BLOOM+1: Adding Language Support to BLOOM for Zero-Shot Prompting

Zheng-Xin Yong[1]*, **Hailey Schoelkopf**[2,3], **Niklas Muennighoff**[4], **Alham Fikri Aji**[5], **David Ifeoluwa Adelani**[6], **Khalid Almubarak**[7], **M Saiful Bari**[8], **Lintang Sutawika**[2,9], **Jungo Kasai**[10], **Ahmed Baruwa**[11], **Genta Indra Winata**[12], **Stella Biderman**[2,13], **Edward Raff**[13], **Dragomir Radev**[3], **Vassilina Nikoulina**[14]

[1]Brown University [2]EleutherAI [3]Yale University [4]Hugging Face [5]MBZUAI
[6]University College London [7]PSAU [8]Nanyang Technological University
[9]Datasaur.ai [10]Paul G. Allen School of CSE, University of Washington
[11]University of Oregon [12]The Hong Kong University of Science and Technology
[13]Booz Allen Hamilton [14]NAVER LABS Europe

Abstract

The BLOOM model is a large publicly available multilingual language model, but its pretraining was limited to 46 languages. To extend the benefits of BLOOM to other languages without incurring prohibitively large costs, it is desirable to adapt BLOOM to new languages not seen during pretraining. In this work, we apply existing language adaptation strategies to BLOOM and benchmark its zero-shot prompting performance on eight new languages in a *resource-constrained* setting. We find language adaptation to be effective at improving zero-shot performance in new languages. Surprisingly, we find that adapter-based finetuning is more effective than continued pretraining for large models. In addition, we discover that prompting performance is not significantly affected by language specifics, such as the writing system. It is primarily determined by the size of the language adaptation data. We also add new languages to BLOOMZ, which is a multitask finetuned version of BLOOM capable of following task instructions zero-shot. We find including a new language in the multitask fine-tuning mixture to be the most effective method to teach BLOOMZ a new language. We conclude that with sufficient training data language adaptation can generalize well to diverse languages. Our code is available at https://github.com/bigscience-workshop/multilingual-modeling.

1 Introduction

Although access to transformer-based language models has expanded greatly over the past several years (Black et al., 2021; Wang and Komatsuzaki, 2021; Artetxe et al., 2021; Black et al., 2022; Zhang et al., 2022), these technologies are overwhelmingly concentrated in a few high resource languages (Talat et al., 2022). BLOOM (Scao et al., 2022), the largest publicly available multilingual language model to date with 176B parameters, covers only 46 natural languages and even excludes high-resource languages such as Korean and Russian which has tens of millions of speakers. This limitation was driven by a number of factors, most notably only considering languages for which the community had enough expertise to manually validate the data quality (Kreutzer et al., 2022), deduplicate and remove personally identifiable information (Laurençon et al., 2022) and had sufficient access to licensed unlabeled text (Joshi et al., 2020). All of these factors are contingent facts about the group that trained the model, and leave open the idea that other researchers could contribute more languages. As regularly retraining such a model is prohibitively expensive, the question of whether this model can be productively *adapted* to understand additional languages after training becomes pressing.

We hypothesize that language adaptation scenario is especially interesting for low-resource languages that would benefit from knowledge transfer. Therefore, we adapt BLOOM models to support eight new languages (German, Russian, Bulgarian, Thai, Turkish, Greek, Korean, and Guarani) in the resource-constrained settings, where we only use a limited amount of samples (maximum 100K samples) for each language. We evaluate their zero-shot prompting on various NLU tasks after adaptation. The new languages cover both seen and unseen scripts in the pretraining data, and they differ in their language families and word orders. We benchmark existing language adaptation methods, such as continued pretraining and MAD-X (Pfeiffer et al., 2020), as well as a state-of-the-art parameter-efficient transfer learning method, $(IA)^3$ (Liu et al., 2022).

*Corresponding author: contact.yong@brown.edu

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 11682–11703
July 9-14, 2023

Current work on adapting large multilingual models has mostly explored continued pretraining (Müller and Laurent, 2022; NovelAI, 2022; De la Rosa and Fernández, 2022) of EleutherAI's GPT-J-6B (Wang and Komatsuzaki, 2021). Moreover, Ebrahimi and Kann (2021) showed that continued pretraining outperforms other strategies for adapting small/medium-sized language models (i.e., models with fewer than one billion parameters). However, our experiments demonstrate that, for large language models such as BLOOM with comparable sizes to GPT-J-6B, continued pretraining underperforms adapters under a resource-constrained setting. In addition, our work focuses on studying the effects of language adaptation on prompting, which has been underexplored in previous language adaptation work (Ebrahimi and Kann, 2021; Ansell et al., 2022; Parović et al., 2022; Pfeiffer et al., 2022). Prompting can benefit many languages that lack large amounts of labeled data as it allows language models to generalize to a wide range of tasks with significantly less training cost and data than full finetuning (Liu et al., 2021; Le Scao and Rush, 2021).

1.1 Our Contributions

Our work is the first to explore the *scaling effects* of *language adaptation* strategies for language models with billions of parameters under a *resource-constrained* setting. Contrary to prior work on small/medium-sized multilingual masked language models (Ebrahimi and Kann, 2021), we recommend training adapters instead of continued pretraining for BLOOM with at least 3 billion parameters for better prompting performance. We further connect this recommendation to the way the quality of language independent representation scales with model parameters.

We also demonstrate the positive effects of monolingual language adaptation on the prompting performance of BLOOM on various datasets. BLOOMZ is a variant of BLOOM that is produced by finetuning BLOOM on a multitask mixture in the same languages seen during pretraining. We find that simply adding a new language in the multitask finetuning is effective in improving performance in the new language.

To summarize, our contributions include:

- Studying the effects of language adaptation on zero-shot prompting and instruction tuning.
- Benchmarking parameter-efficient methods for adapting BLOOM models of various scales and analyzing the trade-offs between the amount of required computes and zero-shot prompting performance.
- Quantifying the effect of the size of language adaptation data on language adaptation.

2 Related Work

Language Adaptation Language adaptation enables pretrained language models to support languages outside of their pretraining data. Most works investigating language adaptation consider masked language models such as mBERT (Devlin et al., 2019) and XLM-R (Conneau et al., 2020) that are pretrained on 100+ languages. Language adaptation approaches can be broadly categorized into three categories: (1) *continued pretraining* of the model (restricted to the embedding layer training only in some cases) (Neubig and Hu, 2018; Artetxe et al., 2020; Chau et al., 2020; Muller et al., 2021; Zhang et al., 2020; Wang et al., 2020); (2) training of *language-specific adapters* (Pfeiffer et al., 2020, 2021a,b; Philip et al., 2020; Üstün et al., 2021; Berard, 2021; Faisal and Anastasopoulos, 2022; Parović et al., 2022) for the target language; and (3) training of a *sparse subset* of model parameters (Ansell et al., 2022). The core motivation behind these approaches is to benefit from knowledge transfer encoded in the pretrained language models for the new language processing at a small computational cost (compared to full model retraining from scratch).

One common issue is that the script of the new language is not always supported by the tokenizer. Artetxe et al. (2020); Aji et al. (2020); Pfeiffer et al. (2021b) demonstrate that it is possible to add a new language to these models by training a new embedding layer. Muller et al. (2021) continue training the pretrained mBERT on the new language data, and find that transliteration of languages using non-Latin script boosts performance on these languages. Berard (2021) add new languages into pretrained multilingual machine translation models by training embedding and adapter layers. They show that adding a new target language (the language to translate to) is harder to learn than a new language to translate from.

Closest work to our benchmarking efforts is Ebrahimi and Kann's (2021) study on different approaches (i.e., continued pretraining, vocabulary expansion and adapter layers) to extend the XLM-R model to 30 new languages on token-level clas-

sification tasks. They conclude that continued pretraining is the most promising direction. However, the cost of such pretraining will grow with the size of the pretrained model and can be prohibitive for many researchers working with low-resource languages. Our results also show that continued pretraining does not necessarily bring a prompting performance gain for larger language models.

Multilingual Prompting Prompting reformulates NLP tasks into masked or generative language modeling problem, depending on the models' pretraining objective. Zhao and Schütze (2021) and Qi et al. (2022) show that finetuning XLM-R on cloze-style prompts yield better performance than standard finetuning under a low-resource regime for XNLI. On the other hand, Winata et al. (2022) find that standard finetuning of XLM-R outperforms prompt-based learning for sentiment prediction in low-resource Indonesian dialects.

Some work shows that multitask prompt-based training on a variety of tasks and English or translated prompts improves zero-shot cross-lingual and cross-task performance (Muennighoff et al., 2022; Fu et al., 2022). Multilingual prompt-based learning can also be achieved without performing gradient updates for downstream tasks. For instance, Lin et al. (2021) demonstrate success in prompting GPT-like pretrained models with in-context learning for NLU tasks, using either English or translated prompt templates. Shi et al. (2023) find that when language models scale up, they can perform better multilingual chain-of-thought reasoning.

3 Experimental settings

3.1 BLOOM pretrained models

We focus on adding language support to the BLOOM language model (Scao et al., 2022) from 560 million to 7.1 billion parameters. BLOOM has a decoder-only Transformer architecture that uses AliBi positional embeddings (Press et al., 2022) and layer normalization after embedding layers. Its tokenizer is trained with byte-level Byte Pair Encoding (BPE) algorithm (Gage, 1994; Sennrich et al., 2016) with a vocabulary size of 250,680.

BLOOM is pretrained for around 350 billion tokens on the ROOTS corpus (Laurençon et al., 2022), which covers 46 natural languages and 13 programming languages. Appendix M shows the distribution of the natural languages in the ROOTS corpus.

3.2 New Languages

We consider all six languages of XNLI (Conneau et al., 2018) that are currently unsupported by BLOOM: German, Bulgarian, Russian, Greek, Turkish, and Thai. We also include Korean to follow up on past work on adapting the previous version of BLOOM (Yong and Nikoulina, 2022) and Guarani, which is a truly low-resource Native American language. Table 1 summarizes the unseen languages used in our experiments. They cover different language families and some of them do not share scripts with BLOOM's supported languages.

