Lifting the Curse of Multilinguality by Pre-training Modular Transformers

Jonas Pfeiffer[*1,2,3], **Naman Goyal**[3], **Xi Victoria Lin**[3], **Xian Li**[3], **James Cross**[3], **Sebastian Riedel**[3], **Mikel Artetxe**[3]
[1]New York University, [2]TU Darmstadt,
[3]Meta AI

Abstract

Multilingual pre-trained models are known to suffer from *the curse of multilinguality*, which causes per-language performance to drop as they cover more languages. We address this issue by introducing language-specific modules, which allows us to grow the total capacity of the model, while keeping the total number of trainable parameters per language constant. In contrast with prior work that learns language-specific components post-hoc, we pre-train the modules of our **Cross**-lingual **Mod**ular (**X-MOD**) models from the start. Our experiments on natural language inference, named entity recognition and question answering show that our approach not only mitigates the negative interference between languages, but also enables positive transfer, resulting in improved monolingual and cross-lingual performance. Furthermore, our approach enables adding languages post-hoc with no measurable drop in performance, no longer limiting the model usage to the set of pre-trained languages.

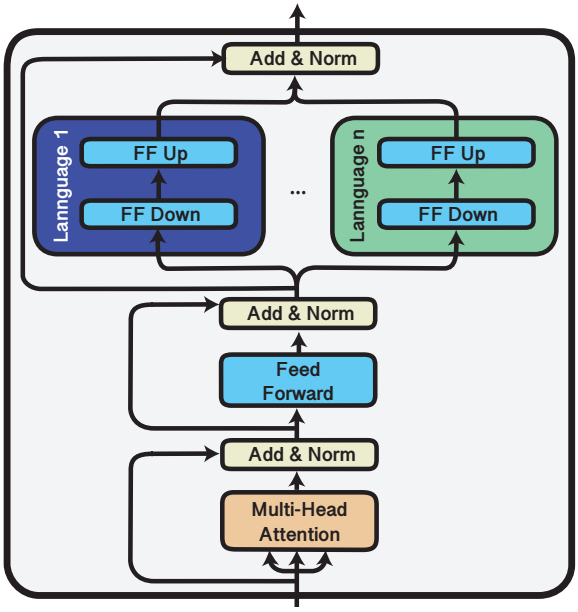


Figure 1: A transformer layer of our proposed modular architecture. The dark blue and green components illustrate the modular layers, which are language specific. The Multi-Head Attention and Feed-Forward components are shared by all languages.

1 Introduction

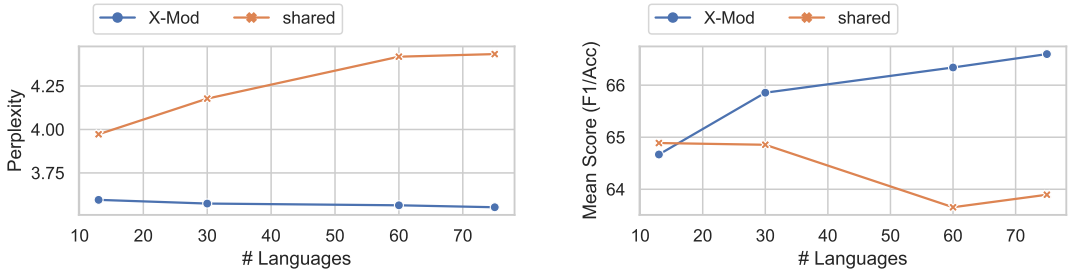
Recent work on multilingual NLP has focused on pre-training transformer-based models (Vaswani et al., 2017) on concatenated corpora of a large number of languages (Devlin et al., 2019; Conneau et al., 2020). These multilingual models have been shown to work surprisingly well in cross-lingual settings, despite the fact that they do not rely on direct cross-lingual supervision (e.g., parallel data or translation dictionaries; Pires et al., 2019; Wu and Dredze, 2019; Artetxe et al., 2020; Hu et al., 2020; K et al., 2020; Rust et al., 2021).

However, recent work has uncovered fundamental limitations of multilingual transformers. Conneau et al. (2020) observe that pre-training a model with a fixed capacity on an increasing amount of languages only improves its cross-lingual performance up to a certain point, after which performance drops can be measured—a phenomenon known as *the curse of multilinguality* (Figure 2). As such, prior work had to find a trade-off between supporting more languages and obtaining better performance on a smaller set of languages.

In this work, we address this problem by introducing language-specific, modular components during pre-training (Figure 1). Our **Cross**-lingual, **Mod**ular (**X-MOD**) language model shares the majority of the transformer parameters between all pre-training languages, while providing each language with individual capacity to learn idiosyncratic information without increasing the total number of trainable parameters per language. While previous adapter-based approaches (Figure 3a) extend pre-trained multilingual language models (LMs) with modular components *after* pre-training, we add modular components *during* pre-training, thereby

[*] Work done while interning at Meta AI.

Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 3479 - 3495



(a) Mean Perplexity.

(b) Mean Performance on XNLI and NER.

Figure 2: Average (a) perplexity and (b) transfer performance on XNLI and NER across pre-trained languages when training on an increasing number of languages. Each model has seen the **same amount of examples** in each language. Lower perplexity and higher downstream score indicate better performance. Refer to Figure 4 for per-task performance, and Appendix A for per-language performance.

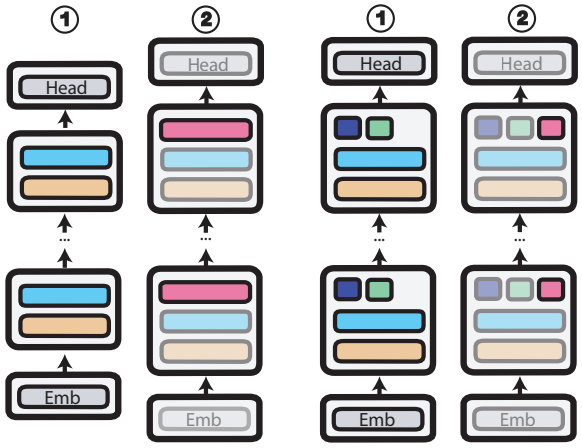
preparing the model to be extended to new languages post-hoc. Our experiments on natural language inference (NLI), named entity recognition (NER), and question answering (QA) demonstrate that our modular architecture not only is effective at mitigating interference between languages, but also achieves positive transfer, resulting in improved monolingual and cross-lingual performance. In addition, we show that X-MOD can be extended to unseen languages, with no measurable drop in performance, by learning its corresponding modules and leaving the shared parameters frozen. All in all, we propose a multilingual architecture that can scale to a large number of languages without any loss in performance, and can be further extended to new languages after pre-training.[1]

2 Background and related work

We provide a background on multilingual and modular language modelling, as well as approaches that extend LMs to new languages.

2.1 Multilingual transformers

Recent LMs (Devlin et al., 2019; Conneau et al., 2020), based on transformer architectures (Vaswani et al., 2017) and pre-trained on massive amounts of multilingual data, have surpassed (static) cross-lingual word embedding spaces (Ruder et al., 2019; Glavas et al., 2019) for cross-lingual transfer in NLP (Pires et al., 2019; Wu and Dredze, 2019; Wu et al., 2020; Hu et al., 2020; K et al., 2020). Transformer-based models are **1)** pre-trained on textual corpora using Masked Language Modelling (MLM). They are then **2)** fine-tuned on labelled data of a downstream task in a *source* language and **3)** directly applied to perform inference in a *target* language (Hu et al., 2020).



(a) Adapter-based

(b) X-MOD

Figure 3: Our proposed architecture in comparison to adapter-based approaches. (a) Previous approaches ① utilize non-modular pre-trained transformer models and ② extend them with modular adapter components. (b) We ① pre-train the transformer with modular units from the get-go, *preparing* the model to be ② extended with additional modular units later on. Yellow and light blue components indicate standard Multi-Head Attention and Feed-Forward layers. The remaining (non-gray) components are bottleneck (modular) units. Grayed-out components are frozen.

2.2 Modular language models

Modular approaches have a long standing history in NLP, preceding pre-trained models (Andreas et al., 2016). They have recently re-gained interest for transformer-based models, where mix-

[1]Code and pre-trained models are available at: https://github.com/pytorch/fairseq/tree/main/examples/xmod.

ture of experts (MoE; Shazeer et al., 2017) approaches have enabled training trillion parameters models in a distributed fashion (Fedus et al., 2021). More recently modular MoE approaches have been shown to improve domain-specific pre-training of LMs (Gururangan et al., 2021). In a similar trend, 'expert' modules have been added to (non-modular) pre-trained LMs post-hoc, predominantly referred to as adapters (Rebuffi et al., 2017, 2018; Houlsby et al., 2019). Next to being extremely parameter (Houlsby et al., 2019; Mahabadi et al., 2021a; He et al., 2022) and training efficient (Pfeiffer et al., 2020a; Rücklé et al., 2021), these modular approaches allow models to be extended to new data settings (Chen et al., 2019; Rücklé et al., 2020), where newly learned knowledge can be combined (Stickland and Murray, 2019; Wang et al., 2021a; Pfeiffer et al., 2021a; Lauscher et al., 2020a; Mahabadi et al., 2021b; Poth et al., 2021), or stacked for combinatory cross-lingual (Pfeiffer et al., 2020b, 2021b; Üstün et al., 2020; Vidoni et al., 2020; Ansell et al., 2021b,a; Wang et al., 2021b) as well as NMT scenarios (Bapna and Firat, 2019; Philip et al., 2020; Chronopoulou et al., 2020; Le et al., 2021; Üstün et al., 2021; Stickland et al., 2021; Garcia et al., 2021).

