Monolingual or Multilingual Instruction Tuning: Which Makes a Better Alpaca

Anonymous ACL submission

Abstract

Foundation large language models (LLMs) can be instruction-tuned to perform open-domain question answering, facilitating applications such as AI assistants. While such efforts are often carried out in a single language, we empirically analyze cost-efficient approaches to multilingual tuning. 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. Under a fixed budget, comparisons show that multilingual tuning is on par or better than separately tuning a model for each language. Further, 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 with constrained computational resources.

1 Introduction

In natural language processing with large language models (LLMs), language capacity has attracted much attention (Conneau et al., 2020). Some pioneering works like BERT (Devlin et al., 2019) focused on a single language, while recent research usually adopts multilingual datasets, e.g. BLOOM (Scao et al., 2022). Models pre-trained with texts in many languages seem attractive when the downstream tasks require multilingual capabilities, because these reduce operational costs such as storage and enable zero-shot language transfer (Artetxe and Schwenk, 2019).

With autoregressive LLMs trained to complete inputs, instruction tuning makes them follow and respond to inputs (Sanh et al., 2022; Wei et al., 2022). Building on research that turns an LLM into a chat model in an inexpensive way (Alpaca, Taori et al., 2023), this work extends it to multilingualism. Unlike prior works on multilingual multitask tuning (Mishra et al., 2022; Muennighoff et al., 2023), we focus on open-ended question answering.

Our data setting combines two low-cost practices: self-instruct, which distils data from a powerful LLM (Wang et al., 2023) and the idea of leveraging machine translation to create multilingual datasets (Muennighoff et al., 2023). We fine-tune several LLMs with both full-parameter fine-tuning (FFT) and low-rank adaptation (LoRA, Hu et al., 2022), using different language combinations. Our experiments use a fixed 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. We also propose a budget-aware multilingual tuning scheme that is demonstrated to be more robust. Finally, we examine our conclusions by generalizing to unseen languages and various LLMs of roughly the same size.

2 Methodology

2.1 Instruction data

We follow the work of Alpaca by tuning base LLMs with instruction-response data. We use the publicly available cleaned version with 52K instances, and machine-translated it into eight languages: Bulgarian, Czech, Chinese, German, Finnish, French, Russian, and Spanish, using open-source systems.[1]

2.2 Budget-constrained instruction tuning

For monolingual tuning, we tune an LLM on each language data separately, whereas, for multilingual tuning, we combine and shuffle the data in all languages. This enables a resource-constrained comparison between monolingual and multilingual tuning, where a fixed computational budget is given 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

[1] Data and trained models will be disclosed and released.

it costs $N \times C_{alpaca}$ to tune individual models to support N languages of interest.

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

(3) As a baseline, we use an *English-tuned model* to respond to other languages. It has the same cost C_{alpaca} as a single monolingual Alpaca.

(4) *Downsampled multilingual*: we downsample the multilingual dataset in (2) to the size of a single monolingual dataset, with training cost C_{alpaca} too.

Our study covers two training paradigms: *low-rank adaptation* and *full-parameter fine-tuning*. Both continue-train an LLM with the causal language modelling objective using the instruction-response data, with hyperparameters listed in Appendix A.1. Five LLMs are involved: Baichuan-2 (Yang et al., 2023), BLOOM (Scao et al., 2022), LLaMA (Touvron et al., 2023), OpenLLaMA (Geng and Liu, 2023), and Pythia (Biderman et al., 2023), 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 in languages. We attach a detailed LLM description in Appendix A.2.

2.3 Evaluation setup

Test data Our instruction-tuned LLMs are benchmarked on languages both seen and unseen during 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, and French and Spanish are high-resource and 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 and uses a different script, whereas Norwegian is under-resourced but overlaps with English writing script to some extent.

LLM evaluation To avoid expensive evaluation costs, we adopt LLM-as-a-judge (Zheng et al., 2023) to score instruction-response pairs from 1 to 3, and the final score is obtained by summing up a model's scores across all test samples. We use GPT-3.5 (`gpt-3.5-turbo-0613`) as the judge; it is queried with a question-model 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 a question and a response should be in the same language. The exact wording is listed in Figure 5 in Appendix B.1.

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

LLM-human agreement Finally, we confirm strong LLM-human agreement in evaluation. We pick a total of 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, with details in Appendix B.3 and Table 3. Four languages, English, Spanish, Bulgarian, and Chinese, are human-evaluated. 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 evaluation scores. This indicates the reliability of LLM-as-a-judge in our study to draw meaningful conclusions.