3.3 Language Adaptation Strategies

We carry out three language adaptation strategies to analyze their effects on zero-shot prompting. [1]

Continued Pretraining Continued pretraining strategy refers to continually training the BLOOM model with its causal language modeling pretraining objective on monolingual text of the new language (Chau et al., 2020; Ebrahimi and Kann, 2021; Muller et al., 2021).

MAD-X We use the language adapter and the invertible adapter of the MAD-X configuration (Pfeiffer et al., 2020) to adapt BLOOM to new languages. Language adapter refers to the bottleneck adapter with down- and up-projection feedforward layers (Houlsby et al., 2019; Pfeiffer et al., 2021a) that are inserted into each Transformer block. The invertible adapter is used in the embedding layers to mitigate the mismatch between the original and new language vocabularies.

$(IA)^3$ $(IA)^3$ is a parameter-efficient finetuning method that performs element-wise rescaling of inner Transformer block activations through learnable vectors (Liu et al., 2022). These vectors can be merged with the original pretrained weights of a model at inference to reduce latency by avoiding passing the activations through additional adapter modules.

We experiment with $(IA)^3$ since it outperforms bottleneck adapters, which are used in MAD-X, and other parameter-efficient finetuning methods such as BitFit (Ben Zaken et al., 2022), LoRA (Hu et al., 2022), and FishMask (Sung et al., 2021) on English NLU tasks (Liu et al., 2022). Our preliminary experiments show that $(IA)^3$ performs better

[1] We also ran preliminary experiments on Composable Sparse-Finetuning (see Appendix D), which is one of the state-of-the-art language adaptation strategies.

Language	Language Family	Word Order	Script	Space-Separated	Seen Script
German	Indo-European (Germanic)	SVO	Latin	✓	✓
Bulgarian	Indo-European (Slavic)	SVO	Cyrillic	✓	✗
Russian	Indo-European (Slavic)	SVO	Cyrillic	✓	✗
Greek	Indo-European (Hellenic)	SVO	Greek	✓	✗
Turkish	Turkic	SOV	Latin	✓	✓
Korean	Koreanic	SOV	Hangul	✓	✗
Thai	Tai–Kadai	SVO	Thai	✗	✗
Guarani	Tupian	SVO	Latin	✓	✓

Table 1: Information about the unseen languages used in our experiments.

than these methods (see Appendix G), and thus we only run $(IA)^3$ due to computational constraints.

As $(IA)^3$ does not adapt the embedding layer, we couple $(IA)^3$ with invertible adapters for fairer comparison with MAD-X language adapters. Our preliminary experiments (Table 4) show performance gains when using invertible adapters with $(IA)^3$.

3.4 Language Adaptation Setting

We randomly sample 100K samples from the deduplicated OSCAR subcorpora (Ortiz Suárez et al., 2019) of the respective languages for language adaptation to simulate low-resource settings. Since Guarani only has around 100 samples in OSCAR, we use Jojajovai parallel corpora (Chiruzzo et al., 2022), which contains 30K Guarani sentences. We perform 25K language adaptation training steps using a batch size of 8 and the sequence length of 1,024. See Appendix H for further details.

We do not retrain the tokenizer as BLOOM uses byte-level BPE tokenization, which never produces unknown tokens; therefore, we can perform language adaptation *without* extending the vocabulary. We adapt the embedding layer in two different fashions. For continued pretraining, we make the embedding layer trainable. This follows prior work on language adaptation (Pfeiffer et al., 2020; Chau et al., 2020; Ebrahimi and Kann, 2021; Fujinuma et al., 2022). For MAD-X and $(IA)^3$, we use invertible adapters to adapt the embedding layer while keeping the embeddings frozen.

3.5 Tasks and Prompt Templates

We evaluate the models on five multilingual NLU tasks, which cover natural language inference (XNLI (Conneau et al., 2018), KLUE-NLI (Park et al., 2021), and AmericasNLI (Ebrahimi et al., 2022)), commonsense reasoning (XCOPA (Ponti et al., 2020) and XStoryCloze (Lin et al., 2021)), anaphora resolution (XWinograd (Tikhonov and Ryabinin, 2021)), and paraphrasing (PAWS-X (Yang et al., 2019)). We perform zero-shot prompting *without any task-specific finetuning* and simply reuse the templates used to prompt the XGLM model Lin et al. (2021) without performing any prompt engineering. We translate the prompt templates using automatic translation APIs, and the translated templates can be found in Appendix F.

3.6 Baselines

We compare the adapted BLOOM model against generative multilingual language models which have reported state-of-the-art prompting performance. We also report the prompting performance of the original BLOOM models without any adaptation.

XGLM XGLM models (Lin et al., 2021) cover 30 natural languages and come in five different numbers of parameters: 564M, 1.7B, 2.9B, 4.9B, and 7.5B.

mGPT mGPT (Shliazhko et al., 2022) is a GPT model trained on 60 languages from 25 language families using Wikipedia and Colossal Clean Crawled Corpus. It only has 1.3B parameters.

BLOOMZ and mT0 BLOOMZ and mT0 are BLOOM and mT5 models finetuned on a multilingual task mixture, xP3 (Muennighoff et al., 2022). Here we report performance on the best prompts, which corresponds to instructions being in English while the context and the label are generally non-English. We also do not report performance on PAWS-X data since it is part of the xP3 training mixture.

Among the baselines, XGLM, mGPT, and mT0 have seen all the new languages in Table 1 except Guarani during model pretraining.

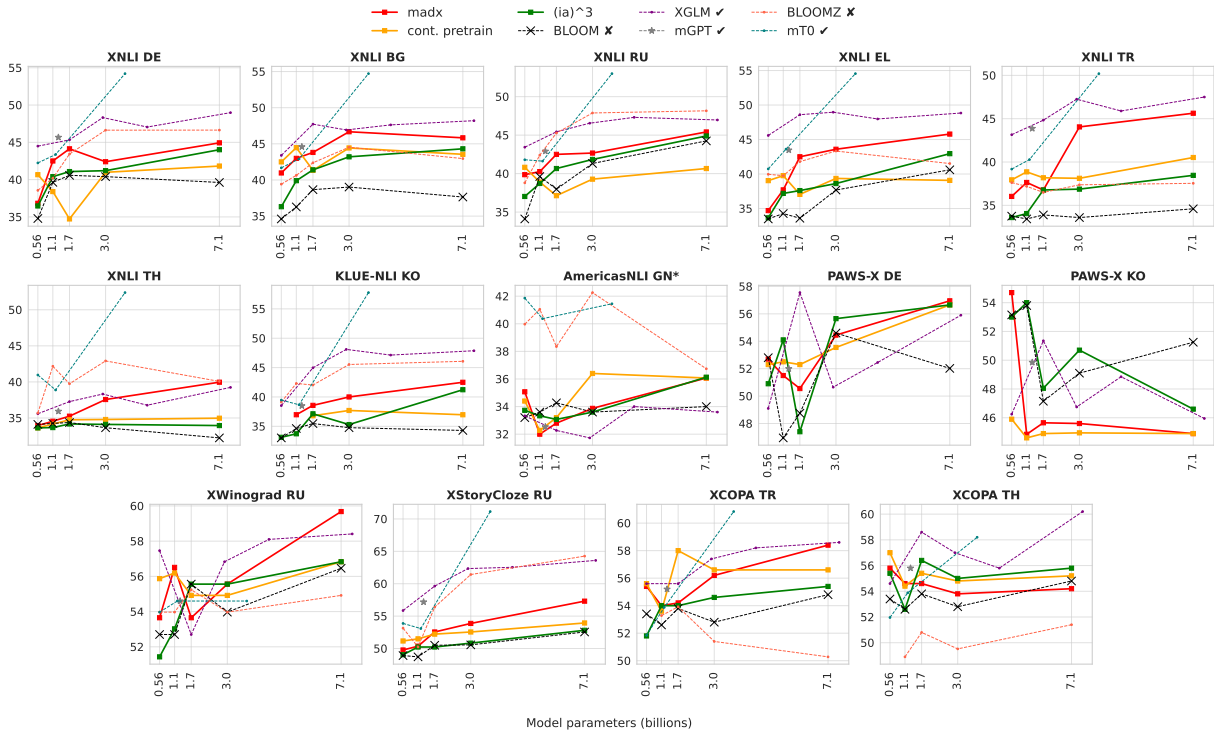

Figure 1: Results for zero-shot prompt-based evaluation of natural language inference, commonsense reasoning, anaphora resolution, and paraphrasing tasks. All tasks are evaluated with accuracy measure. Solid lines indicate language adaptation strategies, and dotted lines indicate baselines. × indicate the non-adapted BLOOM model. Both ✓ and ✗ indicate whether the baseline has seen the language during pretraining, except for Guarani (GN) that is unseen for all models. We also ablate BLOOMZ and mT0 from PAWS-X evaluation as the models has been trained on the task.

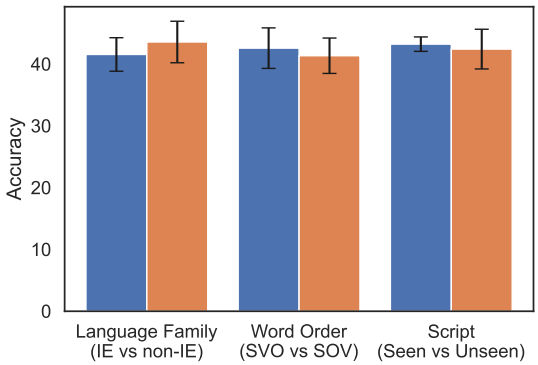

Figure 2: Average XNLI prompting performance for different categories of languages, split by whether it belongs to Indo-European (IE) family (left), whether its word order is SVO or SOV (middle), and whether its script system is seen during pretraining (right).

4 Results and Discussion

4.1 Zero-shot Prompting Performance

Figure 1 shows that language adaptation improves the original BLOOM's zero-shot prompting for unseen languages under the resource-constrained setting. Furthermore, in general, language adaptation follows the scaling law which dictates that performance gains correlate with model sizes. We note that when the BLOOM transformer model becomes wider (from 560M to 1.7B parameters), certain tasks such as German XNLI and PAWSX experience performance drops.