2.3 Weaknesses, improvements, and extensions of language models

Next to the *curse of multilinguality*, recent works have shown substantially reduced cross-lingual and monolingual abilities of models for low-resource languages with smaller pre-training data (Wu and Dredze, 2020; Hu et al., 2020; Lauscher et al., 2020b; Artetxe et al., 2020; Pfeiffer et al., 2020b, 2021b; Chau et al., 2020b; Ponti et al., 2020).

K et al. (2020); Artetxe et al. (2020) show that a shared vocabulary is not necessary for cross-lingual transfer. Chung et al. (2021) demonstrate that decoupling the input embeddings from the prediction head improves the performance on a number of downstream tasks. Dufter and Schütze (2020) show that the number of parameters and training duration is interlinked with the model's multilingual capability. Chung et al. (2020); Rust et al. (2021) show that the tokenizer plays an important role in the per-language downstream task performance, which Clark et al. (2022); Xue et al. (2022); Tay et al. (2021) take to the extreme by proposing tokenizer-free approaches.

To extend a monolingual LM to other languages, Artetxe et al. (2020) train a new embedding layer with a corresponding target-language tokenizer, while freezing the pre-trained transformer weights. Tran (2020) extend a monolingual model to new languages using bilingual corpora. Wang et al. (2020); Chau et al. (2020a) extend the vocabulary of multilingual models with a small number of target-language tokens, to improve the performance in the target language. Muller et al. (2021) propose a transliteration based approach, Vernikos and Popescu-Belis (2021) propose subword mappings, and Pfeiffer et al. (2020b, 2021b); Vidoni et al. (2020); Ansell et al. (2021b) propose adapter-based approaches to extend multilingual models to unseen languages.

While these approaches achieve considerable performance gains over unseen languages, they are outperformed by standard full fine-tuning methods for seen languages. One can further argue that, as the pre-trained models have already been cursed by multilinguality, the adapter-based approaches build upon sub-optimal parameter initializations.[2] In our work, we consequently aim to **1)** modularize the model from the start to prepare the model to be **2)** extendable to new languages post-hoc.

3 Proposed approach

We propose X-MOD, a modular multilingual architecture that combines shared and language-specific parameters. In contrast to prior work, we pre-train modular models from the get-go. Our models can be extended to new languages after pre-training, and used for cross-lingual transfer learning in downstream tasks.

Architecture. As illustrated in Figure 1, we extend the transformer-based architecture from mBERT (Devlin et al., 2019) and XLM-R (Conneau et al., 2020) by incorporating language-specific modules—bottleneck feed-forward layers—at every transformer layer. We learn a separate module for each language, whereas the attention and feed-forward components are shared. While the total number of parameters of the model grows linearly with the number of languages, the training and inference cost does not increase (as measured in FLOPs), as only the module in the relevant language is used for each input. Inspired by the adapter[3] architecture of Pfeiffer et al. (2021a) we

[2] We investigate this claim further in §6.2.

[3] The term 'adapter' refers to newly introduced layers within a pre-trained (frozen) model. These layers *adapt* the

place our 'modules' after the LayerNorm of the feed-forward transformer block, and the residual connection is placed after the LayerNorm;[4] the LayerNorm before and after the modular component is shared.[5]

Pre-training procedure. Similar to Conneau et al. (2020), we pre-train our model on MLM on combined monolingual corpora in multiple languages. Examples of each language are passed through the shared embedding matrix as well as the multi-head attention and feed-forward components at each layer. As each layer contains a language-specific modular component, the examples are routed through the respective designated modular bottleneck layer. Given that each example only requires access to a single module, modules can be efficiently stored on only a subset of GPUs in distributed training.

Extending to new languages. The modular design of our model allows us to extend it to new languages after pre-training. To that end, we learn new embeddings and adapter modules for the target language through MLM, while the rest of the components are frozen.[6] Consequently, we are able to extend the model to a new language by learning a small number of new parameters, without affecting performance in the set of pre-trained languages. Following Pfeiffer et al. (2021b), we learn a new subword vocabulary for the added languages, and initialize the embeddings of lexically overlapping tokens from the original embedding matrix.

Fine-tuning on downstream tasks. To transfer the models to cross-lingual downstream tasks, we fine-tune the shared weights only on the source language data, while keeping the modular components and the embedding layer frozen. We follow the standard fine-tuning procedure of adding a prediction head on top of the CLS token. We then replace the source language modules (as well as embedding layer for *added* languages) with the target language parameters, passing the text of the target language through the model.[7]

representations of the pre-trained mode; we train these modular components together with the transformer weights, and therefore refer to them as modules.

[4]We find that the residual connection proposed by Pfeiffer et al. (2021a) results in training instabilities when trained together with the transformer weights.

[5]Preliminary results showed that sharing the LayerNorm results in better cross-lingual transfer performance.

[6]Following Artetxe et al. (2020) we train positional embeddings.

[7]We initially also experimented with stacking adapters on top of the language modules similar to Pfeiffer et al. (2020b, 2021b). While this approach is considerably more parameter efficient, we find that fine-tuning all shared weights slightly outperformed the adapter-based approach.

4 Experimental design

We detail the baseline and models (§4.1), and their training (§4.2) and evaluation settings (§4.3).

4.1 Model variants

We pre-train separate models for all combinations along the following axes:

X-MOD vs. SHARED. To evaluate the effectiveness of our X-MOD model, we aim to compare ourselves to a conventional non-modular architecture. However, simply removing the modular component would be unfair, as the number of FLOPs and trainable parameters per language would not be the same—both in terms of pre-training, as well as fine-tuning. Consequently, for our baseline model—where all parameters should be *fully* shared between all languages—we include a single bottleneck layer right after the Feed-Forward component. Effectively, this is the same architecture as our X-MOD model, just with a single module that is shared by all languages. We refer to this as the SHARED model throughout this paper.[8] To extend the SHARED model to unseen languages, we follow Artetxe et al. (2020) and only learn a new embedding layer, freezing the transformer parameters. To fine-tune the SHARED model on a downstream task, we freeze the embedding layer, as well as the (single) module, thereby fine-tuning an equal amount of parameters on the downstream task as the X-MOD model.[9]

13 vs. 30 vs. 60 vs. 75 languages. So as to understand how each approach is affected by the curse of multilinguality, we pre-train the X-MOD and SHARED models on 4 increasing sets of languages. We start with an *initial* set of 13 typologically diverse languages that we evaluate on, and add additional languages for larger sets of 30, 60, and 75 languages. In addition, we keep a set of 7 held-out languages that we extend the pre-trained models to. Table 1 lists the specific languages in each

[8]Extending the **total** number of shared parameters would be unfair, as X-MOD and SHARED would not have the same FLOPs nor the same number of trainable parameters when fine-tuning.

[9]Adapter-based approach such as MAD-X (Pfeiffer et al., 2020b) would be an alternative. However, this would require training on languages twice—once during pre-training, and once when adding adapters—which is not directly comparable to X-MOD. Nonetheless, we report results in §6.2.

Pre-trained languages	13-LANGS	***en***, ***ar***, **fr**, ***hi***, ko, ***ru***, ***th***, ***vi***, ta, id, fi, **sw**, ka
	30-LANGS	13-LANGS + cs, eu, hr, hu, hy, it, lt, ml, mn, ms, pl, ro, si, sk, sq, sv, tl
	60-LANGS	30-LANGS + af, am, be, bn, ca, cy, da, eo, et, fa, ga, gl, gu, ha, is, ku, la, lv, mk, ne, nl, no, ps, pt, sa, sd, sl, so, sr, te
	75-LANGS	60-LANGS + as, br, bs, fy, gd, jv, kn, mg, mr, om, or, pa, su, xh, yi,
Added languages		**bg**, ***de***, ***el***, ***es***, ***tr***, **ur**, ***zh***,

Table 1: **Selection of languages.** We pre-train different models on 4 sets of languages, and further extend them to a set of held-out languages post-hoc. We evaluate on XNLI (languages in **bold**), NER (underlined languages) and XQuAD/MLQA (languages in *italic*). For more details about the language selection, see Appendix C.

group. The selection and split of *initial* as well as *added* languages is motivated by typological and geographical diversity, as well as the availability of downstream task evaluation data.

