Breaking the Curse of Multilinguality with Cross-lingual Expert Language Models

Terra Blevins[1†] **Tomasz Limisiewicz**[2*] **Suchin Gururangan**[1] **Margaret Li**[1]
Hila Gonen[1] **Noah A. Smith**[1,3] **Luke Zettlemoyer**[1]
[1]Paul G. Allen School of Computer Science and Engineering, University of Washington
[2]Faculty of Mathematics and Physics, Charles University in Prague
[3]Allen Institute for Artificial Intelligence

Abstract

Despite their popularity in non-English NLP, multilingual language models often underperform monolingual ones due to inter-language competition for model parameters. We propose Cross-lingual Expert Language Models (X-ELM), which mitigate this competition by independently training language models on subsets of the multilingual corpus. This process specializes X-ELMs to different languages while remaining effective as a multilingual ensemble. Our experiments show that when given the same compute budget, X-ELM outperforms jointly trained multilingual models across all 16 considered languages and that these gains transfer to downstream tasks. X-ELM provides additional benefits over performance improvements: new experts can be iteratively added, adapting X-ELM to new languages without catastrophic forgetting. Furthermore, training is asynchronous, reducing the hardware requirements for multilingual training and democratizing multilingual modeling.

1 Introduction

Massively multilingual language models (LMs), which are trained on terabytes of text in a hundred or more languages, underlie almost all non-English and cross-lingual NLP applications (Scao et al., 2022; Lin et al., 2022, i.a.). Despite their wide adoption, these models come at a cost: the many languages are represented in the same, fixed model capacity, causing performance on individual languages to degrade relative to monolingual models (Conneau et al., 2020; Chang et al., 2023). This phenomenon (termed the *curse of multilinguality*) can significantly harm low-resource language performance (Wu and Dredze, 2020).

In this paper, we address this *curse* with **Cross**-lingual **E**xpert **L**anguage **M**odels (X-ELM, Figure 1), an ensemble of language models initialized from a pretrained multilingual model and each independently trained on a different portion of a multilingual corpus with x-BTM, a new extension of the Branch-Train-Merge paradigm (BTM; Li et al., 2022; Gururangan et al., 2023, §2) to the more heterogenous multilingual setting. X-ELM allows for efficient scaling of model capacity to better represent all considered languages.

x-BTM adapts existing BTM techniques to the multilingual setting by introducing a new cluster method for data assignments based on typological language similarity (§3.2). We also propose Heirachical Multi-round Training (HMR; §4), a method for efficiently adapting trained X-ELMs to novel multilingual settings by branching from existing, typologically related X-ELMs; this method for adapting X-ELM to new languages strongly outperforms standard language adaptation methods.

We train X-ELMs on 20 languages—including adapting to 4 unseen ones—with up to 21 billion training tokens, with the 1.7B parameter XGLM model as the base (Lin et al., 2022). Our experiments show that X-ELM strongly outperforms the dense multilingual models given the same compute budget in every considered setting (§6), and that these improvements consistently benefit *every* language. Notably, monolingual experts generally underperform typologically-informed multilingual X-ELMs, indicating that linguistically targeted multilinguality can benefit language modeling. We then show that the language modeling gains of X-ELM hold on downstream evaluations (§7).

Multilingual modeling with X-ELM provides additional benefits beyond improved performance. Training a set of X-ELMs is more computationally efficient than training a comparable dense model; each expert is trained independently, which removes the overhead cost of cross-GPU synchronization (Li et al., 2022) and allows asynchronous model training in low-compute settings. Similarly,

†Correspondence to blvns@cs.washington.edu
*Work done while visiting the University of Washington.

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 10822–10837
November 12-16, 2024

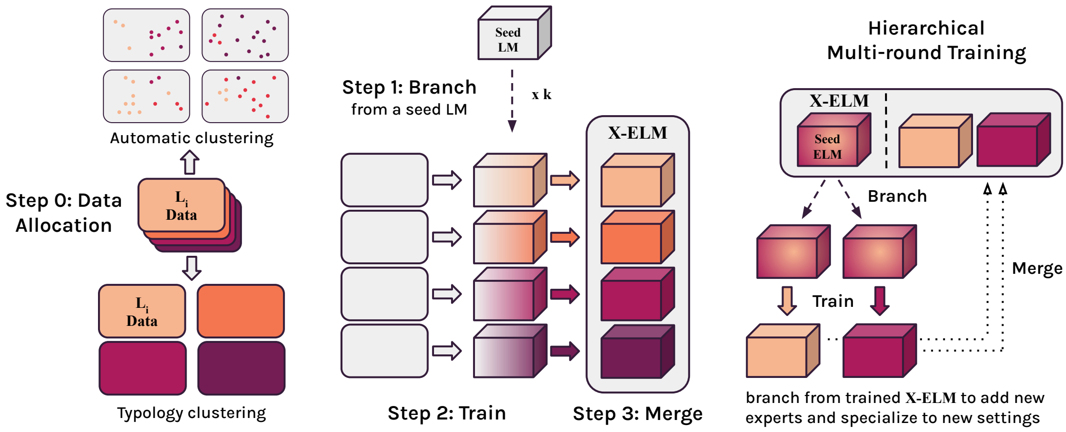

Figure 1: Overview of the X-ELM pretraining procedure. **Left**: We partition the multilingual text corpus into k subsets either through *automatic TF-IDF* clustering of documents or through grouping languages by *linguistic typology*. **Center**: Branch-Train-Merge (BTM) pretraining method. We initialize (*branch*) k experts from a seed LM, *train* each expert on a different cluster from the pretraining corpus, and *merge* the experts into a set of X-ELMs. **Right**: Hierarchical Multi-Round (HMR) training procedure (§4).

adapting X-ELMs to new languages with HMR training—a popular use case of language models (i.e., Chau et al., 2020)—is more efficient than continued training of a dense LM and does not risk forgetting previously learned languages (Yogatama et al., 2019). As a result, X-ELM allows much more efficient and effective multilingual modeling than prior approaches, democratizing multilingual NLP. We release the code and trained models.[1]

2 Background: Branch-Train-Merge

Multilingual LMs are typically trained in a *dense* manner, where a single set of parameters are updated with every training batch. When training large LMs, the dense training setup calculates gradients on and synchronizes model parameters across many GPUs.[2] This requires all GPUs to be available simultaneously and incurs communication costs that prolong training.

Branch-Train-Merge (BTM; Li et al. 2022) alleviates this cost by dividing the total compute among smaller expert language models that are trained independently on different domains and then combined during inference time. While the total number of parameters increases with the number of experts, inference with these models often uses a subset of experts (see §3.3), keeping inference costs manageable. c-BTM (Gururangan et al., 2023) then generalizes the above approach with automatic clustering of domains. Across multiple corpora, they show that (1) the optimal number of experts increases with data and compute and (2) a set of small expert models performs similarly to equivalently sized dense models at vastly reduced FLOP budgets.

Our work extends these studies to the multilingual setting, where experts are specialized to different languages instead of English-language domains. In the multilingual setting, we can also use typological structure to specialize experts, which we show provides additional benefits over automatic clustering. We also demonstrate that leveraging the hierarchy of language families in multi-round training yields further performance gains.