For the smallest BLOOM model with 560 million parameters, we see that continued pretraining yields the best prompting performance. Our result supports Ebrahimi and Kann's (2021) findings that continued pretraining of masked language models of similar size, such as mBERT and XLM-Roberta, gives better NER and POS tagging performance than adapters. However, **when model sizes increases beyond 3 billion parameters, adapter-based language adaptation methods outperform continued pretraining** despite having fewer trainable parameters. Furthermore, contrary to previous findings (Yong and Nikoulina, 2022), BLOOM adapts well to new languages regardless of their lan-

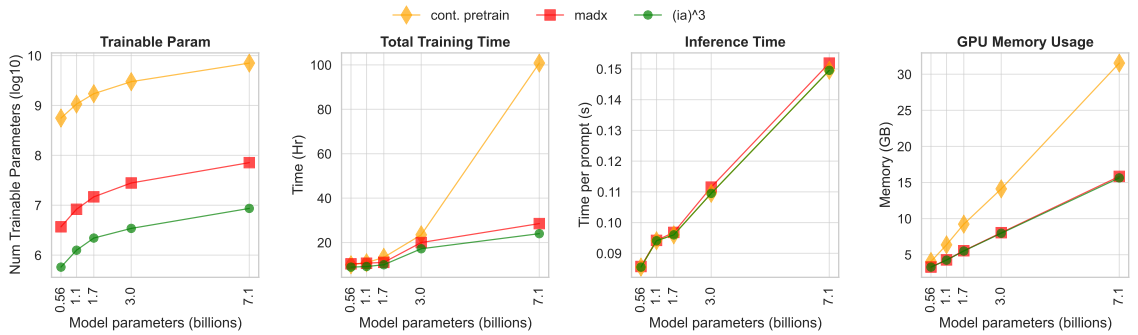

Figure 3: Comparison between different language adaptation strategies for BLOOM models on the number of trainable parameters (↓), total training time (↓), inference 'time per prompt on XNLI test set (↓), and maximum GPU memory usage (↓) on a single A100 GPU machine.

guage family, word order, and whether they share the same script system with languages in pretraining data (Figure 2). We note that there are many differences in Yong and Nikoulina's (2022) setting. Yong and Nikoulina (2022) used a multilingual model that uses learned positional embeddings instead of Alibi (Press et al., 2022) and that only supports 13 languages. They also finetuned both the learned positional and word embedding layers.

We find that the adapted BLOOM matches mGPT's performance in several XNLI tasks and even outperforms XGLM and mT0 on the German PAWS-X and Russian XWinograd tasks. Nonetheless, mT0, which has seen the languages during pretraining and is trained on a multilingual task prompts mixture, exhibits the best zero-shot prompting performance when model parameters are increased.

We find the adapted BLOOM performs poorly on Guarani, which is a truly low-resource language. Language adaptation only boosts the performance when models beyond 3 billion parameters are used. We believe this is due to the limited Guarani adaptation training data (30K as opposed to 100K for other languages) as supported by the findings in Section 4.4.

Best Language Adaptation Strategy We recommend that the smallest BLOOM model should be adapted with continued pretraining, but larger BLOOM models should be adapted with adapters due to better performance (Figure 1) and compute efficiency (Figure 3). We find MAD-X language adapters give better average zero-shot prompting performance, but $(IA)^3$ adapters have a slight edge in training efficiency due to significantly fewer trainable parameters and smaller training time for larger models.

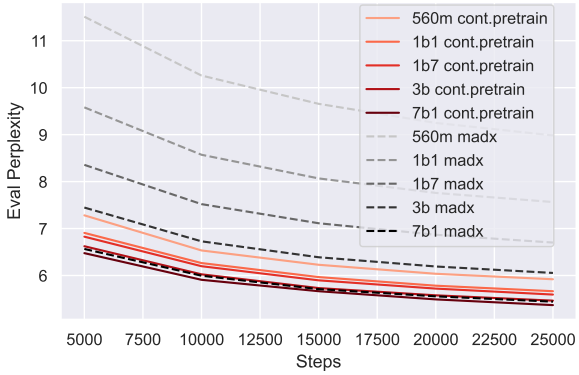

Figure 4: Perplexity curves of continued pretraining and MAD-X language adapters across all BLOOM model sizes on Russian held-out data.

4.2 Perplexity

Perplexity can be viewed as a measure of uncertainty when predicting the next token in a sequence, and better language modeling ability means lower perplexity. Figure 4 shows that evaluation perplexity on Russian texts for continued pretraining and MAD-X language adapters. We find that **perplexity during language adaptation training does not necessarily correlate with prompting performance.** While perplexity becomes lower for larger models, there is a drop in XWinograd performance for both language adaptation strategies when the model capacity increases from 1.1 billion to 1.7 billion parameters. Furthermore, even though continued pretraining has a lower perplexity than MAD-X language adapters, which suggests that continually-pretrained models better model the Russian OSCAR data, continually-pretrained BLOOM underperform their counterparts for larger model sizes in both XWinograd and XNLI tasks. This finding is in line with Liang et al.'s (2022) work that highlights the mismatch between perplexity and downstream

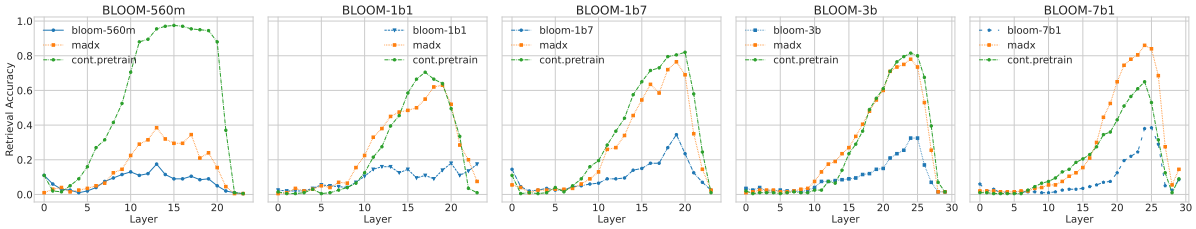

Figure 5: Sentence retrieval accuracy for Russian before and after adaptation with MAD-X adapters and continued pretraining.

task performance.

4.3 Connection to Language Independent Representation

Figure 5 reports sentence retrieval (SR) accuracy for Russian for non-adapted models, as well as models adapted via MAD-X adapters or continued pretraining. We use sentence retrieval accuracy as a way to measure quality of language independent representation, more details in the Appendix B. Note, that in this setting the representations of Russian are based on the adapted model, while representations of English are based on the original model, which excludes the problem of potential catastrophic forgetting. We see that before adaptation, the SR accuracy is very low overall, but bigger model demonstrate better SR results. With adaptation, SR accuracy drastically improves.

For BLOOM adapted with MAD-X, SR accuracy improves as model grows in parameters. The reason is that adapters' trainable parameters grow in size so they represent Russian sentences better and larger model start from better representations of both languages. **Interestingly, for continued pretraining, the best SR accuracy result is achieved with the smallest BLOOM model with 560 million parameters**, while larger models achieve much lower SR accuracy. This phenomenon *goes against the scaling law* and is opposite to what has been observed for MAD-X. [2]

Some previous works (Dufter and Schütze, 2020) suggest that smaller model would emerge better language-independent representations as it is forced to reuse the same parameters for different languages. However, when model grows it has more freedom to partition its' parameters between languages. Note that this observation has been made in the synthetic settings and to the best of our knowledge has not been confirmed in real multilingual models. Our results in Figure 5 could be seen as an additional support to that initial hypothesis. When doing continued pretraining with relatively small set of the language adaptation data, there are many ways for the model to optimize it's performance (cf Lottery ticket hypothesis (Frankle and Carbin, 2019)). If the model had more freedom to partition its' parameters between different languages, there is no guarantee that the continued pretraining would leverage English-related parameters and therefore could diverge its representation space further away from English. We hypothesize that this could be a possible explanation of degradation of continued pretraining sentence retrieval accuracy for larger models.

4.4 Amount of Language Adaptation Data

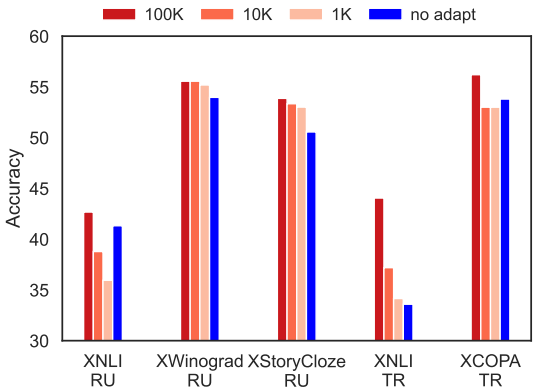

Figure 6: Effects of the amount of language adaptation training data on zero-shot prompting of various Russian (RU) and Turkish (TR) tasks. "No adapt" denotes the non-adapted BLOOM model.

We simulate different low-resource settings with BLOOM-3B using different amounts of adaptation training data. We use 1K, 10K and 100K samples to simulate different degrees of low-resource settings (see Figure 12). Figure 6 demonstrates a positive correlation between the size of adaptation training data and zero-shot prompting performance. We see that, when adapted with less than 100K sam-

[2] We have observed similar trends for models adapted for German.

ples, BLOOM performs worse than its non-adapted counterpart for tasks such as Russian XNLI and Turkish XCOPA. In other words, based on Figure 6 and Table 6, **we need around 100 million tokens of the new language for effective language adaptation**. However, surprisingly, the extent of the negative effect of low-resource setting can be limited to the type of tasks. For instance, for the same language Russian, we observe a limited effect of low-resource setting on XWinograd and XStoryCloze prompting.