Controlling for total vs. per-language updates. Conneau et al. (2020) investigated the effect of adding more languages during pre-training, while training on an equal number of update steps. However, increasing the number of languages while keeping the number of updates constant results in the model seeing less data in each individual language. As such, it remains unclear if the curse of multilinguality happens because of negative interference, or simply because the number of updates for each specific language is smaller. So as to understand this, we compare (1) training on an equal number of *update steps* and (2) training on an equal number of *seen examples* per language. We start with the set of 13 languages (Table 1) and train the respective models for 125k update steps. When adding more languages, we compare (1) training models on each set of languages for 125k update steps, and (2) increasing the number of update steps such that the models are trained on the same number of examples in each of the initial 13 languages. For the latter, this amounts to training for 195k, 265k and 269k update steps, respectively.

4.2 Training details

Data and hyperparameters. We sample languages with $\alpha = 0.7$ and train our models with a batch size of 2048 across 64 V100 GPUs on the CC100 dataset (Conneau et al., 2020) using fairseq (Ott et al., 2019). All our models extend the *base* transformer architecture, with 12 layers and 768 dimensions. Modules are implemented with a bottleneck size of 384. The shared transformer weights account for 270M parameters, whereas each individual module accounts for 7M parameters. We train our models with a linear learning rate decay peaking at $7\mathrm{e}{-4}$ during pre-training and $1\mathrm{e}{-4}$ when adding languages.

Vocabulary. As we aim to identify the impact of *modularity* on the curse of multilinguality, we control for consistent tokenization across the different axes. We therefore tokenize using the XLM-R vocabulary for all our pre-training experiments.[10] However, for languages added post-hoc, we learn a *new* SentencePiece tokenizer for each of the target language,[11] as the languages potentially use scripts unseen by the original tokenizer.

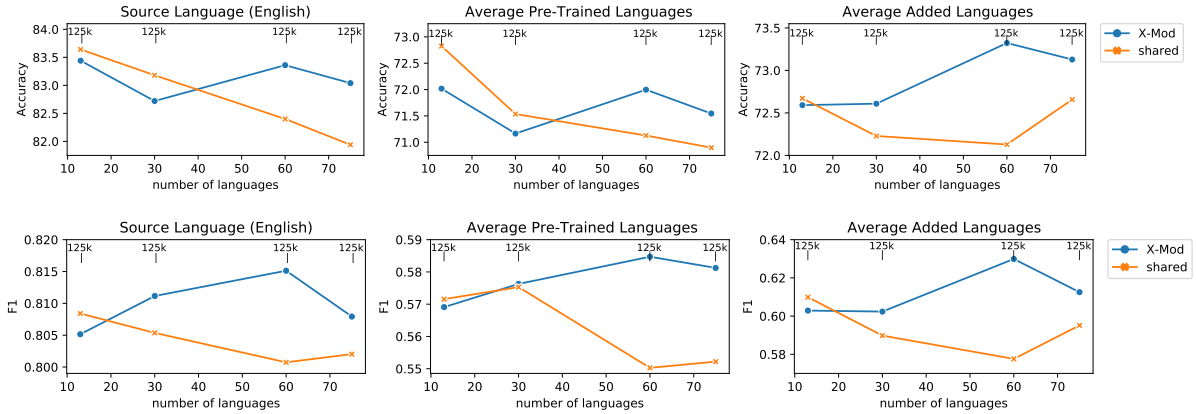
4.3 Evaluation

We conduct experiments on NLI, NER, and QA. In all cases, we fine-tune the model on English and measure the zero-shot transfer performance in other languages. For NLI we train on MultiNLI (Williams et al., 2018) and evaluate on XNLI (Conneau et al., 2018). For QA, we train on SQuAD (Rajpurkar et al., 2016) and evaluate on XQuAD (Artetxe et al., 2020) and MLQA (Lewis et al., 2020). For NER, we use WikiANN (Pan et al., 2017; Rahimi et al., 2019). We experiment with learning rates $1\mathrm{e}{-4}$, $3\mathrm{e}{-4}$, and $5\mathrm{e}{-4}$ and train for 3 or 5 epochs for QA and 5 or 10 epochs for NER and NLI. For NER and NLI we take the hyperparameter setting performing best on the development sets, averaged across the pre-trained languages (Table 1). For SQuAD we take the best performing checkpoint evaluated on the English development set, and report the cross-lingual test set results.[12] All results are averaged across 5 random seed runs.

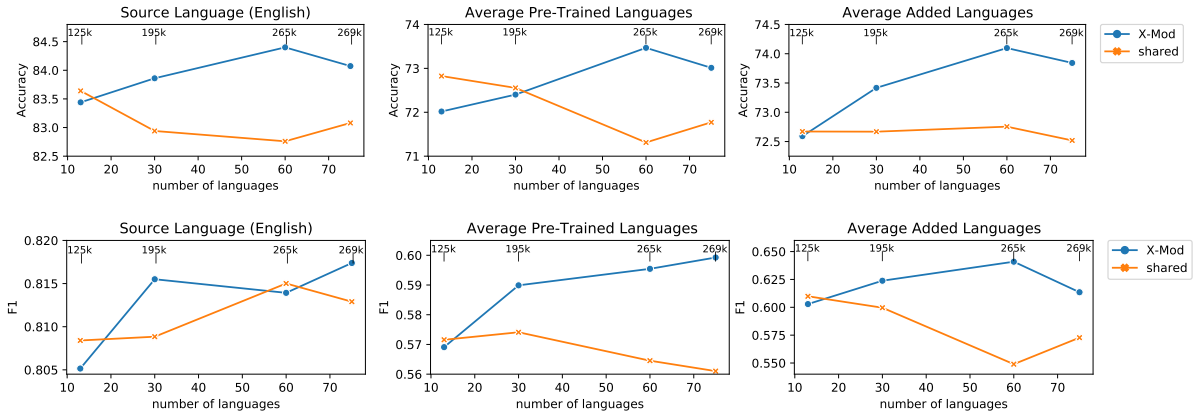
[10] Rust et al. (2021) have previously demonstrated the impact of the multilingual tokenizer on the downstream task performance: languages underrepresented in the sub-word vocabulary exhibit considerable performance drops when compared to vocabularies dedicated to the respective language.

[11] We train the new tokenizers for a vocabulary size of 30k.

[12] In contrast to NER and NLI, the cross-lingual evaluation benchmarks of SQuAD do not provide a development set for each target language on the basis of which the best checkpoint can be selected. Consequently, we select the checkpoint based



(a) All models are trained for 125k update steps. Models trained on **more languages** have seen **less examples** in each language.



(b) Models trained on more languages are trained longer. All models have seen the **same amount of examples** in each language.

Figure 4: Test set results on XNLI (top) and NER (bottom) for models trained on different numbers of languages. *Source Language (English)* only includes scores of the source language. *Average Pre-Trained Languages* includes all evaluation languages that the model was pre-trained on. *Average Added Languages* includes all languages that were added to the model after pre-training. Scores are averaged across all languages and random seeds.

5 Results and discussion

We present results for pre-trained languages in §5.1 and added languages in §5.2.

5.1 Pre-trained languages

In Figure 4 we plot downstream task results of models pre-trained on different amounts of languages. Table 2 reports the individual language performance for the models trained on 60 languages.

The Curse of Multilinguality. Conneau et al. (2020) showed that multilingual LMs trained on *increasing* amounts of languages, while *maintaining* the number of update steps, exhibit drops in downstream task XNLI performance. We reproduce these results, both in terms of language modelling perplexity (Figure 2a),[13] as well as downstream task performance on XNLI *and* NER (Figure 4a). We further find that the curse of multilinguality does not *only* happen *because* the total number of update steps per language decreases, but *also* when all SHARED models are trained on the *same* number of examples per language (Figure 4b). This confirms that fully shared architectures suffer from negative interference.

Lifting the Curse. While for the SHARED model we witness negative interference between languages in terms of perplexity, the X-MOD model is able to *maintain* performance, and even improves for a subset of languages. We observe similar patterns in the downstream task performance: In both our experimental setups—(1) we control for the number of update steps (Figure 4a); (2) we control for the number of per-language seen examples (Figure 4b)—our X-MOD model—in contrast to the SHARED model—is able to maintain, or

on the best performance on the English development set.