3 Cross-lingual Expert Language Models

Multilingual language models are jointly trained on many different languages (e.g., Lin et al., 2022), despite the well-documented effect this has on individual language performance (Conneau et al., 2020; Wang et al., 2020). We propose **Cross-lingual Expert Language Models**, or X-ELMs (Figure 1), which we hypothesize will alleviate the curse of multilinguality while maintaining the cross-lingual properties of dense multilingual LMs.

3.1 x-BTM: Sparse Multilingual Training

This section overviews our algorithm for the sparse training of multilingual experts.

Step 0: Multilingual Data Allocation As a preprocessing step, we partition the multilingual corpus into k clusters to train each X-ELM. We con-

[1] https://github.com/blvns/x-elm/

[2] For example, the XGLM-7.5B model "was trained on 256 A100 GPUs for about 3 weeks" (Lin et al., 2022).

sider both TF-IDF clusters and a new clustering method that groups documents by language identity and linguistic typology (§3.2).

Step 1: Branch A preliminary stage of shared, dense pretraining is important for ensembling expert language models (Li et al., 2022). Therefore, the first step of BTM is to initialize (*branch*) each expert with the parameters from a partially trained language model. For this work, we initialize our X-ELMs with an existing multilingual pretrained model, XGLM (Lin et al., 2022).

Step 2: Train After initialization, we assign each expert a data cluster and train for a fixed number of steps with an autoregressive LM objective. Expert training is independent, with no shared parameters between models.

Step 3: Merge We collect the k X-ELMs into a set and perform inference with them. We consider several methods of inference and expert ensembling in §3.3.

Steps 1–3 describe a single round of x-BTM training. However, we can continue to update the X-ELM set by branching—initializing a new group of experts—from existing models in the ensemble and performing more rounds of x-BTM via the method proposed in §4. This allows us to further improve X-ELM by training and adding new experts.

3.2 Data Allocation Methods

How we assign data to experts is a key component of training X-ELM, particularly as the data becomes more diverse (i.e., spanning many languages). We consider two methods of data allocation:

Balanced TF-IDF Clustering We partition the multilingual corpus automatically into k components with k-means clustering. First, we encode each document into a word-level TF-IDF representation;[3] we then perform balanced k-means clustering on these representations to obtain approximately balanced subsets of the data on which to train each X-ELM. Further details on the balanced k-means clustering method can be found in Gururangan et al. (2023). This allocation method uses no language information outside of what is inherent in the text (e.g., script, vocabulary).

[3]Data tokenization is independent of the downstream model. Here, we use the sklearn text-vectorizer tokenizer.

Linguistic Typology Clustering We also consider segmenting the corpus by language identity.[4] Rather than balancing the amount of data allocated to each cluster in this setting, we keep the number of languages per cluster fixed. Specifically, we learn a balanced hierarchical clustering of the languages (Figure 2). We build this hierarchy using the language similarity metrics in LANG2VEC (Littell et al., 2017), which represents languages based on linguistic features in resources such as WALS[5] and estimates language similarity with distance in this feature space. We initialize each cluster with a single language; at each step, we merge each cluster with exactly one other based on the *minimum* distances between cluster centroids. We then group languages by the resulting hierarchy and desired number of experts. When the number of languages equals the number of experts, typological clustering results in monolingual training, as every language is assigned a separate expert.

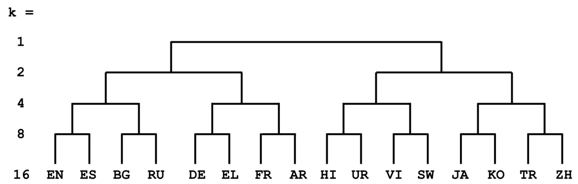

Figure 2: Hierarchical clustering of languages used to train our X-ELM ensembles.

3.3 Inference with X-ELMs

We evaluate multiple methods for performing inference with X-ELMs:

Top-1 Expert This method performs inference with a single expert chosen prior to evaluation, which incurs the same inference cost as the dense baselines. When evaluating *Typology* experts on a particular language ℓ, we choose the expert that included ℓ in the set of languages on which they continued pretraining. Similarly, when evaluating *TF-IDF*, we choose the X-ELM trained on the highest percentage of ℓ's data.

Ensembling TF-IDF Experts We also consider ensembling TF-IDF experts by adapting the c-BTM routing method (Gururangan et al., 2023). Here, we calculate ensembling α, or weights, over the experts at *each* inference step based on the prior con-

[4]This requires knowledge of each document's language. We use the language tags provided with mC4.

[5]World Atlas of Language Structures, `https://wals.info/`

text's TF-IDF distance from the experts' k-means centroids. These weights are then used to ensemble the output probabilities from each expert.

More specifically, given a probability from each expert $p_e(x_t \mid x_{<t})$ and the corresponding ensemble weight $\alpha_e = p(e \mid x_{<t}) \propto \exp(-\text{dist}(x_{<t}, c_e)^2/T)$, the probability of the ensemble $p_E(x_t \mid x_{<t}) = \sum_{e \in E} \alpha_e \cdot p_e(x_t \mid x_{<t})$. Here, $\text{dist}(x_{<t}, c_e)$ is obtained by embedding $x_{<t}$ with the learned TF-IDF vectorizer and calculating the Euclidean distance from c_e (the centroid over the data representations allocated to expert e), and T is a temperature parameter over the ensemble weight distribution.

Ensembling X-ELM outputs increases the cost of inference relative to dense models or top-1 inference. However, it can potentially better fit different subsets of data in a diverse evaluation set. We also do not assume we know the language of each example when ensembling, which makes this approach more flexible than the top-1 setting. In most cases, we ensemble all k experts; however, we can also reduce computational costs by *sparsifying* the ensemble and only activating the m ($< k$) experts that most contribute to an example: $p_E(x_t \mid x_{<t}) = \sum_{e \in E} \alpha_e \cdot p_e(x_t \mid x_{<t}) : \alpha_e \in \text{top-m}(\alpha_{\mathbf{E}})$. Appendix Table 8 presents the performance tradeoff with sparser TF-IDF ensembles.

4 Hierarchical Multi-Round Training

We previously described a single round of training for X-ELM (§3.1). However, BTM can also be used iteratively to train new experts seeded with those learned in a prior round. The multilingual setting provides a natural extension of multi-round training that leverages typological structure.

We propose **Hierarchical Multi-Round** (**HMR**) pretraining (Figure 1), which uses the learned typological tree structure from *Linguistic Typology* clustering to iteratively train more specific X-ELMs. Specifically, given an expert model x trained on a cluster of languages L, we initialize a new set of experts $X' = x'_1, x'_2, ..., x'_n$ with the parent expert x. Each new expert in X' is then further trained on a different sub-cluster $\ell' \subset L$.

HMR pretraining gives multiple benefits over single-round BTM. In particular, HMR training saves compute and more easily adapts our X-ELMs to new settings. A specific application of this is adding new languages to the model: while updating dense multilingual LMs with new languages is difficult and can lead to catastrophic forgetting of existing languages (Winata et al., 2023), hierarchically training an expert on a new language adds it to the X-ELM set without altering the existing information in other experts. We evaluate HMR training with this use case in §6.3.