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 7 due to space constraints. Moving on to full-

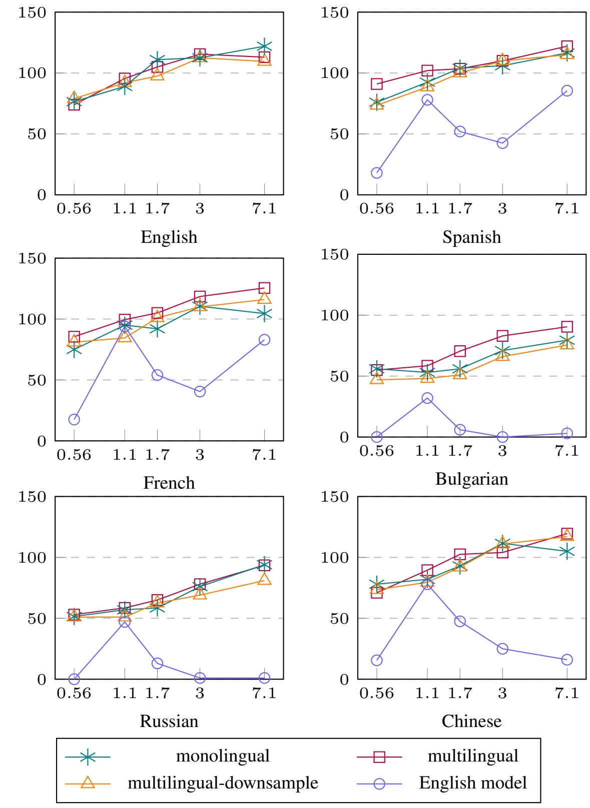

Figure 1: **LoRA** with **BLOOM** at different sizes. Caption: language; y-axis: evaluation score; x-axis: model size (B).

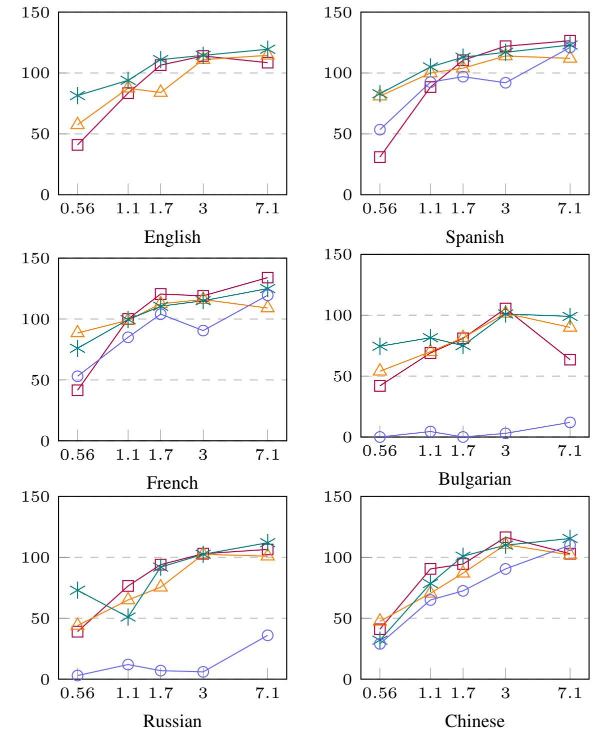

Figure 2: **FFT** with **BLOOM** at different sizes. Caption: language; y-axis: evaluation score; x-axis: model size (B). Same legend as Figure 1.

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 7 (in appendix C.1), 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 ().

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 tests but does not stand out in the English test. At this specific size, it learned to follow multilingual instructions despite being tuned in English, without losing much multilingual transfer ability from pre-training,

In contrast, using the same computational budget, downsampled multilingual tuning is 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 even created with machine translation instead of the full English data**. Nonetheless, if the budget permits, training with the full multilingual data is still slightly better.

3.3 Unseen languages

Further in Figure 3, we look at BLOOM models which underwent LoRA or FFT, but were subsequently used to respond in unseen languages at inference time. English-tuned LLMs behave differently for LoRA and FFT. With the former, they are nowhere near multilingual tuned models, but with the latter, we see close or even better performance. It thus implies that full-parameter tuning can even lift performance for languages not present in the instruction dataset. However, FFT results on Nor-

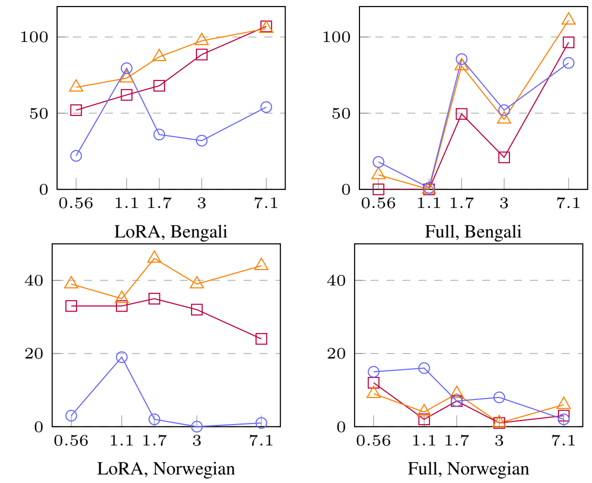

Figure 3: **LoRA** and **FFT** with **BLOOM** at different sizes and tested on **unseen** languages. Caption: tuning method and language; y-axis: evaluation score; x-axis: model size (B).