[13]For per-language perplexity see Appendix A.

		en	ar	fr	hi	ko	ru	th	vi	ta	id	fi	sw	ka	avg
NER	X-MOD	81.4	**78.9**	**77.2**	**70.1**	**53.0**	**59.1**	2.8	**66.2**	51.1	50.5	**78.6**	**73.4**	**67.3**	**62.8**
	SHARED	**81.5**	74.1	74.7	64.4	46.0	58.3	**4.0**	63.7	**52.5**	**51.5**	74.4	57.2	61.5	58.8
XNLI	X-MOD	**84.4**	**71.2**	**77.6**	**68.3**	-	**74.1**	**71.7**	**73.4**	-	-	-	**66.9**	-	**73.5**
	SHARED	82.8	69.2	75.6	66.6	-	73.2	68.5	72.5	-	-	-	62.1	-	72.5
XQuAD	X-MOD	**85.1**	**68.1**	-	**67.5**	-	**75.0**	**66.3**	**74.9**	-	-	-	-	-	**72.8**
	SHARED	83.8	64.6	-	65.8	-	72.7	63.0	72.6	-	-	-	-	-	70.4
MLQA	X-MOD	**80.1**	**58.6**	-	**60.7**	-	-	-	**67.5**	-	-	-	-	-	**66.7**
	SHARED	79.6	53.6	-	58.7	-	-	-	64.9	-	-	-	-	-	64.2

Table 2: Pre-trained language results for the modular and shared model variants, pre-trained on the set of 60 languages for 265k update steps. For NER and MLQA we report F_1, for XNLI *accuracy* scores. Scores are averaged across all 5 random seeds of the best hyperparameter setting, evaluated on the development set.

		bg	de	el	es	tr	ur	zh	avg
NER	X-MOD	**77.6**	**75.1**	**75.2**	**71.9**	**72.6**	**54.7**	**21.6**	**64.1**
	SHARED	74.9	66.3	69.6	49.1	64.8	50.4	9.2	54.9
XNLI	X-MOD	**77.4**	**75.4**	**76.2**	**78.5**	**72.4**	**64.9**	**73.8**	**74.1**
	SHARED	76.3	74.1	74.9	77.3	71.0	64.3	71.4	72.8
MLQA	X-MOD	-	**63.8**	-	**68.6**	-	-	**61.7**	**64.8**
	SHARED	-	58.9	-	66.7	-	-	56.5	60.7

Table 3: Results for added languages, for models pre-trained on the set of 60 languages for 265k update steps. We report F_1 and *accuracy* scores which are averaged across all 5 random seeds of the best hyperparameter setting on the development set.

even outperform model variants trained on less languages. These results demonstrate that the added per-language capacity is sufficient for the model to adequately represent all languages.

Surprisingly, X-MOD not only maintains performance, but actually slightly improves while we increase the number of languages we pre-train on. This is even the case for settings where the model sees *less* examples in the target language. This suggests that increasing the language diversity can have a positive impact on the model's cross-lingual representation capability.

X-MOD vs SHARED. Overall, the X-MOD model pre-trained on 60 languages achieves the best cross-lingual performance.[14] Our results on XNLI, NER, MLQA, and XQuAD in Table 2 demonstrate consistent performance gains over the SHARED model for every task and across (almost) all high- as well as low-resource languages.

[14]We find that the X-MOD model trained on 75 languages is less stable than the versions trained on less languages. We think that this can be attributed to the 15 added languages being extremely low resource—we only train for an additional 4k update steps—resulting in the respective randomly initialized modules being updated very infrequently. This variance could potentially be mitigated by training for longer.

5.2 Extending to unseen languages

We further evaluate the cross-lingual performance of languages added in the second step; (1) on the architectural side—comparing the SHARED with the X-MOD modelling variant—and (2) by comparing the performance when *pre-training* on the language, vs. when *adding* the language post-hoc.

Modular vs Shared. We evaluate if the additional per-language capacity improves the extendability of the X-MOD model. On the right in Figure 4a we plot the results for added languages on XNLI (top) and NER (bottom). Similarly, we plot the results for the models where we control for the number of seen examples per target language in Figure 4b. We find that the X-MOD model consistently outperforms the SHARED model, with a peak performance when pre-training on 60 languages, demonstrating that the language specific capacity is beneficial for adding new languages post-hoc. We report results for the 60 language versions in Table 3, demonstrating the consistent advantage of the X-MOD over the SHARED model.

Pre-training vs Adding Languages. To evaluate if there is a measurable difference on downstream performance for languages that we *pre-train* on vs. those we *add post-hoc*, we train 2 models on *different* initial sets of languages, adding the respectively missing ones in the second step. So as to understand if the typological similarity of languages has impact on the downstream task performance, we split the *initial* and *added* languages (Table 1) of our previous experiments into two parts. The *first* split consists of languages where the model was pre-trained on at least one language of the same language family (e.g. English vs. German). The *second* split consists of languages that are part of a **unique** language family, i.e. the model was **not**

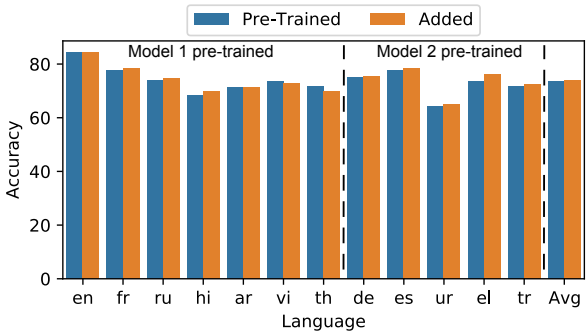


Figure 5: XNLI test set accuracy of X-MOD models pre-trained on different languages in comparison to those added post-hoc (Table 4).

Language	iso	Family	Script	Model 1	Model 2
English	en	IE: Germanic	Latin	pre-train	add
German	de	IE: Germanic	Latin	add	pre-train
French	fr	IE: Romance	Latin	pre-train	add
Spanish	es	IE: Romance	Latin	add	pre-train
Russian	ru	IE: Slavic	Cyrillic	pre-train	add
Ukranian	uk	IE: Slavic	Cyrillic	add	pre-train
Hindi	hi	IE: Iranian	Devanagari	pre-train	add
Urdu	ur	IE: Iranian	Arabic	add	pre-train
Arabic	ar	Afro-Asiatic	Arabic	pre-train	add
Hebrew	he	Afro-Asiatic	Hebrew	add	pre-train
Vietnamese	vi	Austro-Asiatic	Latin	pre-train	add
Thai	th	Kra-Dai	Thai	pre-train	add
Korean	ko	Koreanic	Korean	pre-train	add
Japanese	ja	Japonic	Japanese	add	pre-train
Greek	el	IE: Hellenic	Greek	add	pre-train
Turkish	tr	Turkic	Latin	add	pre-train

Table 4: Selection of 2 sets of languages that we either pre-train on, or add post-hoc. The last 6 languages in the list are part of language families which are *unique* in the total list of languages we pre-train on (Table 1), i.e. none of our models was pre-trained on a language of the same family.

pre-trained on a language of the same family (Table 4). Consequently, we pre-train two models on two sets of languages, adding the respective other set post-hoc.[15]

Our XNLI results (Figure 5) demonstrate that the per-language performance is on par when pre-training vs. when adding the language post-hoc.[16] We also find that the family does not have a measurable effect on the performance of the language. Our results therefore suggest that it is sufficient to train X-MOD on only a subset of languages for which sufficient pre-training data exists. Essentially, X-MOD has the potential to cover all languages of the world, as the model has the capability to be adapted to new languages post-hoc.

[15]In previous experiments, the modular model trained on 60 languages achieved the best performance. Therefore, the models in these experiments are also trained on 60 languages. Both models are trained on the same additional languages, i.e. the 60-LANGS of Table 1, where only the 13-LANGS differ.

[16]The models have seen an equal amount of examples in the respective languages in each case.

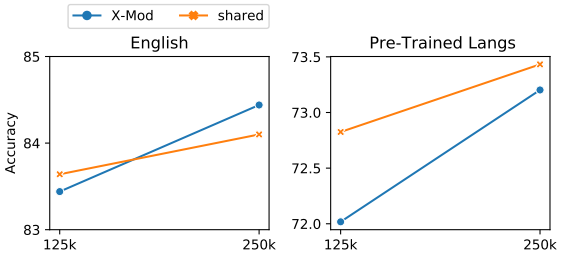


Figure 6: Results on XNLI when when pre-training on 13 languages for 125k and 250k update steps.

6 Further analysis

We further analyze the impact of the number of update steps on X-MOD (§6.1) and compare our method to adapter-based approaches (§6.2).

6.1 The importance of update steps

In Figure 4 we have witnessed a slight edge of the SHARED model over the X-MOD model, when training on only 13 languages and only training for 125k update steps. Dufter and Schütze (2020) found that it requires a large number of update steps for a model pre-trained on multiple languages to become multilingual; with the added per-language capacity we hypothesize that update steps also play an important role for modular models. We compare the downstream task performance of models pre-trained on 13 languages, when training for 125k with 250k update steps in Figure 6. When training for longer we find that the X-MOD model begins to outperforms the SHARED model in the source language, while almost closing the gap in the cross-lingual setting. This supports the hypothesis that the X-MOD model requires more update steps when training only on a small number of languages, in order for modularity to "kick-in".