5 Experimental Design

5.1 Pretraining Data and Languages

We train our X-ELMs on mC4, an open-source, multilingual pretraining corpus derived from CommonCrawl (Xue et al., 2021).[6] mC4 provides language tags for each document in the corpus, which were automatically assigned with cld3[7] when the dataset was constructed; we use these language tags during typological clustering (§3.2). We focus our experiments on the 16 highest-resourced languages out of the 30 languages on which the seed LM, XGLM-1.7B, was trained. For languages with significantly more data than the others (e.g., English), we subsample their data to the first 1,024 shards. Appendix Table 5 gives the languages and data quantities in our pretraining corpus.

5.2 Pretraining Settings

Each expert in the X-ELM experiments is a 1.7B parameter model with the same architecture as the 1.7B XGLM transformer model (Lin et al., 2022), and they are initialized with XGLM's weights in the initial round of BTM training. Unless otherwise stated, we keep the training parameters from the original XGLM training procedure; further details are given in Appendix A.1.

We train the experts for a fixed number of training steps. The exact parameters and resources used for each X-ELM experiment are reported in Table 4: in every setting, we control for the number of tokens seen during training. This ensures that all experts in a setting see the same amount of data (and undergo the same number of training updates) and that experiments across different expert set sizes but under the same training budget are comparable. For most experiments, we use a shared budget of 10.5B tokens; where indicated, we increase this to 21.0B tokens to test the effect of further training.

[6] While one could also continue pretraining with the same corpus that the seed LM was trained on, the pretraining data for XGLM is not publicly available.

[7] https://github.com/google/cld3

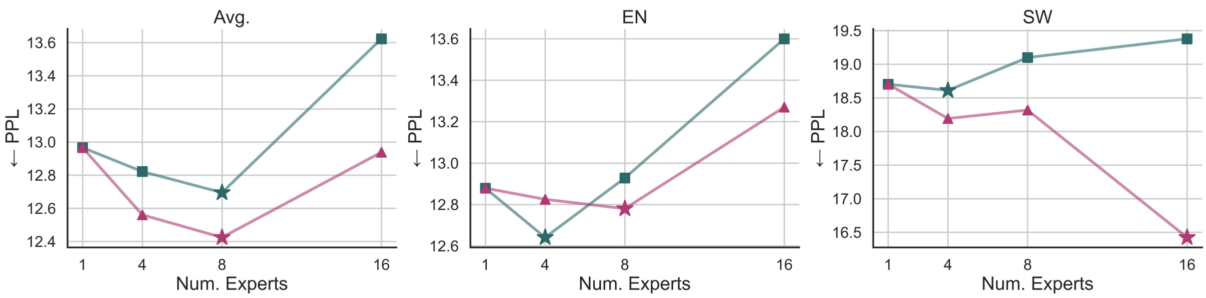

Figure 3: Average and language-specific (EN and SW) perplexities across different expert counts k (Num. Experts). In each evaluation setting, we compare clustering training data for experts with the TF-IDF$_{top1}$ (square) and Linguistic Typology (triangle) methods (§3.2). The best choice of k for each setting is marked with a star.

5.3 Baselines

We compare the performance of our X-ELM experiments to two dense baselines: **XGLM**, which is the 1.7B parameter seed model used to initialize each expert prior to x-BTM training, and a single **dense** model, which is also initialized with XGLM-1.7B's weights and then trained with the full data and compute budget split across experts in other X-ELM settings. This is equivalent to k=1 experts in cases where we vary the number of experts.

Since the curse of multilinguality is often evaluated in comparison to monolingual modeling, We also consider the setting where we train **monolingual** expert models on each target language (§6.1). Given that we consider sixteen languages in our X-ELM experiments, this corresponds to the $k = 16$ typological clustering setting.

5.4 Perplexity Evaluation

To evaluate the language modeling performance of the X-ELMs, we separately calculate the perplexity on the mC4 validation sets of each pretraining language. For languages with larger evaluation sets, we estimate performance on the first 5,000 validation examples. This perplexity metric is not comparable across languages, as they have different validation sets.

6 Language Modeling Experiments

We now test the effectiveness of sparse language modeling in the multilingual setting. First, we determine the optimal number of clusters for our given compute budget and dataset (§6.1). We then demonstrate that X-ELMs outperform comparable dense models on seen languages (§6.2) and more effectively adapt to new, unseen languages (§6.3).

6.1 Choosing the Number of X-ELMs

We first consider which number of experts gives the best multilingual language modeling performance. Figure 3 compares the choice of $k \in \{1, 4, 8, 16\}$ X-ELMs when trained on 10.5B tokens.[8] $k = 8$ is the best-performing setting on 75% of languages when clustering with TF-IDF and for 15 of the 16 pretraining languages when clustering by language similarity. Furthermore, typological clustering consistently outperforms TF-IDF.

These experiments indicate that, for the budget we evaluate, **the best overall X-ELM setting is bilingual models (k=8) clustered by language similarity**. This result is surprising, as it is intuitive to assume that simply continuing to pretrain each expert on a single language (i.e., the $k = 16$ setting) would lead to better perplexity. We find that one language, Swahili, does benefit from the monolingual $k = 16$ setting—possibly because Swahili is paired with a distant language (Vietnamese) by the typological clustering process. However, perplexity is higher in the $k = 16$ setting for all other languages, and in some cases, even underperforms the dense ($k = 1$) model.

6.2 Perplexity Results on Seen Languages

We now examine the performance of X-ELM in the best setting (k = 8) for the sixteen languages seen during BTM training on computational budgets of 10.5B and 21.0B tokens. Table 1 presents the perplexities of the TF-IDF clustered X-ELMs as well as the typologically (Typ.) clustered X-ELMs. As baselines, we compare against the original XGLM-1.7B model and a dense model trained on both computational budgets. We find that the best setting, $k = 8$ with typologically clustered experts,

[8] The $k = 16$ setting is equivalent to training monolingual experts for every language. Full results are in Table 1 for $k = 8$ and Appendix Table 7 for $k = 4$ and $k = 16$.

Lang.	XGLM	Dense	10.5B Training Tokens TF-IDF$_{top1}$	TF-IDF$_{ens}$*	Typ.	Dense	21.0B Training Tokens TF-IDF$_{top1}$	TF-IDF$_{ens}$*	Typ.
AR	16.85	15.29	**14.51**	14.56	14.66	14.97	**14.00**	14.05	14.16
BG	11.31	10.44	10.39	10.39	**10.25**	10.34	10.27	10.26	**10.09**
DE	15.53	14.02	**13.41**	13.50	13.42	13.72	**12.95**	13.05	12.97
EL	10.44	9.40	9.20	9.18	**9.17**	9.24	9.03	9.00	**8.98**
EN	14.37	12.88	12.93	**12.73**	12.78	12.69	12.68	**12.47**	12.55
ES	16.02	14.13	13.92	**13.76**	13.99	13.87	13.54	**13.37**	13.69
FR	13.12	11.78	**11.19**	11.28	11.29	11.54	**10.79**	10.88	10.91
HI	18.28	14.28	14.86	14.19	**11.25**	13.68	14.36	13.62	**10.52**
JA	14.57	12.31	11.95	11.95	**11.49**	11.79	11.36	11.37	**10.88**
KO	8.82	7.79	7.72	**7.67**	**7.67**	7.67	7.61	**7.53**	7.54
RU	13.43	12.52	12.14	12.21	**12.08**	12.33	11.83	11.90	**11.74**
SW	19.85	18.70	19.10	18.76	**18.32**	18.61	19.04	18.67	**18.07**
TR	17.81	15.34	14.13	14.28	**13.80**	14.88	13.41	13.58	**13.03**
UR	14.38	13.45	13.40	13.57	**12.60**	13.38	13.26	13.52	**12.20**
VI	13.07	11.39	11.00	10.86	**10.22**	11.09	10.56	10.42	**9.69**
ZH	17.91	13.74	13.28	13.53	**11.98**	13.12	12.61	12.87	**11.24**
Avg.	14.74	12.97	12.70	12.60	**12.19**	12.68	12.33	12.28	**11.77**