4.5 Adapters' Capacity

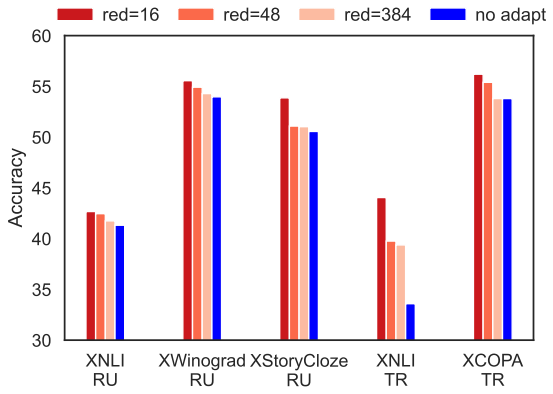

Figure 7: Effects of the MAD-X language adapters' reduction factors on zero-shot prompting of various Russian (RU) and Turkish (TR) tasks. "No adapt" denotes the non-adapted BLOOM model.

We investigate the effect of the size of adapters' capacity by varying the reduction factor (also known as compression rate (Rücklé et al., 2021)) in the adapter's bottleneck layer.[3] A smaller reduction value would lead to a larger amount of adapter parameters. Contrary to Yong and Nikoulina (2022), we observe a positive correlation between the amount of adapters' parameters and prompting performance (see Figure 7).

4.6 Adapting BLOOMZ

We also investigate language adaptation strategies for BLOOMZ, which is BLOOM finetuned on many different task prompts to achieve better cross-lingual and cross-task generalization (Muennighoff et al., 2022).

[3]We also investigate the effects of the placement of adapters, invertible adapters, and model pretraining on language adaptation (see Appendix J and K).

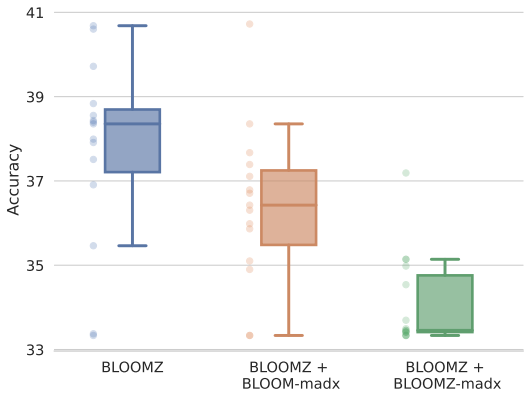

Figure 8: Zero-shot prompting performance of adapted BLOOMZ-560m on German XNLI task. Each dot represents the accuracy of one prompt template, where blue dots indicate the results of non-adapted BLOOMZ and red dots BLOOMZ with adapters.

4.6.1 Adding Language Support through Unlabeled Data

Similar to adapting BLOOM, we train MAD-X language adapters for BLOOMZ using the same experimental setting on monolingual OSCAR data. In Figure 8, we show that BLOOMZ-560m has a median accuracy of around 38.5% for the German XNLI tasks (left bar), but after language adaptation, it performs the worst with an accuracy as poor as a random classifier at 33% (right bar). However, when equipped with BLOOM's language adapters (this is possible because BLOOM and BLOOMZ share the same architecture), BLOOMZ retains its prompting ability (middle bar). The result suggests that **BLOOMZ loses its prompting capability gained from multitask instruction tuning after language adaptation** on the free-form text of monolingual OSCAR corpora.

4.6.2 Adding Language Support through Instruction Tuning

We experiment with learning a new language during instruction tuning using the same recipe as BLOOMZ (Muennighoff et al., 2022). We use Russian, which BLOOM models have not intentionally seen during pretraining. We collect supervised natural language task data in Russian and finetune the pretrained 7.1 billion parameter BLOOM model to create two variants: (a) BLOOMZ-7.1B-RU, which is finetuned only on the Russian task data, and (b) BLOOMZ-7.1B-xP3RU, which is finetuned on the full xP3 dataset (Muennighoff et al., 2022) with Russian data added to it. We compare the two

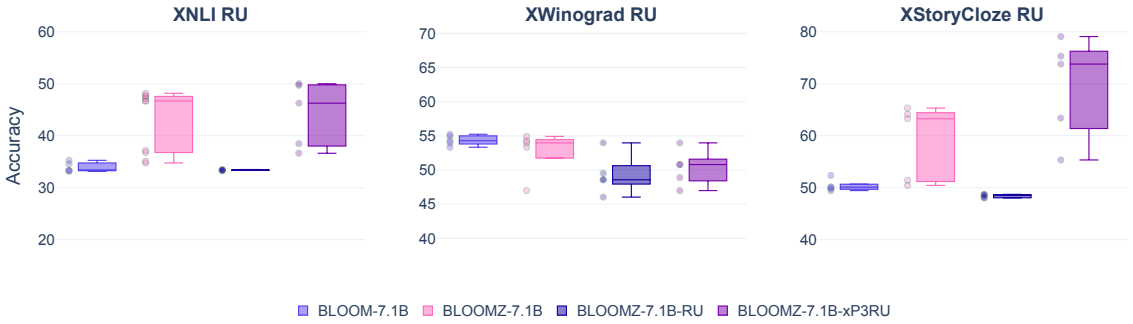

Figure 9: Performance on unseen language tasks in Russian of BLOOMZ variants.

models with BLOOM-7.1B and BLOOMZ-7.1B in Figure 9. We find that finetuning on only Russian (BLOOMZ-7.1B-RU) without the other languages and tasks in the xP3 mixture shows only tiny improvements over the pretrained baseline on XStoryCloze. This is likely due to the lack of diversity in the finetuning of BLOOMZ-7.1B-RU (Chung et al., 2022), as the Russian-only split contains fewer tasks and prompts than the full xP3 dataset. On the other hand, **when adding Russian to the instruction tuning mixture (BLOOMZ-7.1B-xP3RU), the performance of the best prompt improves on XNLI and XStoryCloze.** This means that adding new languages during multitask finetuning can be effective but requires additional diverse tasks in other languages.

5 Conclusion

We compare the compute-performance trade-off of different language adaptation strategies for extending BLOOM of various sizes to new languages. Contrary to previous work, we find that adapter-based strategies best adapt larger BLOOM models for prompting under low-resource settings. We also investigate different language adaptation factors such as the size of language adaptation data and capacity of adapters. Finally, we investigate the relationship between language adaptation and instruction tuning using the BLOOMZ model, where we find including new languages during instruction tuning most effective.

6 Limitations

6.1 Vocabulary and Embedding Adaptation

We do not explore vocabulary and embedding adaptation. Our models used byte-level tokenization, and therefore can handle unseen scripts. However, one can argue that the tokenization of unseen scripts might be suboptimal. For instance, languages with unseen script will require longer post-tokenization, therefore impacting the performance efficiency. Koto et al. (2021) have shown that when adapting to a new domain, LM achieved better performance, despite the fact that the old vocabulary can support the new domain as well. Exploring the quality impact of token adaptation for new languages and new scripts would be very interesting. In parallel, exploring the best way to initialize embeddings of the newly formed tokens is also interesting.

6.2 Parameter-Efficient Finetuning Strategies

We have only considered a limited number of parameter-efficient finetuning strategies (see Section 3.3 and Appendix G) due to computational constraints. Nonetheless, we believe that other strategies such as prompt tuning (Lester et al., 2021; Tu et al., 2022) and ladder side-tuning (Sung et al., 2022) can adapt BLOOM as well as the adapter-based strategies explored in our experimental setting. Recent work has also shown that combining different types of parameter-efficient finetuning methods, including adapters, can lead to better performance (Mao et al., 2022; He et al., 2022). As we recommend adapter-based language adaptation for larger language models, it would be interesting to explore methods that combine adapters for better prompting performance.

6.3 Low-Resource Languages

One limitation of our work is that our set of new languages only covers one truly low-resource language, which is Guarani. As our work shows that 100 million tokens are needed for effective adaptation to prompt in a new language (see Section 4.4), a truly low-resource language usually lacks sufficient unlabeled data for such adaptation (Joshi

et al., 2020). Therefore, we urge the community to study data-efficient methods for adapting large language models to prompt under an extremely low-resource setting.

6.4 Generative Tasks

Since we only cover natural language understanding tasks in our experimental setup, our findings may not generalize to generation tasks such as summarization. Furthermore, language adaptation on monolingual data can lead to catastrophic forgetting of seen languages (see Appendix L); therefore, adapted models are not suitable for multilingual generative tasks that require an understanding of multiple languages such as machine translation. Future work is needed for studying solutions to mitigate catastrophic forgetting.

6.5 Experimental Settings

We used the sequence length of 1024 by mistake (instead of 2048 as described in Scao et al. (2022)) as we followed prior work on adapting BLOOM models to new languages (Yong and Nikoulina, 2022). However, in principle, it should not change the conclusions we draw from our study since none of the evaluation tasks are done on sequences longer than 1024 tokens. Our post-hoc experimental results with the correct sequence length of 2048 (see Appendix N) also align with our results discussed in Section 4.1.

We did not carry out adaptation for the largest BLOOM model and BLOOMZ model with 176 billion parameters due to prohibitive computational costs. We leave them for future work to explore language adaptation for language models with hundreds of billions of parameters.

References

Alham Fikri Aji, Nikolay Bogoychev, Kenneth Heafield, and Rico Sennrich. 2020. In neural machine translation, what does transfer learning transfer? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7701–7710.

Alan Ansell, Edoardo Ponti, Anna Korhonen, and Ivan Vulić. 2022. Composable sparse fine-tuning for cross-lingual transfer. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1778–1796, Dublin, Ireland. Association for Computational Linguistics.

Mikel Artetxe, Shruti Bhosale, Naman Goyal, Todor Mihaylov, Myle Ott, Sam Shleifer, Xi Victoria Lin, Jingfei Du, Srinivasan Iyer, Ramakanth Pasunuru, et al. 2021. Efficient large scale language modeling with mixtures of experts. *arXiv preprint arXiv:2112.10684*.

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020. On the cross-lingual transferability of monolingual representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4623–4637, Online. Association for Computational Linguistics.

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.

Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. 2022. BitFit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1–9, Dublin, Ireland. Association for Computational Linguistics.

Alexandre Berard. 2021. Continual learning in multilingual NMT via language-specific embeddings. In *Proceedings of the Sixth Conference on Machine Translation*, pages 542–565, Online. Association for Computational Linguistics.