6.2 X-MOD vs. Adapters

As illustrated in Figure 3, from an architecture perspective X-MOD is similar to previously proposed multilingual Adapter-based methods (MAD-X; Pfeiffer et al., 2020b). MAD-X utilizes a pre-trained massively multilingual transformer-based model and fine-tunes newly introduced adapter weights on languages the model has seen during pre-training, and ones the model has not been

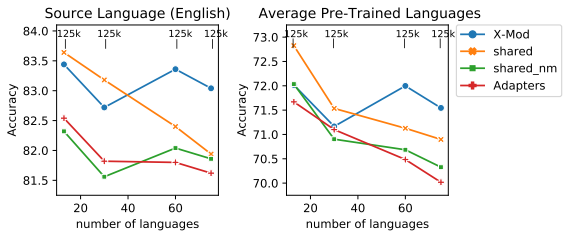


Figure 7: Comparison on XNLI of X-MOD and *shared* models with an Adapter baseline, all models are pre-trained for 125k update steps.

trained on. For a fair comparison in terms of *seen examples* and *number of update steps* we train a transformer model without module components (*shared_nm*) for 100k update steps on the respective languages (Table 1). We subsequently train adapters on each of the target languages for another 25k update steps.[17] We report results in comparison to X-MOD in Figure 7, here results for *shared_nm* are for a model that was trained for 125k update steps to instantiate a fair comparison.

Our results demonstrate that the additional capacity of adapters added *after* pre-training is not able to mitigate the curse of multilinguality which has already had a catastrophic impact on the shared transformer weights; the performance of the adapters strongly correlates with the performance of the corresponding fully shared model *shared_nm*. Consequently, adding language-specific capacity *during* pre-training is important, as the curse of multilinguality cannot be lifted post-hoc.

7 Conclusions

In this paper, we have evaluated the effectiveness of modular multilingual language modelling across multiple axes. We have demonstrated that by providing additional per-language capacity, while maintaining the total number of trainable parameters per language, we are not only able to mitigate negative interference between languages, but additionally achieve positive transfer. Our results suggest that it is sufficient to train our proposed X-MOD model only on a subset of languages for which sufficient amounts of textual data is available. Unseen languages can be added post-hoc, with no measurable drop in performance on XNLI. By *pre-training* the model in a modular fashion, we thus mitigate negative interference of idiosyncratic information, while simultaneously preparing the model to be extendable to unseen languages.

While in this work we have simulated language adding scenarios with a held out set of languages, in future work we aim to evaluate the performance on truly low-resource languages such as MasakhaNER (Adelani et al., 2021) and AmericasNLI (Ebrahimi et al., 2021). We further aim to evaluate the cross-lingual transfer performance from typologically more diverse source languages, besides English.

[17] We follow Pfeiffer et al. (2020b) and train adapter weights with a learning rate of 0.0001. While they have found that cross-lingual transfer performance of adapters converges at ∼20k update-steps, we would like to stress that our experimental setup is only **one** of multiple different valid versions. A more thorough investigation to find the optimal number of update steps for pre-training and subsequent adapter training is necessary, which was out of scope for this work.

Acknowledgments

We thank Samuel Broscheit for insightful feedback and suggestions on a draft of this paper, as well as the ARR reviewers and meta-reviewers for their valuable comments.

References

David Ifeoluwa Adelani, Jade Z. Abbott, Graham Neubig, Daniel D'souza, Julia Kreutzer, Constantine Lignos, Chester Palen-Michel, Happy Buzaaba, Shruti Rijhwani, Sebastian Ruder, Stephen Mayhew, Israel Abebe Azime, Shamsuddeen Hassan Muhammad, Chris Chinenye Emezue, Joyce Nakatumba-Nabende, Perez Ogayo, Aremu Anuoluwapo, Catherine Gitau, Derguene Mbaye, Jesujoba O. Alabi, Seid Muhie Yimam, Tajuddeen Gwadabe, Ignatius Ezeani, Rubungo Andre Niyongabo, Jonathan Mukiibi, Verrah Otiende, Iroro Orife, Davis David, Samba Ngom, Tosin P. Adewumi, Paul Rayson, Mofetoluwa Adeyemi, Gerald Muriuki, Emmanuel Anebi, Chiamaka Chukwuneke, Nkiruka Odu, Eric Peter Wairagala, Samuel Oyerinde, Clemencia Siro, Tobius Saul Bateesa, Temilola Oloyede, Yvonne Wambui, Victor Akinode, Deborah Nabagereka, Maurice Katusiime, Ayodele Awokoya, Mouhamadane Mboup, Dibora Gebreyohannes, Henok Tilaye, Kelechi Nwaike, Degaga Wolde, Abdoulaye Faye, Blessing Sibanda, Orevaoghene Ahia, Bonaventure F. P. Dossou, Kelechi Ogueji, Thierno Ibrahima Diop, Abdoulaye Diallo, Adewale Akinfaderin, Tendai Marengereke, and Salomey Osei. 2021. MasakhaNER: Named Entity Recognition for African Languages. In *Transactions of the Association for Computational Linguistics 2021*.

Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. 2016. Learning to compose neural networks for question answering. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics:*

Human Language Technologies, San Diego California, USA, June 12-17, 2016, pages 1545–1554. The Association for Computational Linguistics.

Alan Ansell, Edoardo Maria Ponti, Anna Korhonen, and Ivan Vulic. 2021a. Composable sparse fine-tuning for cross-lingual transfer. *arXiv preprint*.

Alan Ansell, Edoardo Maria Ponti, Jonas Pfeiffer, Sebastian Ruder, Goran Glavaš, Ivan Vulić, and Anna Korhonen. 2021b. MAD-G: Multilingual adapter generation for efficient cross-lingual transfer. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4762–4781, Punta Cana, Dominican Republic. Association for Computational Linguistics.

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

Ankur Bapna and Orhan Firat. 2019. Simple, scalable adaptation for neural machine translation. 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 2019, Hong Kong, China, November 3-7, 2019*, pages 1538–1548. Association for Computational Linguistics.

Ethan C. Chau, Lucy H. Lin, and Noah A. Smith. 2020a. 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.

Ethan C. Chau, Lucy H. Lin, and Noah A. Smith. 2020b. Parsing with multilingual bert, a small treebank, and a small corpus. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, EMNLP 2020, Online Event, 16-20 November 2020*, pages 1324–1334.

Vincent S. Chen, Sen Wu, Alexander J. Ratner, Jen Weng, and Christopher Ré. 2019. Slice-based learning: A programming model for residual learning in critical data slices. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 9392–9402.

Alexandra Chronopoulou, Dario Stojanovski, and Alexander Fraser. 2020. Reusing a Pretrained Language Model on Languages with Limited Corpora for Unsupervised NMT. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2703–2711, Online. Association for Computational Linguistics.

Hyung Won Chung, Thibault Févry, Henry Tsai, Melvin Johnson, and Sebastian Ruder. 2021. Rethinking embedding coupling in pre-trained language models. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Hyung Won Chung, Dan Garrette, Kiat Chuan Tan, and Jason Riesa. 2020. Improving multilingual models with language-clustered vocabularies. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 4536–4546. Association for Computational Linguistics.

Jonathan H. Clark, Dan Garrette, Iulia Turc, and John Wieting. 2022. CANINE: pre-training an efficient tokenization-free encoder for language representation. *Transactions of the Association for Computational Linguistics*, 10.

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 Conference of the Association for Computational Linguistics, ACL 2020, Virtual Conference, July 6-8, 2020*, pages 8440–8451.

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.

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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186.

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, Manuel Mager, Arturo Oncevay, Vishrav Chaudhary, Luis Chiruzzo, Angela Fan, John Ortega, Ricardo Ramos, Annette Rios, Ivan Vladimir, Gustavo A. Giménez-Lugo, Elisabeth Mager, Graham Neubig, Alexis Palmer, Rolando A. Coto Solano, Ngoc Thang Vu, and Katharina Kann. 2021. AmericasNLI: Evaluating Zero-shot Natural Language Understanding of Pretrained Multilingual Models in Truly Low-resource Languages. *arXiv preprint*.

William Fedus, Barret Zoph, and Noam Shazeer. 2021. Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity. *arXiv preprint*.

Xavier Garcia, Noah Constant, Ankur Parikh, and Orhan Firat. 2021. Towards continual learning for multilingual machine translation via vocabulary substitution. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1184–1192, Online. Association for Computational Linguistics.

Goran Glavas, Robert Litschko, Sebastian Ruder, and Ivan Vulic. 2019. How to (properly) evaluate cross-lingual word embeddings: On strong baselines, comparative analyses, and some misconceptions. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 710–721.

Suchin Gururangan, Mike Lewis, Ari Holtzman, Noah A. Smith, and Luke Zettlemoyer. 2021. Demix layers: Disentangling domains for modular language modeling. *arXiv preprint*.

Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. 2022. Towards a unified view of parameter-efficient transfer learning. In *10th International Conference on Learning Representations, ICLR 2022, Virtual Conference, April 25 - 29, 2022*.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzkebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, pages 2790–2799.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. XTREME: A massively multilingual multi-task benchmark for evaluating cross-lingual generalization. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 12-18 July 2020, Virtual Conference*.

Karthikeyan K, Zihan Wang, Stephen Mayhew, and Dan Roth. 2020. Cross-lingual ability of multilingual BERT: an empirical study. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*.

Anne Lauscher, Olga Majewska, Leonardo F. R. Ribeiro, Iryna Gurevych, Nikolai Rozanov, and Goran Glavaš. 2020a. Common sense or world knowledge? investigating adapter-based knowledge injection into pretrained transformers. In *Proceedings of Deep Learning Inside Out (DeeLIO): The First Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 43–49, Online. Association for Computational Linguistics.

Anne Lauscher, Vinit Ravishankar, Ivan Vulić, and Goran Glavaš. 2020b. From zero to hero: On the limitations of zero-shot language transfer with multilingual Transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4483–4499, Online.

Hang Le, Juan Miguel Pino, Changhan Wang, Jiatao Gu, Didier Schwab, and Laurent Besacier. 2021. Lightweight adapter tuning for multilingual speech translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 2: Short Papers), Virtual Event, August 1-6, 2021*, pages 817–824. Association for Computational Linguistics.

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

Rabeeh Karimi Mahabadi, James Henderson, and Sebastian Ruder. 2021a. Compacter: Efficient low-rank hypercomplex adapter layers. *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021*.

Rabeeh Karimi Mahabadi, Sebastian Ruder, Mostafa Dehghani, and James Henderson. 2021b. Parameter-efficient multi-task fine-tuning for transformers via shared hypernetworks. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 565–576. Association for Computational Linguistics.

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.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of*

the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations), pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Xiaoman Pan, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, and Heng Ji. 2017. Cross-lingual name tagging and linking for 282 languages. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1946–1958.

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, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. 2020a. AdapterHub: A Framework for Adapting Transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (System Demonstrations), EMNLP 2020, Virtual Conference, 2020*.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020b. 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, EMNLP 2021, Online, November , 2021*.

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.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual bert? In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4996–5001.

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.

Clifton Poth, Jonas Pfeiffer, Andreas Rücklé, and Iryna Gurevych. 2021. What to pre-train on? efficient intermediate task selection. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 10585–10605. Association for Computational Linguistics.

Afshin Rahimi, Yuan Li, and Trevor Cohn. 2019. Massively multilingual transfer for NER. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 151–164.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 2383–2392.

Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. 2017. Learning multiple visual domains with residual adapters. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 506–516.

Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. 2018. Efficient parametrization of multi-domain deep neural networks. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 8119–8127.

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, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 7930–7946. Association for Computational Linguistics.

Andreas Rücklé, Jonas Pfeiffer, and Iryna Gurevych. 2020. Multicqa: Zero-shot transfer of self-supervised text matching models on a massive scale. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 2471–2486. Association for Computational Linguistics.

Sebastian Ruder, Ivan Vulić, and Anders Søgaard. 2019. A survey of cross-lingual embedding models. *Journal of Artificial Intelligence Research*, 65:569–631.

Phillip Rust, Jonas Pfeiffer, Ivan Vulić, Sebastian Ruder, and Iryna Gurevych. 2021. How good is your tokenizer? on the monolingual performance of multilingual language models. 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 3118–3135, Online. Association for Computational Linguistics.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc V. Le, Geoffrey E. Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Asa Cooper Stickland, Alexandre Berard, and Vassilina Nikoulina. 2021. Multilingual domain adaptation for NMT: decoupling language and domain information with adapters. In *Proceedings of the Sixth Conference on Machine Translation, WMT@EMNLP 2021, Online Event, November 10-11, 2021*, pages 578–598. Association for Computational Linguistics.

Asa Cooper Stickland and Iain Murray. 2019. BERT and pals: Projected attention layers for efficient adaptation in multi-task learning. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 5986–5995. PMLR.

Yi Tay, Vinh Q. Tran, Sebastian Ruder, Jai Prakash Gupta, Hyung Won Chung, Dara Bahri, Zhen Qin, Simon Baumgartner, Cong Yu, and Donald Metzler. 2021. Charformer: Fast character transformers via gradient-based subword tokenization. *arXiv preprint*.

Ke M. Tran. 2020. From english to foreign languages: Transferring pre-trained language models. *arXiv preprint*.

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, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 6650–6662. Association for Computational Linguistics.

Ahmet Üstün, Arianna Bisazza, Gosse Bouma, and Gertjan van Noord. 2020. UDapter: Language adaptation for truly Universal Dependency parsing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2302–2315, Online. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 5998–6008.

Giorgos Vernikos and Andrei Popescu-Belis. 2021. Subword mapping and anchoring across languages. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2633–2647, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Marko Vidoni, Ivan Vulić, and Goran Glavaš. 2020. Orthogonal language and task adapters in zero-shot cross-lingual transfer. In *arXiv preprint*.

Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing Huang, Jianshu Ji, Guihong Cao, Daxin Jiang, and Ming Zhou. 2021a. K-adapter: Infusing knowledge into pre-trained models with adapters. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 1405–1418. Association for Computational Linguistics.

Xinyi Wang, Yulia Tsvetkov, Sebastian Ruder, and Graham Neubig. 2021b. Efficient test time adapter ensembling for low-resource language varieties. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 730–737, Punta Cana, Dominican Republic. Association for Computational Linguistics.

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.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Shijie Wu, Alexis Conneau, Haoran Li, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Emerging cross-lingual structure in pretrained language models. In *Proceedings of the 58th Conference of the Association for Computational Linguistics, ACL 2020, Virtual Conference, July 6-8, 2020*, pages 6022–6034.

Shijie Wu and Mark Dredze. 2019. Beto, bentz, becas: The surprising cross-lingual effectiveness of BERT. 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

833–844, Hong Kong, China. Association for Computational Linguistics.

Shijie Wu and Mark Dredze. 2020. Are all languages created equal in multilingual BERT? In *Proceedings of the 5th Workshop on Representation Learning for NLP*, pages 120–130, Online. Association for Computational Linguistics.

Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel. 2022. Byt5: Towards a token-free future with pre-trained byte-to-byte models. *Transactions of the Association for Computational Linguistics 2022*.

A Additional results

We report MLQA and XQuAD results on pre-trained languages in Tables 5 and 6, respectively, and MLQA results on added languages in Table 7. Table 8 report NER results on more languages.

Figures 9, 10 and 11 report per-language results as we increase the amount of languages on language modeling perplexity, XNLI and NER, respectively.

B Intermediate checkpoints

Our results in §6.1 suggest that, when the number of languages is small, X-MOD becomes more competitive with SHARED as the number of training steps increases. So as to understand if this behavior also holds for models covering more languages, we evaluate intermediate checkpoints for the 60-LANG model on XNLI. As shown in Figure 8, we find that the X-MOD model continuously outperforms the SHARED model. This suggests that the SHARED model immediately suffers from negative interference between languages, while the added, language-specific components of the X-MOD model are able to mitigate the curse of multilinguality, resulting in considerable performance gains at all evaluated checkpoints.

C Language selection

We provide more details about our selection of languages in Table 9.

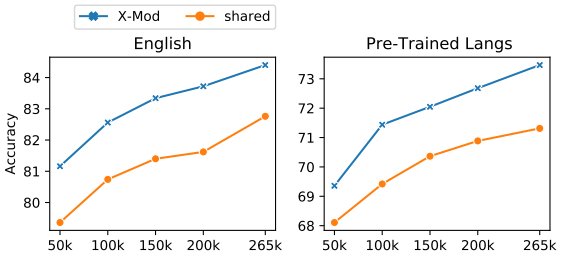


Figure 8: Results on XNLI using intermediate checkpoints of the models trained on 60 languages.

	en F1 / EM	ar F1 / EM	hi F1 / EM	vi F1 / EM	*avg* F1 / EM
X-MOD	**80.1 / 66.9**	**58.6 / 38.9**	**60.7 / 42.4**	**67.5 / 46.1**	**66.7 / 48.6**
SHARED	79.6 / 66.5	53.6 / 33.9	58.7 / 40.4	64.9 / 43.8	64.2 / 46.2

Table 5: Average F_1 and Exact Match results for **pre-trained languages**, on the test set of **MLQA** for the X-MOD and SHARED model variants, pre-trained on the set of 60 languages for 265k update steps. **Bold** numbers indicate better performance for the respective language.

	en F1 / EM	ar F1 / EM	hi F1 / EM	ru F1 / EM	th F1 / EM	vi F1 / EM	*avg* F1 / EM
X-MOD	**85.1 / 73.4**	**68.1 / 52.4**	**67.5 / 50.3**	**75.0 / 57.8**	**66.3 / 52.6**	**74.9 / 54.6**	**72.8 / 56.9**
SHARED	83.8 / 72.1	64.6 / 48.5	65.8 / 48.3	72.7 / 54.5	63.0 / 48.0	72.6 / 52.1	70.4 / 53.9