Table 1: Per-language and average perplexity results for the $k = 8$ X-ELM experiments (original XGLM and $k = 1$ dense model included for comparison). Lower numbers are better. The best setting for each language is bolded per compute budget. *TF-IDF ensemble uses more parameters for inference than other evaluations; see Table 8 for the effect of sparsifying these ensembles on perplexity.

improves by 2.97 and 1.20 on average over the seed and dense baseline models and has individual language gains of up to 7.77 and 3.76 over these models, respectively.

Expert language models outperform dense continued training For most languages (10 of 16), typologically clustered experts are the best-performing setting. For some high-resource languages (EN and ES), ensembling the TF-IDF experts works better than a single expert. However, this inference setting requires more parameters, as it uses all X-ELMs instead of just the single best expert per language. Furthermore, training X-ELMs for longer unsurprisingly outperforms lower compute settings. All of our experimental settings outperform the seed XGLM model; similarly, the experiments with the 21.0B token compute budget perform better than the respective experiment trained with 10.5B tokens.

X-ELMs improve language modeling on all languages We also show that multilingual language modeling with X-ELMs does not disproportionally benefit languages with more pretraining data (Figure 4). Instead, perplexity improvements over both the seed LM and the dense LM baseline *may* slightly favor low-resource languages ($\rho = -0.19, -0.26$, respectively).

6.3 Unseen Languages and Modeling New Languages with X-ELM

A common method for applying multilingual language models to new settings is language-adaptive pretraining (LAPT; Chau et al., 2020)), as this reuses multilingual knowledge in existing LMs while training on the target language(s). We now examine how well X-ELM performs in this paradigm by (1) evaluating X-ELM perplexity on unseen languages and (2) adapting an existing X-ELM set to new languages. Specifically, we consider both zero-shot evaluation and further training of X-ELM on four languages not included in the

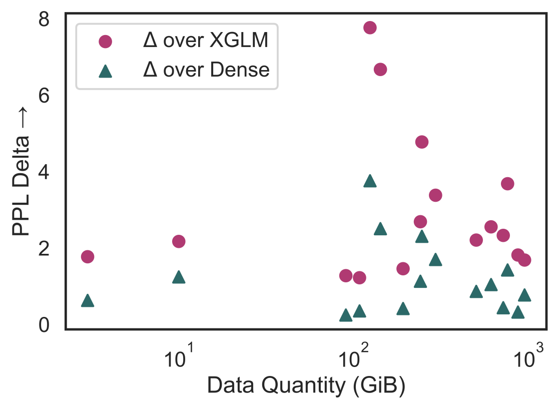

Figure 4: PPL improvements per language over XGLM-1.7B (circle) and dense baseline (triangle) against the training data quantity (for typ. clustered experts).

original XGLM seed model: Azerbaijani (AZ), Hebrew (HE), Polish (PL), and Swedish (SV).[9]

Unseen Language Evaluation We evaluate the existing dense baseline and ensembled TF-IDF clustered experts from the 21B token compute budget (§6.2) to test whether continued pretraining with x-BTM improves performance on unseen languages (**X-ELM Training**). We also compare these results to XGLM. We note these models *never* trained on the target languages.

Table 2 presents the unseen target language perplexities in the **XGLM** and **X-ELM Training** columns. We find that the original XGLM model performs poorly on the new languages, particularly those less related to XGLM's highest-resourced ones (i.e., AZ and HE). While the perplexities remain high in the dense model and TF-IDF ensembles, continued training on other languages improves the seed model.

Adapting X-ELM to new languages We now consider how well Hierarchical Multi-Round training (**HMR**, §4) works for language adaptive pretraining (LAPT, Chau et al., 2020), which incorporates new target languages into the continued pretraining process. Here, we group each *target* language with a higher-resource *donor* language already in our pretraining set; these are assigned with the language similarity metric used for typological clustering. We seed each new language's expert with an expert specialized to that language's donor; the new expert is then trained on the donor/target language pair. For HMR inference, we evaluate perplexity with the expert trained for that target language; we also evaluate on the donor languages to see what benefit, if any, they receive from the adaptation process.

We compare HMR against jointly continuing training on all four new languages and their respective donors in a single model (**Dense**). Each setting builds on models from the 10.5B compute budget: we continue training on the dense baseline for dense LAPT and branch from the donor languages' k=8 typological experts for HMR training.

All of the LAPT settings provide considerable improvements on the new target languages over the unseen language experiments (Table 2, **LAPT** columns). The HMR setting outperforms continued dense training on every new language. Furthermore, HMR training removes the risk of *catastrophic forgetting* (Yogatama et al., 2019) in other LAPT schemes, as this process adds new experts to X-ELM rather than changing existing ones.[10]

We also find that this setting provides performance gains on two donor languages over the experiments in §6.2. This is likely due to further training with more closely related languages for these languages (e.g., performing training on Arabic with Hebrew rather than French), consequently providing a more informative training signal for the higher-resource donor language as well.

[9]Data for these languages is also obtained from mC4, with the same preprocessing as other languages in our experiments.

Lang	XGLM	X-ELM Training		LAPT	
		Dense	TF-IDF$^{*}_{ens}$	Dense	HMR
Target					
AZ	1467.45	739.58	722.10	65.73	**32.74**
HE	1817.07	685.02	815.96	53.08	**26.21**
PL	211.76	160.70	178.63	17.71	**16.60**
SV	105.27	92.55	99.24	27.37	**26.16**
Donor					
TR	17.81	15.34	14.28	14.69	**12.72**
AR	16.85	15.29	14.56	14.80	**13.52**
RU	13.43	12.52	12.21	12.28	12.02
EN	14.37	12.88	12.73	12.65	12.63

Table 2: Perplexity results on unseen target languages and their respective donor languages. Donor language performance is only **bolded** if these results outperform all other X-ELM settings in that language (Table 1).

7 In-Context Learning Experiments

We also measure whether the perplexity improvements in X-ELMs correspond to better performance on downstream tasks. We test X-ELMs on three tasks in an in-context learning (ICL) framework, showing X-ELM language modeling gains *do* translate to ICL improvements over the baseline models.

7.1 Experimental Setup

We test the in-context learning abilities of X-ELM on three downstream tasks:

XNLI (Conneau et al., 2018) is a multilingual natural language inference benchmark covering 14 of our 16 pretraining languages (excluding JA and KO). Since there are no gold training examples for XNLI, we use the test set for evaluation and sample demonstrations from the validation set.

XStoryCloze (Lin et al., 2022) is a manually translated benchmark extending StoryCloze (Mostafazadeh et al., 2016) to other languages. This is a story-completion task wherein the model identifies the correct final sentence of a short story.