Sid Black, Leo Gao, Phil Wang, Connor Leahy, and Stella Biderman. 2021. Gpt-neo: Large scale autoregressive language modeling with mesh-tensorflow. *If you use this software, please cite it using these metadata*, 58.

Sidney Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, Michael Pieler, Usvsn Sai Prashanth, Shivanshu Purohit, Laria Reynolds, Jonathan Tow, Ben Wang, and Samuel Weinbach. 2022. GPT-NeoX-20B: An open-source autoregressive language model. In *Proceedings of BigScience Episode #5 – Workshop on Challenges & Perspectives in Creating Large Language Models*, pages 95–136, virtual+Dublin. Association for Computational Linguistics.

Ethan C. Chau, Lucy H. Lin, and Noah A. Smith. 2020. Parsing with multilingual BERT, a small corpus, and a small treebank. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1324–1334, Online. Association for Computational Linguistics.

Luis Chiruzzo, Santiago Góngora, Aldo Alvarez, Gustavo Giménez-Lugo, Marvin Agüero-Torales, and Yliana Rodríguez. 2022. Jojajovai: A parallel Guarani-Spanish corpus for MT benchmarking. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 2098–2107, Marseille, France. European Language Resources Association.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

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

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium. Association for Computational Linguistics.

Javier De la Rosa and Andrés Fernández. 2022. Zero-shot reading comprehension and reasoning for spanish with bertin gpt-j-6b. *Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2022)*.

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

Philipp Dufter and Hinrich Schütze. 2020. Identifying elements essential for BERT's multilinguality. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4423–4437, Online. Association for Computational Linguistics.

Abteen Ebrahimi and Katharina Kann. 2021. How to adapt your pretrained multilingual model to 1600 languages. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4555–4567, Online. Association for Computational Linguistics.

Abteen Ebrahimi, Manuel Mager, Arturo Oncevay, Vishrav Chaudhary, Luis Chiruzzo, Angela Fan, John Ortega, Ricardo Ramos, Annette Rios, Ivan Vladimir Meza Ruiz, Gustavo Giménez-Lugo, Elisabeth Mager, Graham Neubig, Alexis Palmer, Rolando Coto-Solano, Thang Vu, and Katharina Kann. 2022. AmericasNLI: Evaluating zero-shot natural language understanding of pretrained multilingual models in truly low-resource languages. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6279–6299, Dublin, Ireland. Association for Computational Linguistics.

Fahim Faisal and Antonios Anastasopoulos. 2022. Phylogeny-inspired adaptation of multilingual models to new languages. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 434–452, Online only. Association for Computational Linguistics.

Jonathan Frankle and Michael Carbin. 2019. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations*.

Jinlan Fu, See-Kiong Ng, and Pengfei Liu. 2022. Polyglot prompt: Multilingual multitask promptraining. *arXiv preprint arXiv:2204.14264*.

Yoshinari Fujinuma, Jordan Boyd-Graber, and Katharina Kann. 2022. Match the script, adapt if multilingual: Analyzing the effect of multilingual pretraining on cross-lingual transferability. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1500–1512, Dublin, Ireland. Association for Computational Linguistics.

Philip Gage. 1994. A new algorithm for data compression. *C Users Journal*, 12(2):23–38.

Demi Guo, Alexander Rush, and Yoon Kim. 2021. Parameter-efficient transfer learning with diff pruning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4884–4896, Online. Association for Computational Linguistics.

Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. 2022. Towards a unified view of parameter-efficient transfer learning. In *International Conference on Learning Representations*.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799. PMLR.

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*.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The state and fate of linguistic diversity and inclusion in the NLP world. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293, Online. Association for Computational Linguistics.

Fajri Koto, Jey Han Lau, and Timothy Baldwin. 2021. Indobertweet: A pretrained language model for indonesian twitter with effective domain-specific vocabulary initialization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10660–10668.

Julia Kreutzer, Isaac Caswell, Lisa Wang, Ahsan Wahab, Daan van Esch, Nasanbayar Ulzii-Orshikh, Allahsera Tapo, Nishant Subramani, Artem Sokolov, Claytone Sikasote, Monang Setyawan, Supheakmungkol Sarin, Sokhar Samb, Benoît Sagot, Clara Rivera, Annette Rios, Isabel Papadimitriou, Salomey Osei, Pedro Ortiz Suarez, Iroro Orife, Kelechi Ogueji, Andre Niyongabo Rubungo, Toan Q. Nguyen, Mathias Müller, André Müller, Shamsuddeen Hassan Muhammad, Nanda Muhammad, Ayanda Mnyakeni, Jamshidbek Mirzakhalov, Tapiwanashe Matangira, Colin Leong, Nze Lawson, Sneha Kudugunta, Yacine Jernite, Mathias Jenny, Orhan Firat, Bonaventure F. P. Dossou, Sakhile Dlamini, Nisansa de Silva, Sakine Çabuk Ballı, Stella Biderman, Alessia Battisti, Ahmed Baruwa, Ankur Bapna, Pallavi Baljekar, Israel Abebe Azime, Ayodele Awokoya, Duygu Ataman, Orevaoghene Ahia, Oghenefego Ahia, Sweta Agrawal, and Mofetoluwa Adeyemi. 2022. Quality at a Glance: An Audit of Web-Crawled Multilingual Datasets. *Transactions of the Association for Computational Linguistics*, 10:50–72.

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, Jörg Frohberg, Mario Šaško, Quentin Lhoest, Angelina McMillan-Major, Gérard Dupont, Stella Biderman, Anna Rogers, Loubna Ben allal, Francesco De Toni, Giada Pistilli, Olivier Nguyen, Somaieh Nikpoor, Maraim Masoud, Pierre Colombo, Javier de la Rosa, Paulo Villegas, Tristan Thrush, Shayne Longpre, Sebastian Nagel, Leon Weber, Manuel Romero Muñoz, Jian Zhu, Daniel Van Strien, Zaid Alyafeai, Khalid Almubarak, Vu Minh Chien, Itziar Gonzalez-Dios, Aitor Soroa, Kyle Lo, Manan Dey, Pedro Ortiz Suarez, Aaron Gokaslan, Shamik Bose, David Ifeoluwa Adelani, Long Phan, Hieu Tran, Ian Yu, Suhas Pai, Jenny Chim, Violette Lepercq, Suzana Ilic, Margaret Mitchell, Sasha Luccioni, and Yacine Jernite. 2022. The bigscience ROOTS corpus: A 1.6TB composite multilingual dataset. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Teven Le Scao and Alexander Rush. 2021. How many data points is a prompt worth? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2627–2636, Online. Association for Computational Linguistics.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, Benjamin Newman, Binhang Yuan, Bobby Yan, Ce Zhang, Christian Cosgrove, Christopher D. Manning, Christopher Ré, Diana Acosta-Navas, Drew A. Hudson, Eric Zelikman, Esin Durmus, Faisal Ladhak, Frieda Rong, Hongyu Ren, Huaxiu Yao, Jue Wang, Keshav Santhanam, Laurel Orr, Lucia Zheng, Mert Yuksekgonul, Mirac Suzgun, Nathan Kim, Neel Guha, Niladri Chatterji, Omar Khattab, Peter Henderson, Qian Huang, Ryan Chi, Sang Michael Xie, Shibani Santurkar, Surya Ganguli, Tatsunori Hashimoto, Thomas Icard, Tianyi Zhang, Vishrav Chaudhary, William Wang, Xuechen Li, Yifan Mai, Yuhui Zhang, and Yuta Koreeda. 2022. Holistic evaluation of language models.

Xi Victoria Lin, Todor Mihaylov, Mikel Artetxe, Tianlu Wang, Shuohui Chen, Daniel Simig, Myle Ott, Naman Goyal, Shruti Bhosale, Jingfei Du, Ramakanth Pasunuru, Sam Shleifer, Punit Singh Koura, Vishrav Chaudhary, Brian O'Horo, Jeff Wang, Luke Zettlemoyer, Zornitsa Kozareva, Mona Diab, Veselin Stoyanov, and Xian Li. 2021. Few-shot learning with multilingual language models.

Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin Raffel. 2022. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *arXiv preprint arXiv:2205.05638*.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*.

Yuning Mao, Lambert Mathias, Rui Hou, Amjad Almahairi, Hao Ma, Jiawei Han, Scott Yih, and Madian Khabsa. 2022. UniPELT: A unified framework for parameter-efficient language model tuning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6253–6264, 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*.

Benjamin Muller, Antonios Anastasopoulos, Benoît Sagot, and Djamé Seddah. 2021. When being unseen from mBERT is just the beginning: Handling new languages with multilingual language models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 448–462, Online. Association for Computational Linguistics.

Martin Müller and Florian Laurent. 2022. Cedille: A large autoregressive french language model. *arXiv preprint arXiv:2202.03371*.

Graham Neubig and Junjie Hu. 2018. Rapid adaptation of neural machine translation to new languages. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 875–880, Brussels, Belgium. Association for Computational Linguistics.

NovelAI. 2022. Data efficient language transfer with gpt-j. Accessed: 2023-01-16.

Pedro Javier Ortiz Suárez, Benoît Sagot, and Laurent Romary. 2019. Asynchronous pipelines for processing huge corpora on medium to low resource infrastructures. In *Proceedings of the Workshop on Challenges in the Management of Large Corpora (CMLC-7) 2019. Cardiff, 22nd July 2019*, pages 9 – 16, Mannheim. Leibniz-Institut für Deutsche Sprache.

Sungjoon Park, Jihyung Moon, Sungdong Kim, Won Ik Cho, Ji Yoon Han, Jangwon Park, Chisung Song, Junseong Kim, Youngsook Song, Taehwan Oh, Joohong Lee, Juhyun Oh, Sungwon Lyu, Younghoon Jeong, Inkwon Lee, Sangwoo Seo, Dongjun Lee, Hyunwoo Kim, Myeonghwa Lee, Seongbo Jang, Seungwon Do, Sunkyoung Kim, Kyungtae Lim, Jongwon Lee, Kyumin Park, Jamin Shin, Seonghyun Kim, Lucy Park, Alice Oh, Jung-Woo Ha, and Kyunghyun Cho. 2021. KLUE: Korean language understanding evaluation. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Marinela Parović, Goran Glavaš, Ivan Vulić, and Anna Korhonen. 2022. BAD-X: Bilingual adapters improve zero-shot cross-lingual transfer. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1791–1799, Seattle, United States. Association for Computational Linguistics.