Table 6: Average F_1 and Exact Match results for **pre-trained languages**, on the test set of **XQuAD** for the X-MOD and SHARED model variants, pre-trained on the set of 60 languages for 265k update steps. **Bold** numbers indicate better performance for the respective language.

	de F1 / EM	es F1 / EM	zh F1 / EM	*avg* F1 / EM
X-MOD	**63.8 / 48.9**	**68.8 / 50.3**	**61.7 / 36.4**	**64.8 / 45.2**
SHARED	58.9 / 44.1	66.7 / 48.3	56.5 / 32.2	60.7 / 41.5

Table 7: Average F_1 and Exact Match results for **added languages**, on the test set of **MLQA** for the X-MOD and SHARED model variants, pre-trained on the set of 60 languages for 265k update steps. **Bold** numbers indicate better performance for the respective language.

	en	af	ar	bn	et	eu	fa	fi	fr	hi	hu	id	it	ka	ko	ru	sw	ta	th	vi	avg
X-MOD	81.4	**78.9**	43.5	**63.2**	**76.2**	**62.2**	**44.3**	**78.6**	**77.2**	**70.1**	**78.3**	50.5	**78.7**	**67.3**	**53.0**	**59.1**	**73.4**	51.1	2.8	**66.2**	**62.8**
SHARED	**81.5**	74.1	**44.2**	62.4	70.7	58.1	40.3	74.4	74.7	64.4	74.2	**51.5**	75.5	61.5	46.0	58.3	57.2	**52.5**	**4.0**	63.7	59.5

Table 8: Average F_1 results for **pre-trained languages**, on the test set of **NER** for the X-MOD and SHARED model variants, pre-trained on the set of 60 languages. **Bold** numbers indicate better performance for the respective language.

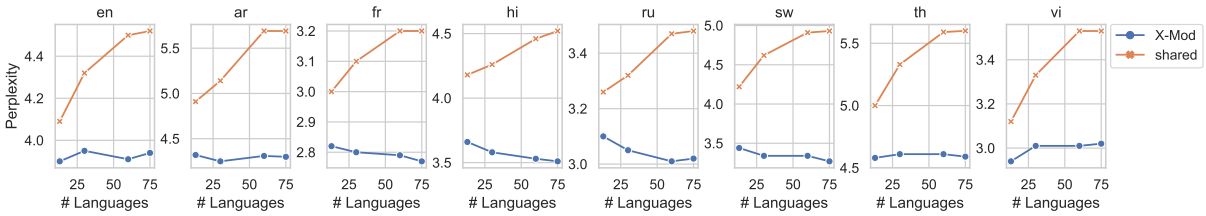
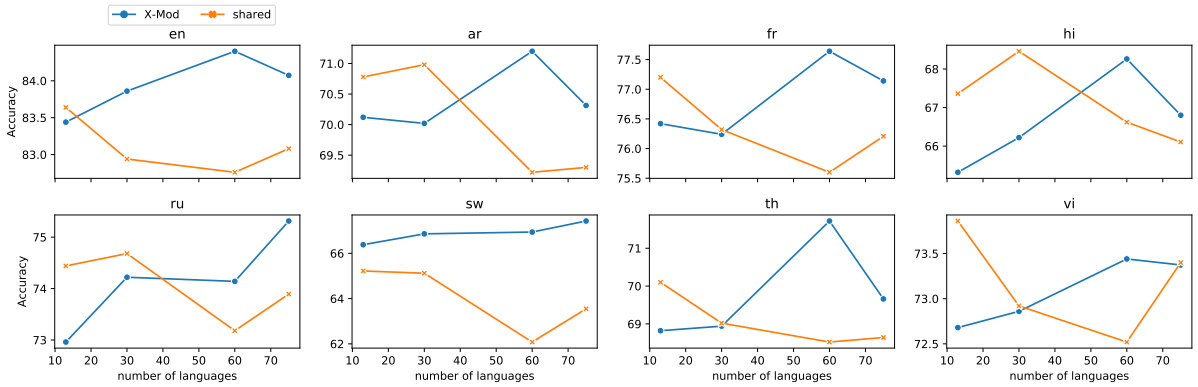
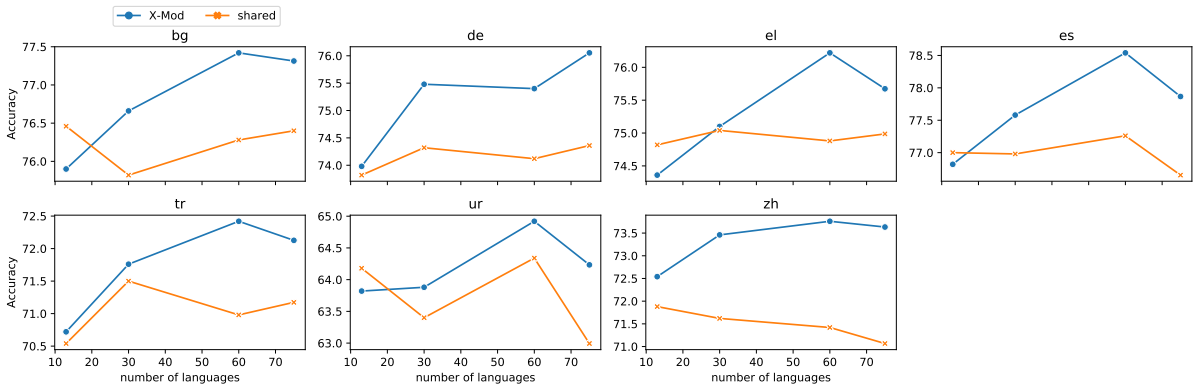


Figure 9: Perplexity when training on more languages. Each model has seen the **same amount of examples** in each language. Lower perplexity indicates better performance.