[10]This forgetting of known languages occurs in our dense LAPT baseline, with perplexity decreasing by 1.91 points on average for languages not included in the adaptation setting.

	Model	XNLI		XStoryCloze		PAWS-X	
		Acc.	Win Rate	Acc.	Win Rate	Acc.	Win Rate
Zero-shot	XGLM (1.7B)	44.88	28.6%	57.76	**28.6%**	48.54	14.3%
	Dense	44.31	7.1%	56.10	0.0%	48.44	28.6%
	Typ. (TRG)	44.17	7.1%	57.79	**28.6%**	49.86	**42.9%**
	TF-IDF (Top-1)	43.77	14.3%	**57.80**	**28.6%**	**50.04**	28.6%
	TF-IDF (Ens.)	**45.10**	**42.9%**	57.46	14.3%	49.93	0.0%
Few-shot	XGLM (1.7B)	42.34	28.6%	53.21	0.0%	54.52	0.0%
	Dense	41.70	0.0%	55.00	0.0%	54.81	14.3%
	Typ. (TRG)	42.15	†14.3%	54.62	†**71.4%**	55.39	†28.6%
	Typ. (EN)	42.43	†7.1%	**55.54**	†28.6%	55.13	14.3%
	TF-IDF (Top-1)	42.55	21.4%	55.03	†14.3%	**55.50**	†**42.9%**
	TF-IDF (Ens.)	**42.93**	**35.7%**	54.72	28.6%	54.57	14.3%

Table 3: Average performance (**Acc.**) and the percentage of languages where this setting outperforms the others (**Win Rate**) on the overlap of task evaluation languages and the X-ELM target languages. The **few-shot** setting provides k=8 English demonstrations to the model and averages performance across five runs. †Indicates (best) performance ties between two evaluation settings on a language when calculating the win rate.

This dataset covers seven of our pretraining languages and four other low-resource languages.

PAWS-X (Yang et al., 2019) is a binary classification task that requires the model to determine whether a pair of sentences are paraphrases. This benchmark covers seven of our pretraining languages, including two (JA and KO) that are not covered by the other ICL benchmarks.

We compare X-ELMs against dense baselines in zero- and few-shot settings. For all benchmarks, we evaluate on 1,000 random examples and perform five runs on different demonstrations for few-shot learning, using English demonstrations for every language to test cross-lingual transfer. Further details about the ICL evaluation are in Appendix A.2.

7.2 Results

We evaluate our best X-ELMs by perplexity—k=8 experts trained on the larger compute budget of 21B training tokens—on their in-context learning abilities (Table 3).[11] Here, we consider two metrics to summarize model performance across languages: **Acc.** is the average accuracy of each model for that ICL task across evaluation languages, and **Win Rate** is the percentage of languages where that model achieves the highest score out of the considered models; if two models get the highest score, they are both considered to *win* that setting.

We find that X-ELMs outperform both the seed and compute-matched dense baselines across the three tasks and in both the zero- and few-shot evaluation settings. Furthermore, though X-ELM improves over the seed model, the dense model underperforms XGLM. This may be due to using different data from the original XGLM pretraining, as data quality issues have been documented for mC4 (Kreutzer et al., 2022; Chung et al., 2023). We also note that few-shot ICL performance on XNLI and XStoryCloze is consistently lower than in the zero-shot setting; this is a recurring issue in multilingual ICL also observed in the seed model (Lin et al., 2022).

[11]Full results for each task are given in Appendix Tables 9, 10, and 11.

8 Related Work

8.1 Multilingual Pretraining and Adapation

Many variants of dense multilingual pretraining have been proposed since multilingual BERT (Devlin et al., 2019): changing the architecture and scaling the model size up (Goyal et al., 2021; Lin et al., 2022), adding additional cross-lingual objectives (Conneau and Lample, 2019; Chi et al., 2022; Reid and Artetxe, 2022), and careful language and data curation (Scao et al., 2022). Most similar to our work is Pfeiffer et al. (2022), which proposes an architecture, X-MOD, with language-specific modules. However, many slimitations of dense modeling persist here as the model and language modules are jointly trained.

Across most multilingual pretraining methods is the *curse of multilinguality* (Conneau et al., 2020), particularly for lower-resource languages in massively multilingual training (Wu and Dredze, 2020). Blevins et al. (2022) find that multilingual models forget information previously learned during training, which they hypothesize is due to this phenomenon; Wang et al. (2020) similarly suggest that this effect is due to training dynamics. More recently, Chang et al. (2023) presented a controlled study corroborating limited model capacity as a cause of this *curse*. A primary motivation of our work is to mitigate this *curse* while maintaining the other benefits of multilingual modeling.

However, not all multilinguality is harmful to language modeling. Chang et al. (2023) show that seeing linguistically similar languages can benefit low-resource language performance; this corroborates our finding that X-ELMS trained on related languages outperfrom monolingual experts. In this vein, a recent direction in multilinguality has been *targeted multilingual modeling*, where models are trained on data from the same language family (Ogueji et al., 2021; Ogunremi et al., 2023; Ljubešić et al., 2024; Downey et al., 2024).

We also consider how X-ELM can be used for language adaption. The most common method, language-adaptive pretraining (LAPT; Chau et al., 2020), continues multilingual pretraining with new languages incorporated into the training regime. Other work proposed using adapters to update the model with new languages (Pfeiffer et al., 2020); notably, Faisal and Anastasopoulos (2022) used similar linguistic motivations to our typological clustering to group languages for adapters. However, Ebrahimi and Kann (2021) found that LAPT outperformed adapters for adaptation.

8.2 Sparse Models for NLP

Sparse language models (Evci et al., 2020; Mostafa and Wang, 2019; Dettmers and Zettlemoyer, 2019) route inputs through a subset of the total model parameters. Our work builds most directly on the Branch-Train-Merge (Li et al., 2022; Gururangan et al., 2023) algorithm, which results in full-model experts specialized on domains defined by metadata or a learned clustering. This design expands on early Mixture-of-Experts (MoE) models (Jacobs et al., 1991) and on DEMix layers (Gururangan et al., 2022), which routes sequences to per-layer experts based on metadata.

Other MoE models have recently been applied to multilingual settings. Pfeiffer et al. (2022) develop a multilingual expert model with language-specific routing, and Kudugunta et al. (2021) develop a machine translation model with routing determined by the source-target language pair or the target language. Similar to BTM, Jang et al. (2023) trains experts specialized to different tasks, including five machine-translation language pairs.

9 Conclusion

This work presents an approach to break the *curse of multilinguality* by extending sparse language modeling to the multilingual setting with X-ELM. We find that X-ELMs achieve better perplexity *on every language* over standard, dense language models trained with the same compute budget; expert language models can also be easily adapted to new languages without catastrophic forgetting. X-ELMs present additional benefits over dense models for multilingual modeling, including training efficiency and flexibility. Finally, we show that these language modeling improvements transfer to downstream, in-context learning performance.

While our experiments show that X-ELM outperforms dense LMs, we foresee many avenues of future work to further tailor sparse modeling to multilinguality. These include better methods for data allocation—such as clustering methods that leverage cross-lingual signal— and algorithmic improvements to better allocate compute and more effectively ensemble models. By proving the efficacy of sparse language modeling in the multilingual setting, we hope to inspire future work in this vein that fairly models every language while leveraging the potential of cross-lingual learning.