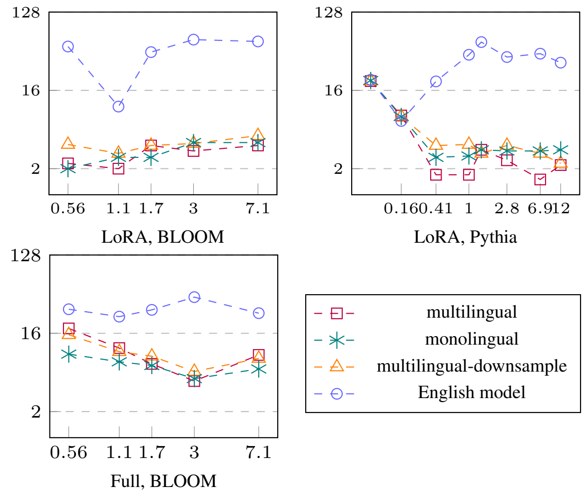

Figure 4: Evaluation **score change** before and after language identification, **averaged** over six seen test languages, at different LLM sizes. Caption: tuning method and base model; y-axis: evaluation score difference; x-axis: model size (B).

wegian 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 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. A (big) difference means that the model produces reasonable answers in an undesired language. We report the *average* of the score differences across all six test languages seen during tuning, displayed in Figure 4.

English-only models are the least robust, as their score differences are greatly above other techniques. With LoRA, full multilingual tuning records the smallest performance drop; with FFT, monolingual tuning is favoured. The insights on language robustness are corroborated by our early findings on overall performance 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 with sizes of around 7 billion. In Appendix C.2 Figure 8, 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 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 merely two cases: LLaMA tested in Russian and Chinese. The cost-efficient multilingual downsampled 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 LLMs from several families confirms that **multilingual fine-tuning performs better with LoRA**.

4 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

We mostly compared LLMs with around 7B parameters due to the limitation of computing resources. The best checkpoint for each fine-tuning condition is selected based on cross-entropy, but there is no guarantee that this leads to the best model 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.

References

Mikel Artetxe and Holger Schwenk. 2019. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *TACL*.

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 *Proceedings of ICML*.

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 *NeurIPS*.

Laurie Burchell, Alexandra Birch, Nikolay Bogoychev, and Kenneth Heafield. 2023. An open dataset and model for language identification. In *Proceedings of ACL*.

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 ACL*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL*.

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.

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 *Proceedings of ICLR*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of EACL*.

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 *NeurIPS Datasets and Benchmarks Track*.

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*.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, and Adam Roberts. 2023. The flan collection: Designing data and methods for effective instruction tuning. In *Proceedings of ICML*.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2022. Cross-task generalization via natural language crowdsourcing instructions. In *Proceedings of ACL*.

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 ACL*.

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 *Proceedings of ICLR*.

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*.

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*.

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 *ACL*.

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 *Proceedings of ICLR*.

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 NAACL*.

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*.

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. *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 around 15 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 (regarded 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.

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

Table 1: Hyperparameter configurations of LoRA and 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 paper are publicly available and free for academic purposes.

Baichuan-2 It is a multilingual LLM trained on 2.6 trillion tokens. While its data composition does not outline the languages included, the LLM performs strongly on tasks in English and Chinese. We use its 7B checkpoint.

BLOOM It 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 It 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 This 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 It 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

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

B.2 A case of 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.