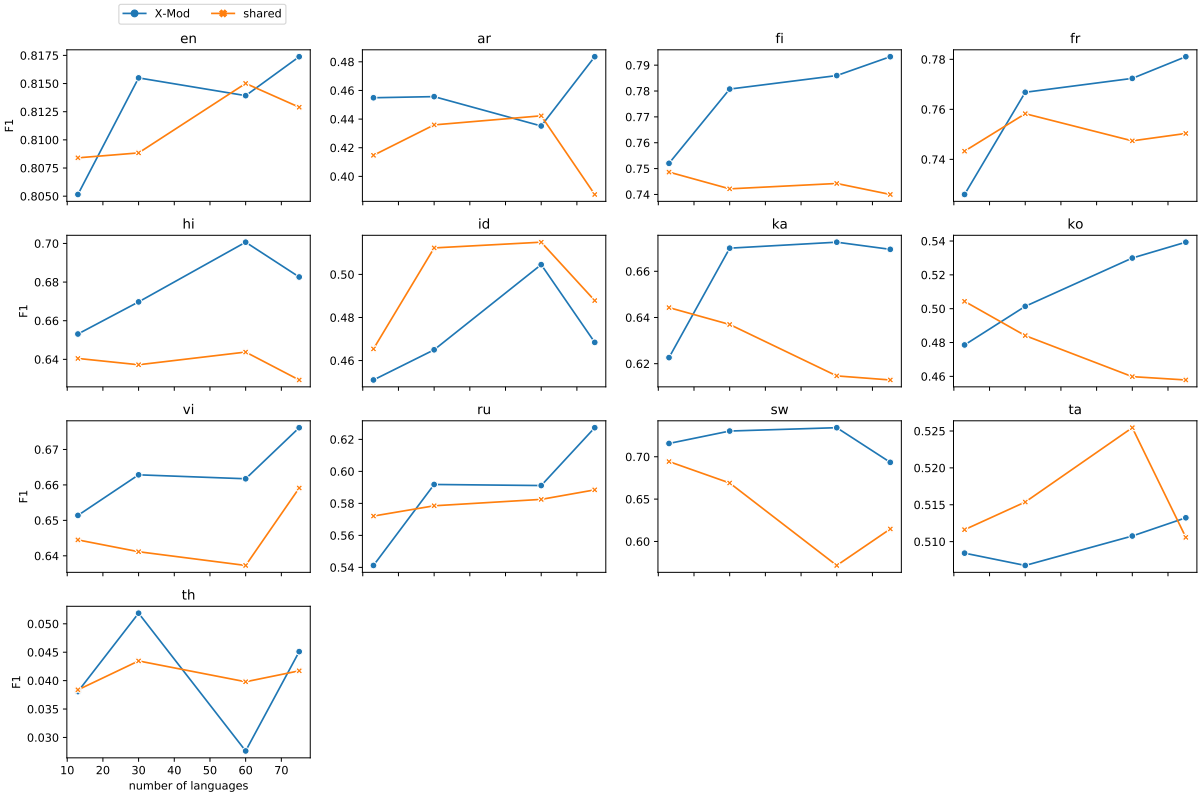


(a) Pre-Trained Languages

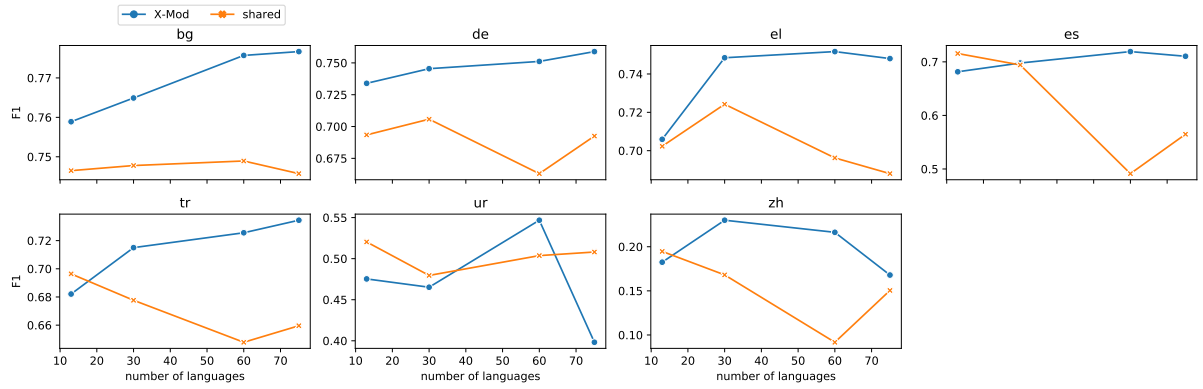


(b) Added Languages

Figure 10: Testset results on **XNLI** of pre-trained (top) and added (bottom) languages trained on different numbers of languages. Models trained on more languages are trained for longer → all models have seen the **same amount of examples** in each individual language. Scores are averaged across all random seeds.



(a) Pre-Trained Languages



(b) Added Languages

Figure 11: Testset results on **NER** of pre-trained (top) and added (bottom) languages trained on different numbers of languages. Models trained on more languages are trained for longer → all models have seen the **same amount of examples** in each individual language. Scores are averaged across all random seeds.

Language	iso	Family	Script	13	30	60	75
Afrikaans	af	IE:Germanic	Latin			✓	✓
Albanian	sq	IE:Albanian	Latin		✓	✓	✓
Amharic	am	Afro-Asiatic	Amharic		✓	✓	
Arabic	ar	Afro-Asiatic	Arabic	✓,(+)	✓,(+)	✓,(+)	✓,(+)
Armenian	hy	IE:Armenian	Armenian		✓	✓	✓
Assamese	as	IE:Iranian	Assamese				✓
Basque	eu	Isolate	Latin		✓	✓	✓
Belarusian	be	IE:Slavic	Cyrillic			✓	✓
Bengali	bn	IE:Iranian	Bengali			✓	✓
Bosnian	bs	IE:Slavic	Latin				✓
Breton	br	IE:Celtic	Latin				✓
Bulgarian	bg	IE:Slavic	Cyrillic	+	+	+	+
Catalan	ca	IE:Romance	Latin			✓	✓
Chinese	zh	Sino-Tibetan	Chinese	+	+	+	+
Croatian	hr	IE:Slavic	Latin		✓	✓	✓
Czech	cs	IE:Slavic	Latin		✓	✓	✓
Danish	da	IE:Germanic	Latin			✓	✓
Dutch	nl	IE:Germanic	Latin			✓	✓
English	en	IE:Germanic	Latin	✓,(+)	✓,(+)	✓,(+)	✓,(+)
Estonian	et	Uralic	Latin			✓	✓
Esperanto	eo	Constructed	Latin			✓	✓
Finnish	fi	Uralic	Latin	✓	✓	✓	✓
French	fr	IE:Romance	Latin	✓,(+)	✓,(+)	✓,(+)	✓,(+)
Frisian	fy	IE:Germanic	Latin				✓
Galician	gl	IE:Romance	Latin			✓	✓
Georgian	ka	Kartvelian	Georgian	✓	✓	✓	✓
German	de	IE:Germanic	Latin	+,(✓)	+,(✓)	+,(✓)	+,(✓)
Greek	el	IE:Hellenic	Greek	+,(✓)	+,(✓)	+,(✓)	+,(✓)
Gujarati	gu	IE:Iranian	Gujarati			✓	✓
Hausa	ha	Afro-Asiatic	Latin			✓	✓
Hebrew	he	Afro-Asiatic	Hebrew	+,(✓)	+,(✓)	+,(✓)	+,(✓)
Hindi	hi	IE:Iranian	Devanagari	✓,(+)	✓,(+)	✓,(+)	✓,(+)
Hungarian	hu	Uralic	Latin		✓	✓	✓
Icelandic	is	IE:Germanic	Latin			✓	✓
Indonesian	id	Austronesian	Latin	✓	✓	✓	✓
Irish	ga	IE:Celtic	Latin			✓	✓
Italian	it	IE:Romance	Latin		✓	✓	✓
Japanese	ja	Japonic	Japanese	+,(✓)	+,(✓)	+,(✓)	+,(✓)
Javanese	jv	Austronesian	Latin				✓
Kannada	kn	Dravidian	Kannada				✓
Korean	ko	Koreanic	Korean	✓,(+)	✓,(+)	✓,(+)	✓,(+)
Kurdish	ku	IE:Iranian	Latin			✓	✓
Latin	la	IE:Romance	Latin			✓	✓

Language	iso	Family	Script	13	30	60	75
Latvian	lv	IE:Slavic	Latin			✓	✓
Lithuanian	lt	IE:Slavic	Latin		✓	✓	✓
Macedonian	mk	IE:Slavic	Cyrillic			✓	✓
Malagasy	mg	Austronesian	Latin				✓
Malay	ms	Austronesian	Latin		✓	✓	✓
Malayalam	ml	Dravidian	Malayalam		✓	✓	✓
Marathi	mr	IE:Iranian	Devanagari				✓
Mongolian	mn	Mongolian	Cyrillic		✓	✓	✓
Nepali	ne	IE:Iranian	Devanagari			✓	✓
Norwegian	no	IE:Germanic	Latin			✓	✓
Oriya	or	IE:Iranian	Odia				✓
Oromo	om	Afro-Asiatic	Ge'ez				✓
Pashto	ps	IE:Iranian	Arabic			✓	✓
Persian	fa	IE:Iranian	Arabic			✓	✓
Polish	pl	IE:Slavic	Latin		✓	✓	✓
Portuguese	pt	IE:Romance	Latin			✓	✓
Punjabi	pa	IE:Iranian	Gurmukhi				✓
Romanian	ro	IE:Romance	Latin		✓	✓	✓
Russian	ru	IE:Slavic	Cyrillic	✓,(+)	✓,(+)	✓,(+)	✓,(+)
Sanskrit	sa	IE:Iranian	Devanagari			✓	✓
Scottish Gaelic	gd	IE:Germanic	Latin				✓
Serbian	sr	IE:Slavic	Cyrillic			✓	✓
Sindhi	sd	IE:Iranian	Arabic			✓	✓
Sinhala	si	IE:Iranian	Sinhala		✓	✓	✓
Slovak	sk	IE:Slavic	Latin		✓	✓	✓
Slovenian	sl	IE:Slavic	Latin			✓	✓
Somali	so	Afro-Asiatic	Latin			✓	✓
Spanish	es	IE:Romance	Latin	+,(✓)	+,(✓)	+,(✓)	+,(✓)
Sundanese	su	Austronesian	Latin				✓
Swahili	sw	Niger-Congo	Latin	✓	✓	✓	✓
Swedish	sv	IE:Germanic	Latin		✓	✓	✓
Tagalog	tl	Austronesian	Latin		✓	✓	✓
Tamil	ta	Dravidian	Tamil	✓	✓	✓	✓
Telugu	te	Dravidian	Telugu			✓	✓
Thai	th	Kra-Dai	Thai	✓,(+)	✓,(+)	✓,(+)	✓,(+)
Turkish	tr	Turkic	Latin	+,(✓)	+,(✓)	+,(✓)	+,(✓)
Ukrainian	uk	IE:Slavic	Cyrillic	+,(✓)	+,(✓)	+,(✓)	+,(✓)
Urdu	ur	IE:Iranian	Arabic	+,(✓)	+,(✓)	+,(✓)	+,(✓)
Vietnamese	vi	Austroasiatic	Latin	✓,(+)	✓,(+)	✓,(+)	✓,(+)
Welsh	cy	IE:Celtic	Latin			✓	✓
Xhosa	xh	Niger-Congo	Latin				✓
Yiddish	yi	IE:Germanic	Hebrew				✓

Table 9: List of languages we pre-train ✓on or add + in the different sets (13, 30, 60, 75). (·) indicates the respectively different pre-training/added languages of models 1 and 2 as described in §5.2 and Table 4. IE stands for Indo-European.