Jonas Pfeiffer, Naman Goyal, Xi Lin, Xian Li, James Cross, Sebastian Riedel, and Mikel Artetxe. 2022. Lifting the curse of multilinguality by pre-training modular transformers. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3479–3495, Seattle, United States. Association for Computational Linguistics.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. 2021a. AdapterFusion: Non-destructive task composition for transfer learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 487–503, Online. Association for Computational Linguistics.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020. MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7654–7673, Online. Association for Computational Linguistics.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2021b. UNKs everywhere: Adapting multilingual language models to new scripts. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10186–10203, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jerin Philip, Alexandre Berard, Matthias Gallé, and Laurent Besacier. 2020. Monolingual adapters for zero-shot neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4465–4470, Online. Association for Computational Linguistics.

Edoardo Maria Ponti, Goran Glavaš, Olga Majewska, Qianchu Liu, Ivan Vulić, and Anna Korhonen. 2020. XCOPA: A multilingual dataset for causal commonsense reasoning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2362–2376, Online. Association for Computational Linguistics.

Ofir Press, Noah Smith, and Mike Lewis. 2022. Train short, test long: Attention with linear biases enables input length extrapolation. In *International Conference on Learning Representations*.

Kunxun Qi, Hai Wan, Jianfeng Du, and Haolan Chen. 2022. Enhancing cross-lingual natural language inference by prompt-learning from cross-lingual templates. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1910–1923, Dublin, Ireland. Association for Computational Linguistics.

Andreas Rücklé, Gregor Geigle, Max Glockner, Tilman Beck, Jonas Pfeiffer, Nils Reimers, and Iryna Gurevych. 2021. AdapterDrop: On the efficiency of adapters in transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7930–7946, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725.

Freda Shi, Mirac Suzgun, Markus Freitag, Xuezhi Wang, Suraj Srivats, Soroush Vosoughi, Hyung Won Chung, Yi Tay, Sebastian Ruder, Denny Zhou, Dipanjan Das, and Jason Wei. 2023. Language models are multilingual chain-of-thought reasoners. In *The Eleventh International Conference on Learning Representations*.

Oleh Shliazhko, Alena Fenogenova, Maria Tikhonova, Vladislav Mikhailov, Anastasia Kozlova, and Tatiana Shavrina. 2022. mgpt: Few-shot learners go multilingual.

Yi-Lin Sung, Jaemin Cho, and Mohit Bansal. 2022. Lst: Ladder side-tuning for parameter and memory efficient transfer learning. *arXiv preprint arXiv:2206.06522*.

Yi-Lin Sung, Varun Nair, and Colin Raffel. 2021. Training neural networks with fixed sparse masks. In *Advances in Neural Information Processing Systems*.

Zeerak Talat, Aurélie Névéol, Stella Biderman, Miruna Clinciu, Manan Dey, Shayne Longpre, Sasha Luccioni, Maraim Masoud, Margaret Mitchell, Dragomir Radev, Shanya Sharma, Arjun Subramonian, Jaesung Tae, Samson Tan, Deepak Tunuguntla, and Oskar Van Der Wal. 2022. You reap what you sow: On the challenges of bias evaluation under multilingual settings. In *Proceedings of BigScience Episode #5 – Workshop on Challenges & Perspectives in Creating Large Language Models*, pages 26–41, virtual+Dublin. Association for Computational Linguistics.

Alexey Tikhonov and Max Ryabinin. 2021. It's All in the Heads: Using Attention Heads as a Baseline for Cross-Lingual Transfer in Commonsense Reasoning. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3534–3546, Online. Association for Computational Linguistics.

Lifu Tu, Caiming Xiong, and Yingbo Zhou. 2022. Prompt-tuning can be much better than fine-tuning on cross-lingual understanding with multilingual language models. *arXiv preprint arXiv:2210.12360*.

Ahmet Üstün, Alexandre Berard, Laurent Besacier, and Matthias Gallé. 2021. Multilingual unsupervised neural machine translation with denoising adapters. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6650–6662, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ben Wang and Aran Komatsuzaki. 2021. Gpt-j-6b: A 6 billion parameter autoregressive language model.

Zihan Wang, Karthikeyan K, Stephen Mayhew, and Dan Roth. 2020. Extending multilingual BERT to low-resource languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2649–2656, Online. Association for Computational Linguistics.

Genta Winata, Shijie Wu, Mayank Kulkarni, Thamar Solorio, and Daniel Preotiuc-Pietro. 2022. Cross-lingual few-shot learning on unseen languages. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 777–791, Online only. Association for Computational Linguistics.

Yinfei Yang, Yuan Zhang, Chris Tar, and Jason Baldridge. 2019. PAWS-X: A cross-lingual adversarial dataset for paraphrase identification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3687–3692, Hong Kong, China. Association for Computational Linguistics.

Zheng-Xin Yong and Vassilina Nikoulina. 2022. Adapting bigscience multilingual model to unseen languages. *arXiv preprint arXiv:2204.04873*.

Rong Zhang, Revanth Gangi Reddy, Md Arafat Sultan, Vittorio Castelli, Anthony Ferritto, Radu Florian, Efsun Sarioglu Kayi, Salim Roukos, Avi Sil, and Todd Ward. 2020. Multi-stage pre-training for low-resource domain adaptation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5461–5468, Online. Association for Computational Linguistics.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Mengjie Zhao and Hinrich Schütze. 2021. Discrete and soft prompting for multilingual models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8547–8555, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Appendix

A Authors' Contributions

Our work extended the language support of the BLOOM model (Scao et al., 2022) that was created under the BigScience project, a year-long initiative to create open-source large multilingual language models in a transparent manner which involves 600 researchers from over 50 countries and 250 institutions. All authors came from the BigScience

multilingual modeling working group, and in the following list, we document our contributions made to this work.

Zheng-Xin Yong led the project, set up training and evaluation pipelines, coordinated resources and experiments, and wrote most of the paper.

Vassilina Nikoulina advised the project.

Zheng-Xin Yong and Vassilina Nikoulina initially conceptualized the project.

Zheng-Xin Yong, Hailey Schoelkopf, and Lintang Sutawika implemented various parameter-efficient finetuning methods.

Zheng-Xin Yong, Hailey Schoelkopf, Alham Fikri Aji, David Ifeoluwa Adelani, Khalid Almubarak, M Saiful Bari, Ahmed Baruwa, Jungo Kasai, and Vassilina Nikoulina performed language adaptation training and prompting evaluation to collect results.

Zheng-Xin Yong and Niklas Muennighoff performed BLOOMZ language adaptation experiments.

Vassilina Nikoulina performed the sentence retrieval experiments.

Zheng-Xin Yong, Hailey Schoelkopf, Niklas Muennighoff, Alham Fikri Aji, David Ifeoluwa Adelani, Khalid Almubarak, Ahmed Baruwa, Jungo Kasai, Genta Indra Winata, Stella Biderman, Edward Raff, Dragomir Radev, and Vassilina Nikoulina contributed to the paper.

B How does Language Independent Representation changes with Model Sizes

In this work we try to establish the connection between the quality of *language-independent* representation a pretrained LM can emerge, and its adaptability to the new language. In order to evaluate the quality of *language-independent* representation we rely on sentence retrieval task (similar to (Dufter and Schütze, 2020; Artetxe and Schwenk, 2019)) computed on FLORES dataset.[4] Sentence retrieval task is to identify closest sentence in English given a representation of the sentence in the new language, which imitates most most popular knowledge transfer scenario, where we have final task data available in English only. In addition to what has been done previously, we compute sentence retrieval accuracy at each layer of the different pretrained models, to better understand where and how the language-independent represetnation emerges. Figure 10 reports the sentence retrieval accuracy for the subset of languages used to train BLOOM model, for different model sizes. We notice that all the models follow very similar pattern: first and last layers of the model show quite low SR accuracy, but intermediate layers are able to achieve almost perfect sentence retrieval accuracy for all model sizes. An exception is a set of very low-resource languages which seem to have very low Sentence Retrieval Accuracy from English. We do not notice any significant between models of different sizes for the languages that have been observed during training.

[4] We take a subset of 200 sentences of the dev set

C Batch Sizes

Figure 11 shows that the batch size of 8 is an optimal batch size considering the performance-compute trade-off. Performance increases quickly when batch size increases to 8 and slowly afterward.

D Composable Sparse-Finetuning

Composable Sparse-Finetuning (C-SFT) is a sparse-finetuning method that finetunes language-specific and task-specific sparse subset of language model's parameters (mask), both of which demonstrates composability (Ansell et al., 2022). Since the authors demonstrate that this method outperforms MAD-X in language adaptation for POS and NER tasks, we also experimented with it on prompting. In our setting, we only finetuned the language-specific mask, and we followed Ansell et al. (2022) by freezing the output embedding and all layer normalization parameters. We reused the same hyperparameters but with an even split of 12,500 steps in both first and second stage of C-SFT. We ran our experiments using the publicly released code `https://github.com/cambridgeltl/composable-sft/tree/6e3ef08cf0fc465d59285e529569387246028538`.

Our preliminary results with smaller BLOOM models show that models adapted by C-SFT are not capable of prompting (see Table 2) even though it improves sentence retrieval score (red ▼ in Figure 14). In addition to the poor prompting performance, C-SFT requires finetuning the entire model and needs twice the GPU RAM memory than continued pretraining for storing a copy of the original model to compute the sparse mask. We found that we can improve prompting performance with longer C-SFT training steps. When we ran 25K

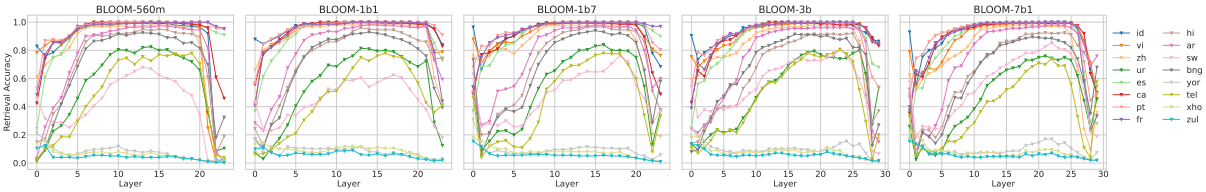

Figure 10: Sentence Retrieval accuracy for known languages for different BLOOM models across layers.

