilingual instruction fine-tuning with translation instructions can improve the performance of machine translation. Muennighoff et al. (2022) found multilingual instruction fine-tuning gained better performance on natural language tasks than English-only fine-tuning. They also found that using low-cost machine translations is superior to tuning with human-written non-English prompts on multitask natural language understanding. Our study takes one step further by utilizing machine translation to produce parallel instruction data. This enables controlled settings for empirical analysis of monolingual language-specific and multilingual instruction tuning of LLMs.

6 Conclusion

This paper explores instruction fine-tuning of large language models in both monolingual and multilingual contexts. Our study using the Alpaca dataset and machine translations yields valuable insights in a controlled setting. We report the best data strategies for full-parameter and parameter-efficient tuning separately. Furthermore, when fine-tuning under a limited computational budget, a multilingual dataset offers more benefits compared to monolingual datasets, even when downsampled. Overall, our research provides practical guidance for expanding or maintaining the language capabilities in LLMs via instruction tuning within resource constraints.

Acknowledgements

This work stemmed from a [hackathon](#) project organized by the High Performance Language Technologies (HPLT) consortium. We are grateful to Andrey Kutuzov, David Samuel, Lucas Charpentier, Zhicheng Guo, Sampo Pyysalo, Jörg Tiedemann, Alicia Núñez Alcover, and Joonas Kytöniemi for discussions and test data translation.

The work has received funding from the European Union’s Horizon Europe research and innovation programme under grant agreement No 101070350 and from UK Research and Innovation (UKRI) under the UK government’s Horizon Europe funding guarantee [grant number 10052546]. The contents of this paper are the sole responsibility of the authors and do not necessarily reflect the opinion of the funders.

We also acknowledge CSC – IT Center for Science, Finland, as well as the Ministry of Education,

Youth and Sports of the Czech Republic through the e-INFRA CZ (ID:90254) for providing computational resources.

References

- 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*, pages 2397–2430. PMLR.
- Terra Blevins and Luke Zettlemoyer. 2022. Language contamination helps explain the cross-lingual capabilities of english pretrained models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3563–3574.
- Eleftheria Briakou, Colin Cherry, and George Foster. 2023. Searching for needles in a haystack: On the role of incidental bilingualism in palm’s translation capability. In *Proceedings of ACL*.
- 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. *Advances in neural information processing systems*, 33:1877–1901.
- 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)*, pages 865–879, Toronto, Canada. Association for Computational Linguistics.
- Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Édouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451.
- Yiming Cui, Ziqing Yang, and Xin Yao. 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*.

- 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 arXiv:2101.00027*.
- Xinyang Geng and Hao Liu. 2023. Openllama: An open reproduction of llama. GitHub.
- 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*, pages 427–431. Association for Computational Linguistics.
- 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 arXiv:2304.07327*.
- 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 1.6 tb composite multilingual dataset. *Advances in Neural Information Processing Systems*, 35:31809–31826.
- Jiahuan Li, Hao Zhou, Shujian Huang, Shanbo Chen, and Jiajun Chen. 2023. Eliciting the translation ability of large language models via multilingual finetuning with translation instructions. *arXiv preprint arXiv:2305.15083*.
- 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. *arXiv preprint arXiv:2301.13688*.
- 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)*, pages 3470–3487, Dublin, Ireland. Association for Computational Linguistics.
- Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng-Xin Yong, Hailey Schoelkopf, et al. 2022. Crosslingual generalization through multitask finetuning. *arXiv preprint arXiv:2211.01786*.
- 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, Elie 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.
- 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.
- Together Computer. 2023. RedPajama: An open source recipe to reproduce LLaMA training dataset. GitHub.
- 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 arXiv:2302.13971*.
- Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023a. 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, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023b. 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.
- Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. 2022a. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed H Chi, Quoc V Le, Denny Zhou, et al. 2022b. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*.
- 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*, pages 483–498.

Jiacheng Ye, Xijia Tao, and Lingpeng Kong. 2023a. Language versatilists vs. specialists: An empirical revisiting on multilingual transfer ability. *arXiv preprint arXiv:2306.06688*.

Seonghyeon Ye, Hyeonbin Hwang, Sohee Yang, Hyeongu Yun, Yireun Kim, and Minjoon Seo. 2023b. In-context instruction learning. *arXiv preprint arXiv:2302.14691*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Haoteng Zhang, Joseph Gonzalez, and Ioan Cristian Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *ArXiv*, abs/2306.05685.

A Language Inconsistency

Table 2 demonstrates an example, where the response from an instruction-tuned model 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.

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.