B.3 Human evaluation details

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 6. 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.

```
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 6: Instructions for human evaluators.

	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 correlation with GPT-3.5 scores at the system level. Models are fine-tuned with LoRA.

C Result Details

C.1 Experiments on LoRA with Pythia

Apart from LoRA fine-tuning on BLOOM models, we conduct the same investigation on Pythia at different sizes. We observe the same pattern as we find on BLOOM models explained in Section 3.1. The plots for the six languages are included as Figure 7.

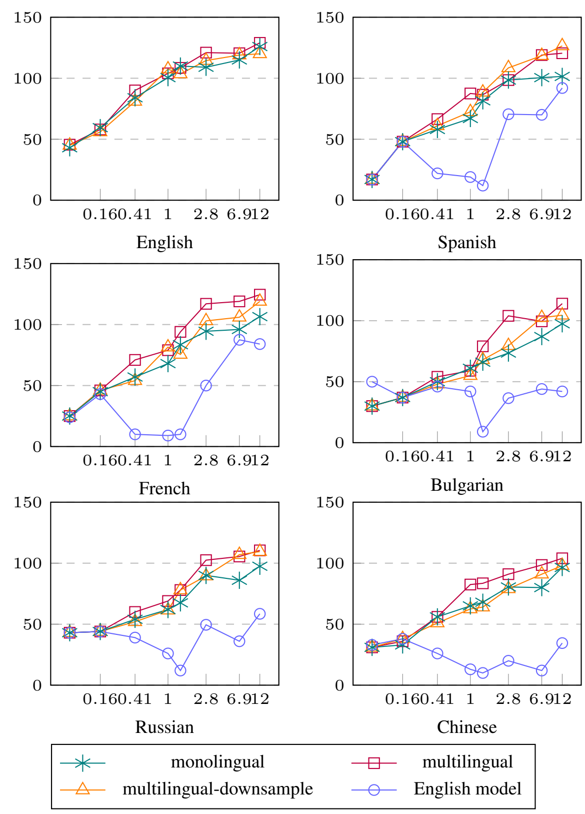

Figure 7: **LoRA** fine-tuning on **Pythia**. Caption: language generated; y-axis: evaluation score; x-axis: model size (B) on a logarithmic scale.

C.2 Experiments LLM families

In Figure 8 we attach the bar plots of LoRA fine-tuning with LLMs at 7B from different families.

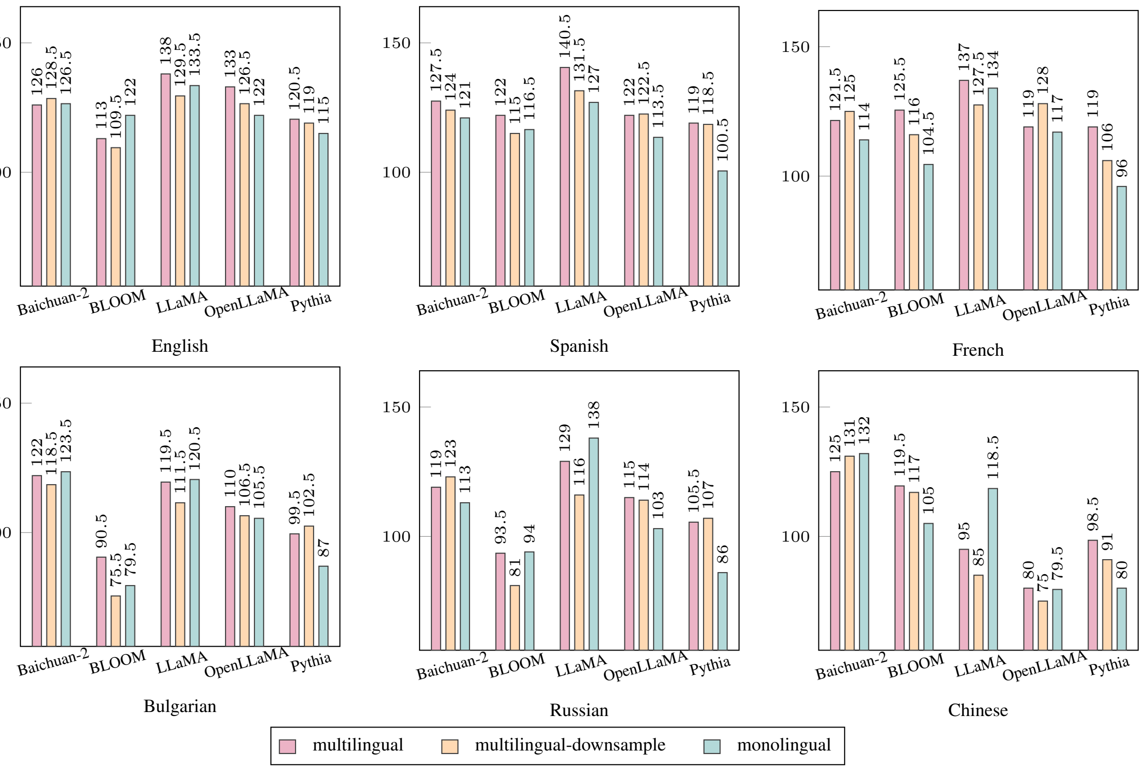

Figure 8: LoRA fine-tuning on 7B LLMs from different families. Caption: language generated; y-axis: evaluation score; x-axis: model family.

D 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., 2022; 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 generation 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. (2023) 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.