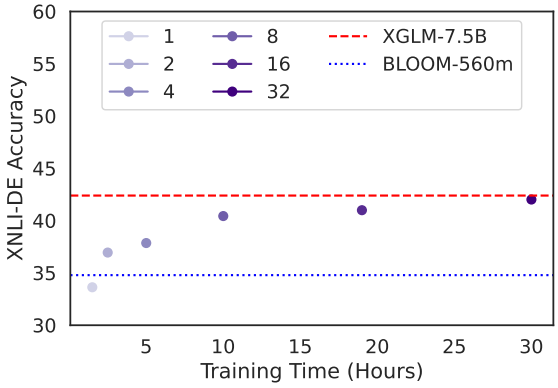

Figure 11: German XNLI prompting performance with the BLOOM-560m model trained with various batch sizes of monolingual language adaptation data.

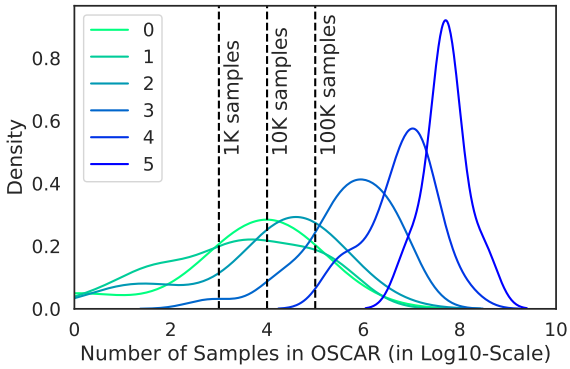

Figure 12: Distribution of language resources on OSCAR (Ortiz Suárez et al., 2019) grouped by the level of resource setting (0 indicates very low-resource, 5 indicates high-resource) according to Joshi et al. (2020).

training steps for both stages of C-SFT, totalling 50K language adaptation steps (instead of 25K total steps), German XNLI prompting performance improved from 33.01% to 35.97%. However, due to computational constraint, we did not run more experiments with C-SFT.

Models	Adapt.	DE	RU	TR
Random	-	33.33%	33.33%	33.33%
BLOOM-560m	-	34.79%	34.11%	33.75%
BLOOM-560m	MAD-X	36.83%	39.86%	36.03%
BLOOM-560m	C-SFT	33.01%	33.05%	33.39%
BLOOM-1b1	-	39.64%	39.62%	33.43%
BLOOM-1b1	MAD-X	42.5%	40.26%	37.64%
BLOOM-1b1	C-SFT	34.93%	33.49%	33.39%

Table 2: XNLI Accuracy for unadapted BLOOM model, MAD-X language adapters, and Composable Sparse-Finetuning (C-SFT).

E Korean PAWS-X

Figure 1 shows that all models perform poorly on the Korean PAWS-X task, where a random classifer baseline scores 50%. Our analysis with English templates shows that XGLM baseline, which is effective at code-mixed prompting setting (Lin et al., 2021), also performs poorly for Korean PAWS-X (see Figure 13). Therefore, we believe that the prompt template is ineffective for Korean PAWS-X task.

F Prompt Templates

We used the same templates proposed by (Lin et al., 2021) for prompting the XGLM model. Table 3 shows the English and translated templates for all the tasks. We did not manage to get Thai templates rendered with pdflatex, but the templates can be found on here for XNLI and here for XCOPA.

G Other Parameter-Efficient Finetuning Strategies

We experimented with various parameter-efficient finetuning strategies for language adaptation, including BitFit (Ben Zaken et al., 2022), $(IA)^3$ (Liu et al., 2022), LoRA (Hu et al., 2022), and FishMask (Guo et al., 2021). We reported the best result from the two sets of hyperparameters, one reported in the original papers proposing the methods and the other reported in Appendix H). On German XNLI task, we found that MAD-X language adapters still yield the best prompting performance (see Table 4).

Tasks	Languages	Templates	Verbalizers
XNLI	EN BG DE EL RU	{PREMISE}, right? [Label], {HYPOTHESIS} {PREMISE}, нали? [Label], {HYPOTHESIS} {PREMISE}, richtig? [Label], {HYPOTHESIS} {PREMISE}, σωστ; [Label], {HYPOTHESIS} {PREMISE}, не так ли? [Label], {HYPOTHESIS}	Yes | No | Also Да | Не | Освен това Ja | Nein | Auch Ναι | χι | Επση Да | Нет | А также
KLUE-NLI	KO	{PREMISE}, 맞지? [Label], {HYPOTHESIS}	예 | 아니요 | 또한
AmericasNLI	GN	{PREMISE}, ¿ajépa? [Label], {HYPOTHESIS}	Heẽ | Nahániri | Ave
PAWS-X	EN DE KO	{SENTENCE 1}, right? [Label], {SENTENCE 2} {SENTENCE 1}, richtig? [Label], {SENTENCE 2} {SENTENCE 1}, 맞죠? [Label], {SENTENCE 2}	Yes | No Ja | Nein 예 | 아니오
XStoryCloze	EN RU	{CONTEXT} [Label]	Identity
XWinograd	EN RU	{CONTEXT} *(with '_' replaced by* [Label])	Identity
XCOPA	EN TR	*cause:* {SENTENCE 1} because [Label] *effect:* {SENTENCE 1} so [Label] *cause:* {SENTENCE 1} çünkü [Label] *effect:* {SENTENCE 1} yani [Label]	Identity

Table 3: Task templates for prompting BLOOM where "[Label]" is replaced with the answer choices in the verbalizers column. *NLI tasks' verbalizers correspond to entailment, contradiction, and neutral respectively, and PAWS-X's corresponds to true and false respectively. Identity verbalizer maps candidate choice to itself in multiple-choice tasks.

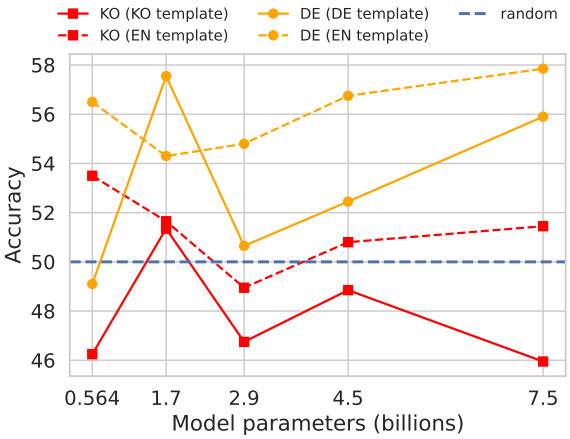

Figure 13: XGLM model's zero-shot prompting performance on German and Korean PAWS-X task with prompt templates in its own language or English language.

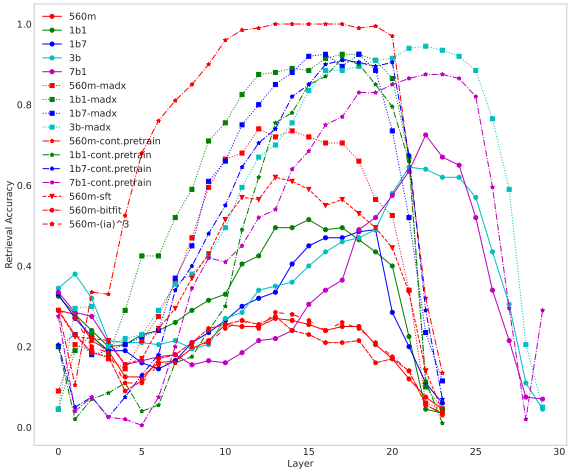

Figure 14: Sentence retrieval accuracy for German with different language adaptation strategies.

Adapt.	Accuracy
No Adaptation	34.79
MAD-X (Bottleneck adapters)	**36.83**
BitFit	33.95
$(IA)^3$	36.31
$(IA)^3$ + invertible adapters	36.47
LoRA	35.79
FishMask	35.59

Table 4: German XNLI prompting performance with the BLOOM-560m model adapted by various parameter-efficient finetuning methods.

H Language Adaptation Experimental Setup Details

We trained for a total of 25,000 steps with a batch size of 8 and sequence length of 1024 on the monolingual corpora of the new language. In other words, the models are trained on around 204 million tokens. We evaluated every 5,000 steps on the perplexity of 1,000 held-out validation samples, and we took the best checkpoint for downstream prompting tasks. We defaulted to using a single RTX 3090 GPU machine for each language adaptation training, unless the model is too large or takes too long to run (for instance, performing continued pretraining for BLOOM with 7.1 billion parameters), which we would use eight A100 GPUs with 40GB RAM for training. We conducted single runs for each language adaptation due to computational constraint.

We performed hyperparameter search on learning rates of {1e-3, 1e-4, 1e-5}, linear and cosine decay, and warm-up ratio of {0, 0.05, 0.1} using the Russian XNLI task and BLOOM-560m and -1b1 models. Table 5 reports the best set of hyperparameters. In general, we found that different sets of hyperparameters caused around 1~2 % small difference in XNLI accuracy. Since our primary goal was to study trends and performance-compute trade-offs for language adaptation strategies, we did not perform extensive hyperparameter search.

Adapt.	LR	Decay	Warm-up Ratio
Continued Pretraining	1e-4	Linear	0
MAD-X	1e-4	Linear	0
$(IA)^3$	1e-4	Linear	0.1

Table 5: Best set of hyperparameters for language adaptation strategies.