Limitations

This work focuses on rigorously examining the effect of training X-ELMs in a limited number of settings, training languages, and data sources; this is both to ensure that we provide comprehensive comparisons with prior approaches to multilingual language modeling and due to computational limitations. Therefore, the proposed method should be further verified in other settings. In particular, in future work, we hope to examine how X-ELM performs at scale when using larger experts, more languages, and larger training budgets. Additionally, while we consider one seed model, XGLM (Lin et al., 2022), future work should examine the effect of other pretrained initializations as well as training our own seed models to test how much multilinguality is needed in X-ELM initialization.

We also note the limited nature of our downstream evaluations, which is due to (1) the limited number of multilingual benchmarks available and (2) our requirement that evaluation benchmarks overlap with (most of) our 16 pretraining languages. Furthermore, since we compare against the seed model, we focus on XGLM's original evaluation tasks and the prompting settings developed for this baseline (rather than developing our own that may be biased towards the X-ELM models).

Finally, training X-ELM rather than a single dense model increases some computational costs, similar to other BTM methods. The primary increase is in storage, as each expert's weights need to be stored separately. In some cases, the inference cost of X-ELM can be higher than the best model (e.g., when using an ensemble of experts); however, we propose several inference methods that only require loading a single model and demonstrate that you can sparsify the TF-IDF ensemble and achieve similar perplexities (Appendix Table 8).

Acknowledgements

We would like to thank Orevaoghene Ahia for helpful feedback on this project. Tomasz Limisiewicz acknowledges the support of grant 338521 of the Charles University Grant Agency, Fellowship from Paul G. Allen School, and the Mobility Fund of Charles University.

References

Terra Blevins, Hila Gonen, and Luke Zettlemoyer. 2022. Analyzing the mono- and cross-lingual pretraining dynamics of multilingual language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*.

Terra Blevins and Luke Zettlemoyer. 2022. Language contamination helps explain the cross-lingual capabilities of English pretrained models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Tyler A Chang, Catherine Arnett, Zhuowen Tu, and Benjamin K Bergen. 2023. When is multilinguality a curse? language modeling for 250 high-and low-resource languages. *arXiv preprint arXiv:2311.09205*.

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.

Zewen Chi, Shaohan Huang, Li Dong, Shuming Ma, Bo Zheng, Saksham Singhal, Payal Bajaj, Xia Song, Xian-Ling Mao, Heyan Huang, and Furu Wei. 2022. XLM-E: Cross-lingual language model pre-training via ELECTRA. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Hyung Won Chung, Noah Constant, Xavier Garcia, Adam Roberts, Yi Tay, Sharan Narang, and Orhan Firat. 2023. Unimax: Fairer and more effective language sampling for large-scale multilingual pretraining. *arXiv preprint arXiv:2304.09151*.

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 and Guillaume Lample. 2019. Cross-lingual language model pretraining. *Advances in neural information processing systems*, 32.

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.

Tim Dettmers and Luke Zettlemoyer. 2019. Sparse networks from scratch: Faster training without losing performance. *CoRR*, abs/1907.04840.

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)*.

Cm Downey, Terra Blevins, Dhwani Serai, Dwija Parikh, and Shane Steinert-Threlkeld. 2024. Targeted multilingual adaptation for low-resource language families. In *Findings of the 2024 Conference on Empirical Methods in Natural Language Processing*.

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.

Utku Evci, Trevor Gale, Jacob Menick, Pablo Samuel Castro, and Erich Elsen. 2020. Rigging the lottery: Making all tickets winners. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 2943–2952. PMLR.

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, pages 434–452.

Naman Goyal, Jingfei Du, Myle Ott, Giri Anantharaman, and Alexis Conneau. 2021. Larger-scale transformers for multilingual masked language modeling. In *Proceedings of the 6th Workshop on Representation Learning for NLP (RepL4NLP-2021)*.

Suchin Gururangan, Mike Lewis, Ari Holtzman, Noah A. Smith, and Luke Zettlemoyer. 2022. DEMix layers: Disentangling domains for modular language modeling. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5557–5576, Seattle, United States. Association for Computational Linguistics.

Suchin Gururangan, Margaret Li, Mike Lewis, Weijia Shi, Tim Althoff, Noah A. Smith, and Luke Zettlemoyer. 2023. Scaling expert language models with unsupervised domain discovery. *Preprint*, arXiv:2303.14177.

Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. 1991. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87.

Joel Jang, Seungone Kim, Seonghyeon Ye, Doyoung Kim, Lajanugen Logeswaran, Moontae Lee, Kyungjae Lee, and Minjoon Seo. 2023. Exploring the benefits of training expert language models over instruction tuning. In *International Conference on Machine Learning*.

Julia Kreutzer, Isaac Caswell, Lisa Wang, Ahsan Wahab, Daan van Esch, Nasanbayar Ulzii-Orshikh, Allahsera Tapo, Nishant Subramani, Artem Sokolov, Claytone Sikasote, et al. 2022. Quality at a Glance: An Audit of Web-Crawled Multilingual Datasets. *Transactions of the Association for Computational Linguistics*, 10:50–72.

Sneha Kudugunta, Yanping Huang, Ankur Bapna, Maxim Krikun, Dmitry Lepikhin, Minh-Thang Luong, and Orhan Firat. 2021. Beyond distillation: Task-level mixture-of-experts for efficient inference. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3577–3599, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Margaret Li, Suchin Gururangan, Tim Dettmers, Mike Lewis, Tim Althoff, Noah A. Smith, and Luke Zettlemoyer. 2022. Branch-train-merge: Embarrassingly parallel training of expert language models. *Preprint*, arXiv:2208.03306.

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. 2022. Few-shot learning with multilingual generative language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9019–9052, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Patrick Littell, David R Mortensen, Ke Lin, Katherine Kairis, Carlisle Turner, and Lori Levin. 2017. Uriel and lang2vec: Representing languages as typological, geographical, and phylogenetic vectors. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, volume 2, pages 8–14.

Nikola Ljubešić, Vit Suchomel, Peter Rupnik, Taja Kuzman, and Rik van Noord. 2024. Language models on a diet: Cost-efficient development of encoders for closely-related languages via additional pretraining. In *Proceedings of the 3rd Annual Meeting of the Special Interest Group on Under-resourced Languages@ LREC-COLING 2024*, pages 189–203.

Hesham Mostafa and Xin Wang. 2019. Parameter efficient training of deep convolutional neural networks by dynamic sparse reparameterization. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 4646–4655. PMLR.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 839–849, San Diego, California. Association for Computational Linguistics.

Kelechi Ogueji, Yuxin Zhu, and Jimmy Lin. 2021. Small data? no problem! exploring the viability of pretrained multilingual language models for low-resourced languages. In *Proceedings of the 1st Workshop on Multilingual Representation Learning*, pages 116–126, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Tolulope Ogunremi, Dan Jurafsky, and Christopher D Manning. 2023. Mini but mighty: Efficient multilingual pretraining with linguistically-informed data selection. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1221–1236.

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.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020. Mad-x: An adapter-based framework for multi-task cross-lingual transfer. *Preprint*, arXiv:2005.00052.