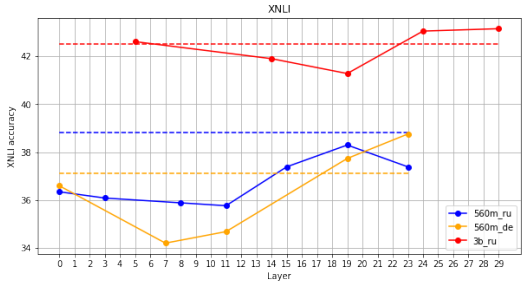

Figure 15: Impact of adapter placement on the quality of model adaptation. Dashed line corresponds to the adapted model with an adapter placed at each layer (referred as mad-x in other experiments).

I Number of Tokens for Language Adaptation Data

We report the number of tokens after preprocessed by BLOOM's BPE tokenizer for all the language adaptation training samples in Table 6.

J Placement of Adapters

We examined how adapters' placement impacts the overall performance. For this, we kept a single adapter at different layers of the model, where we increased the bottleneck size in a way to match the same parameter count of the model with a full set of adapters.[5] Figure 15 compares adapter placement results on XNLI task. We note that layers in the middle benefit less from the language adaptation, and the last layers benefit most from the language adaptation.

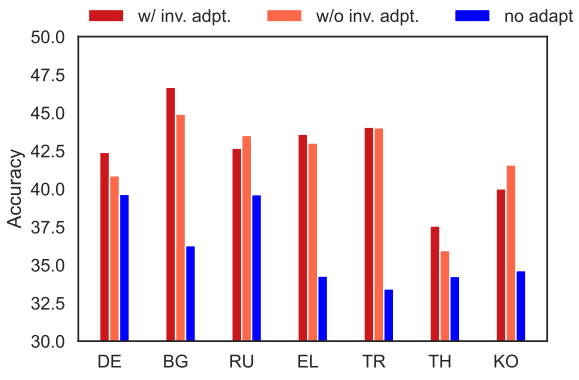

Figure 16: Performance on natural language inference tasks with and without invertible adapters (inv. adpt.) adapting BLOOM's embedding layer. "No adapt" denotes the non-adapted BLOOM model.

[5]For model with 24 layers it would result into 24x larger bottleneck size of the adapter.

K Ablations

Invertible Adapters We analyzed the performance of MAD-X with and without invertible adapters, which are used to adapt the embedding layer of BLOOM-3b, on prompting for natural language inference tasks. Figure 16 shows that invertible adapters only improve performance for German, Bulgarian, and Turkish. This implies that the prompting performance gain from language adaptation mainly results from adapting the Transformer blocks.

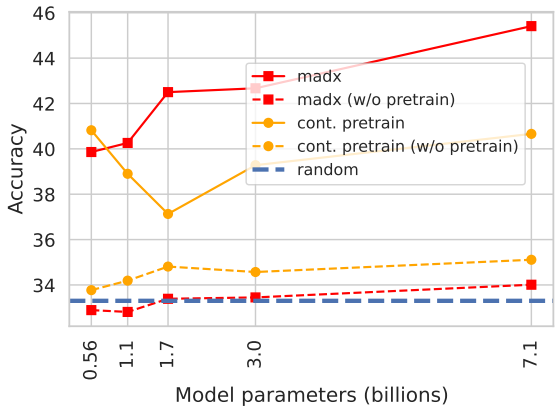

Figure 17: XNLI RU performance with and without pretraining of BLOOM.

Model Pretraining We also performed language adaptation with continued pretraining and MAD-X language adapters on a randomly initialized BLOOM. Figure 17 shows that, without pretraining, the adapted BLOOM model behaves like a random classifier on the XNLI task. Our results confirm that knowledge transfer takes place during language adaptation of pretrained models.

Languages	Number of Samples	Number of Tokens
BG	100K	120M
DE	100K	75M
EL	100K	160M
GN	30K	1M
KO	100K	155M
RU	100K	140M
RU	10K	14M
RU	1K	1.4M
TH	100K	160M
TR	100K	90M
TR	10K	9M
TR	1K	0.9M

Table 6: Number of byte-level tokens in the randomly sampled OSCAR data used for language adaptation. Guarani only has 30K samples, fully taken from Chiruzzo et al.'s (2022) corpora.

L Catastrophic Forgetting

We observe that continued pretraining leads to catastrophic forgetting of seen languages when we evaluated adapted BLOOM on the English XNLI task (Figure 18).

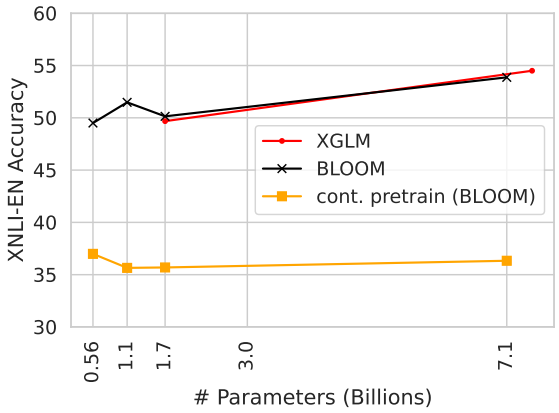

Figure 18: Continued pretraining causes catastrophic forgetting on English, regardless of model sizes.

M Pretraining Languages Existing in BLOOM

Table 7 shows the distribution of natural and programming languages in the ROOTS pretraining data (Scao et al., 2022; Laurençon et al., 2022).

N Post-Hoc Experiments

Sequence Lengths of 2048 We adapted BLOOM-7.1B model for Thai and Greek using with the sequence length of 2048 instead of 1024 and training steps of 12500. We picked these two languages because they have the most number of tokens in the 100K samples (see Table 6), and we halved the training steps to maintain the same number of tokens seen during language adaptation since we doubled the sequence length. The rest of the setup follows Section 3.4. Figure 19 shows that adapters-based strategies still outperform continued-pretraining when we use the sequence length of 2048, which is consistent with our results discussed in Section 4.1.

O Artifacts

For the pretrained models used in our study, BLOOM (Scao et al., 2022) and BLOOMZ models (Muennighoff et al., 2022) are released under the RAIL license, whereas mGPT (Shliazhko et al., 2022) and mT0 (Muennighoff et al., 2022) are re-

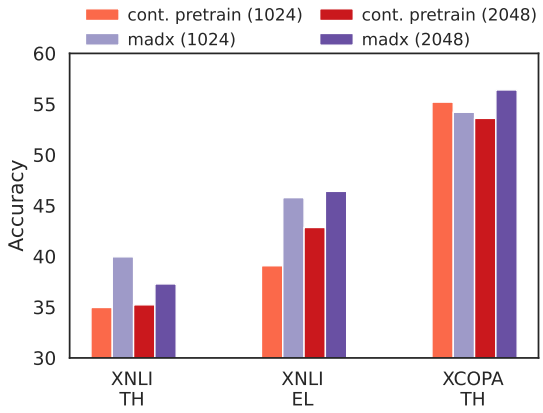

Figure 19: Comparison of prompting performance between sequence lengths of 1024 and 2048 during language adaptation for Thai (TH) and Greek (EL) languages with continued pretraining and MADX adapters.

leased under the Apache 2.0 license. XGLM (Lin et al., 2021) is released under the MIT license.

OSCAR data (Ortiz Suárez et al., 2019), which is used to adapt BLOOM models, are released under the Creative Commons designation CC0 1.0 Universal license. whereas Guarani data (Chiruzzo et al., 2022) are released under the MIT license.

XNLI (Conneau et al., 2018) are released under the Attribution-NonCommercial 4.0 International license, KLUE-NLI (Park et al., 2021) and AmericasNLI (Ebrahimi et al., 2022) under the Attribution-ShareAlike 4.0 International license, XCOPA (Ponti et al., 2020) under the Attribution 4.0 International license, XStoryCloze (Lin et al., 2021) under the MIT license, and PAWS-X (Yang et al., 2019) may be freely used for any purpose.

Language	Proportion (%)
English	30.04
Simplified Chinese	16.2
Traditional Chinese	0.05
French	12.9
Arabic	4.6
Basque	0.15
Catalan	1.1
Indonesian	1.2
Portuguese	4.9
Spanish	10.8
Vietnamese	2.7
Chitumbuka	0.00002
Assamese	0.01
Kikuyu	0.00004
Odia	0.04
Bambara	0.00004
Gujarati	0.04
Akan	0.00007
Marathi	0.05
Xitsonga	0.00007
Punjabi	0.05
Sesotho	0.00007
Kannada	0.06
Chichewa	0.0001
Nepali	0.07
Setswana	0.0002
Telugu	0.09
Northern Sotho	0.0002
Malayalam	0.10
Fon	0.0002
Urdu	0.10
Kirundi	0.0003
Tamil	0.20
Wolof	0.0004
Bengali	0.50
Luganda	0.0004
Lingala	0.0002
Hindi	0.70
chiShona	0.001
isiZulu	0.001
Igbo	0.001
isiXhosa	0.001
Kinyarwanda	0.003
Yoruba	0.006
Swahili	0.02
Code*	10.8

Table 7: Information about the seen languages by BLOOM model.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Yes. Section 6.

☒ A2. Did you discuss any potential risks of your work?
We did not find our work present major risks.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Section 1.1 summarize our core contributions.

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

Section 3.1 (BLOOM), 3.4 (OSCAR), and 3.5 (evaluation tasks)

☑ B1. Did you cite the creators of artifacts you used?
Section 3.1 (BLOOM), 3.4 (OSCAR), and 3.5 (evaluation tasks)

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Appendix O.

☒ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
The existing artifacts used did not elaborate on the intended use, and our efforts in using these artifacts do not present safety risks for the community. We adhere to the license compliance of the existing artifacts. Our created artifacts are simply an extension of the original BLOOM model but for more languages, so we refer to the original paper for the intended use.

☒ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
As we reuse existing datasets, we refer to the original works for such information.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
We explain the new language coverage of our artifacts in section 3.2.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
We explain the number of training samples in section 3.4, and we use the test split of the existing evaluation tasks.

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

C ☑ Did you run computational experiments?

Section 3.

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Figure 3 and Appendix G.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Section 3.4 and Appendix G.

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Results and setup are reported in Figure 1, section 3.4, and Appendix G. It is transparent that we are reporting results from single runs (indicated in setup).

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
We report our parameter settings in section 3 and release the code implementation in the abstract.

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.