Machel Reid and Mikel Artetxe. 2022. PARADISE: Exploiting parallel data for multilingual sequence-to-sequence pretraining. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.

Zirui Wang, Zachary C Lipton, and Yulia Tsvetkov. 2020. On negative interference in multilingual models: Findings and a meta-learning treatment. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4438–4450.

Genta Indra Winata, Lingjue Xie, Karthik Radhakrishnan, Shijie Wu, Xisen Jin, Pengxiang Cheng, Mayank Kulkarni, and Daniel Preotiuc-Pietro. 2023. Overcoming catastrophic forgetting in massively multilingual continual learning. *Preprint*, arXiv:2305.16252.

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, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. 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.

Dani Yogatama, Cyprien de Masson d'Autume, Jerome Connor, Tomas Kocisky, Mike Chrzanowski, Lingpeng Kong, Angeliki Lazaridou, Wang Ling, Lei Yu, Chris Dyer, et al. 2019. Learning and evaluating general linguistic intelligence. *arXiv preprint arXiv:1901.11373*.

A Additional Experimental Details

A.1 Pretraining

Table 5 summarizes the languages we use, as well as their frequencies in the original XGLM pretraining dataset and in our sub-sampled mC4 corpus.

Table 4 presents the compute allocated to each expert and setting at different compute budgets of the X-ELM experiments. The per-model instance batch size (**bsz**) for all experiments is 2, and each training example had a sequence length (**seq. len**) of 2048. The total token budget (**# Tokens**) is the product of (k, # GPUs, # updates, grad acc., bsz, seq. len), normalized by the number of GPUs used for model parallelism (2).

The experts are trained with a linear decay learning rate schedule; we use a maximum learning rate of $1.5e-4$ after performing preliminary learning rate sweeps.

A.2 In-Context Learning

We reimplement the evaluation protocol from Lin et al. (2022), where the model scores multiple versions of every example (with the different possible labels filled in), and the label of the highest-scoring version is considered as the model's prediction. We use the English prompt formats and evaluation protocols developed for the seed LM of our experts, XGLM, for the downstream tasks of XNLI, XStoryCloze, and PAWS-X. The prompt templates we use are reprorted in Table 6.

In the few-shot setting, we perform five evaluation runs with different demonstration samples and reported the average performance. All few-shot experiments are performed with eight random demonstrations. Unless otherwise stated, we evaluate performance on the development set and sample

# Tokens	**k**	**# GPUs**	**# updates**	**grad acc.**
10.5 B	1	8	20,000	32
	4	4	20,000	16
	8	4	20,000	8
	16	2	20,000	8
21.0 B	1	8	40,000	32
	4	4	40,000	16
	8	4	40,000	8
	16	2	40,000	8

Table 4: Overview of the total compute budget and resources used for different X-ELM experiments. **k** is the number of experts, **# GPUs** indicates the number of GPUs used to train each expert, and **grad acc.** gives the number of gradient accumulation steps used.

demonstrations from the training set. As we are testing X-ELM's cross-lingual abilities, the demonstrations are in English for every target language.

A.3 Licensing and Intended Use

All of the artifacts used to build the X-ELMs presented in this work were released for use within academic research (Gururangan et al., 2023; Lin et al., 2022; Xue et al., 2021). This also holds for the evaluation benchmarks used to validate the X-ELMs (Xue et al., 2021; Conneau et al., 2018; Lin et al., 2022; Yang et al., 2019). Therefore, the intended use of the code and models presented in this work is in open-source research as well; paricularly, to improve the performance and useablilty of multilingual models for all languages.

Language	mC4† Size (%)	XGLM Size
AR (Arabic)	243.14 (4.1%)	64.34
BG (Bulgarian)	109.3 (1.9%)	61.10
DE (German)	615.59 (10.4%)	369.30
EL (Greek)	193.63 (3.3%)	180.37
EN (English)	877.43 (14.8%)	3,324.45
ES (Spanish)	723.17 (12.2%)	363.83
FR (French)	506.74 (8.6%)	303.76
HI (Hindi)	125.44 (2.1%)	26.63
JA (Japanese)	764.71 (12.9%)	293.39
KO (Korean)	91.29 (1.5%)	79.08
RU (Russian)	957.02 (16.2%)	1,007.38
SW (Swahili)	3.06 (0.05%)	3.19
TR (Turkish)	248.07 (4.2%)	51.51
UR (Urdu)	10.15 (0.2%)	7.77
VI (Vietnamese)	296.65 (5.0%)	50.45
ZH (Chinese)	143.68 (2.4%)	485.32
AZ (Azerbaijani)	15.23 (–)	–
HE (Hebrew)	67.14 (–)	–
PL (Polish)	393.85 (–)	–
SV (Swedish)	154.54 (–)	–

Table 5: The frequencies and relative percentages of different languages in our training corpus (†an mC4 subsample) and in the XGLM corpus, CC100-XL (Lin et al., 2022). Sizes are in gigabytes (GiB). EN, ES, FR, and RU are downsampled to 1,024 shards for mC4.

B Additional X-ELM Analysis

B.1 Comparing the Data Distribution of Clustering Techniques

Figure 5 shows the difference in language distributions between the *TF-IDF* and *Linguistic Typology* clusters. While *TF-IDF* allows language data to spread across experts, we find that, in practice, the distributions remain relatively sparse. The main exception is at $k = 16$, when the highest-resourced languages in the data (e.g., English or Russian) are split across clusters due to the constraint that balances the amount of data per cluster.

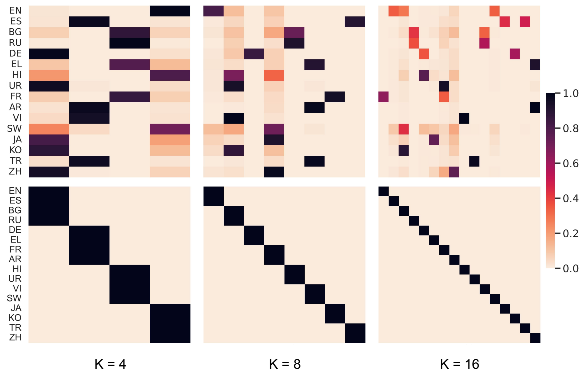

Figure 5: Percentage of language data assigned to different experts with TF-IDF (top row) and Typ. (bottom row) clustering. For Typ. clustering, each language is assigned entirely to a single expert.

B.2 Sparse TF-IDF Ensembling

In §6, we compare ensembling TF-IDF experts in an X-ELM set against choosing a single TF-IDF expert for inference based on the amount of in-language data seen by that expert during training. In the cases of m=2,4, this approach sparsifies the ensemble by dynamically selecting the top m experts based on their current ensemble weights. Here, we additionally consider how *sparsifying* the TF-IDF ensemble holds up against these other settings (Table 8). We find that for seen languages, reducing the number of experts active to just m=2 usually gives very similar performance to the full ensemble (m=8). However, this is not true in the case of *unseen* languages, where the m=8 setting consistently outperforms sparser ensembles.

B.3 X-ELM Forgetting

We evaluate X-ELMs as a set of models by dynamically choosing the best expert for a given evaluation setting or ensembling the experts' outputs. However, each expert is initialized with a model trained on all the languages we consider. This prompts the question: how much do individual experts *forget*[12] about the languages they are not specialized to?

Forgetting occurs as X-ELMs become more specialized. We compare the perplexity of each expert model on all pretraining languages to that of the seed model, XGLM-1.7B (Figure 6).Across the considered values of k, we see less forgetting in the

[12]We consider an expert to have *forgetten* information about a language if its perplexity on that language increases.

Dataset	Prompt	Labels
XNLI	`{Sentence 1}, right? [Mask], {Sentence 2}`	Entailment: `Yes` | Neural: `Also` | Contradiction: `No`
XStoryCloze	`{Context} [Mask]`	Identity
PAWS-X	`{Sentence 1}, right? [Mask], {Sentence 2}`	True: `Yes` | False: `No`

Table 6: Prompts used for the ICL experiments in §7; the `[MASK]` is filled with one of the label forms given in the last column. For XStoryCloze, `{Context}` refers to the format `{Sent. 1} {Sent. 2} {Sent. 3} {Sent. 4}`, and "Identity" refers to the text of one of the answers given for that example.

X-ELMs trained on TF-IDF clusters than in those clustered typologically. For the $k = 8$ expert setting, the TF-IDF experts only forget on 47.7% of settings, and when forgetting occurs, the perplexity increase over the baseline is 3.10 on average. For typologically clustered experts, these measures are 83.6% and 3.14, respectively.

We observe similar trends for the $k = 4$ and $k = 16$ X-ELMs. On average, the $k = 4$ TF-IDF experts experience forgetting in only 18.8% of cases with an average perplexity increase of 1.24 when forgetting occurs; the typology experts forget 78.1% of the time with an average perplexity increase of 1.34. For the $k = 16$ setting, these statistics are 60.9% and 0.9 for the TF-IDF clusters and 89.4% and 1.24 for the typology clusters.

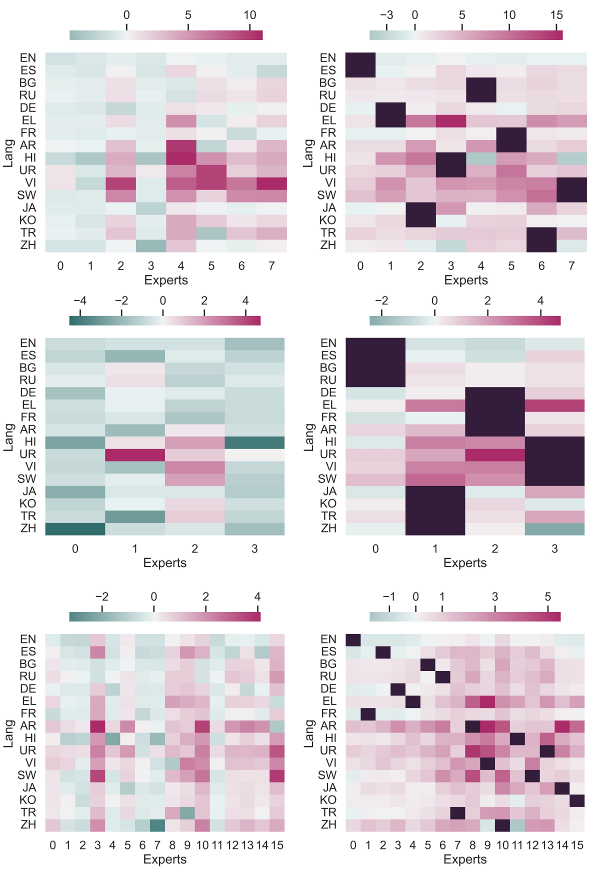

Figure 6: Heatmap comparing individual X-ELM perplexities to the seed LM with TF-IDF (left) and Typ. (right). Rows give results for $k = 8, 4, 16$, respectively. Positive scores indicate that the expert *forgot* that language. For Typ. clusters, languages that the model was explicitly trained on are grayed out.

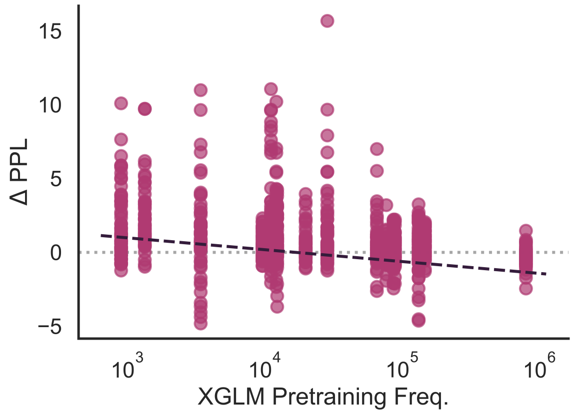

Figure 7: Per-expert deltas compared to the original XGLM-1.7B of every pretraining language plotted against the language's frequency in the original XGLM pretraining corpus ($\rho = -0.33$, $p << 0.001$).

X-ELMs are more likely to forget certain languages. For example, English is rarely forgotten, with only 25% of experts performing worse than the baseline. In comparison, 94.6% of experts perform worse on Urdu than XGLM. One potential cause of this discrepancy is the frequency with which the language was seen during seed training: languages that are more common in the XGLM pretraining corpus see fewer cases of forgetting and have smaller perplexity increases when it does occur (Figure 7). Another likely factor is inaccurate language classification in the BTM training data, which is a common issue when training language models on specific languages (Blevins and Zettlemoyer, 2022); this could lead to related, higher-resourced languages contaminating the datasets for lower-resourced ones (Kreutzer et al., 2022).

C Full Experimental Results

Table 7 presents the full perplexity results for the $k = 4$ and $k = 16$ X-ELM experiments, trained on

Lang.	XGLM	Dense	k=4 Experts TF-IDF$_{top1}$	TF-IDF$_{ens}$*	Typ.	k=16 Experts TF-IDF$_{top1}$	TF-IDF$_{ens}$*	Typ.
AR	16.85	15.29	14.99	15.03	15.00	15.60	15.67	15.40
BG	11.31	10.44	10.39	10.39	10.42	11.10	10.70	10.31
DE	15.53	14.02	13.85	13.89	13.71	14.71	14.43	14.5
EL	10.44	9.40	9.36	9.33	9.28	9.72	9.64	9.41
EN	14.37	12.88	12.64	12.71	12.78	13.60	13.23	13.27
ES	16.02	14.13	13.93	13.96	14.06	14.83	14.58	14.59
FR	13.12	11.78	11.62	11.65	11.55	12.38	12.13	12.15
HI	18.28	14.28	14.22	14.21	12.64	16.11	15.67	13.86
JA	14.57	12.31	12.23	12.12	11.73	13.39	13.14	13.18
KO	8.82	7.78	7.81	7.77	7.70	8.14	8.09	7.75
RU	13.43	12.52	12.30	12.33	12.46	12.96	12.76	12.82
SW	19.85	18.70	18.61	18.62	18.19	19.38	19.13	16.43
TR	17.81	15.34	14.85	14.96	14.81	15.67	15.78	15.52
UR	14.38	13.45	13.56	13.73	13.18	13.88	13.87	12.65
VI	13.07	11.39	11.43	11.21	10.32	11.85	11.65	11.59
ZH	17.91	13.74	13.38	13.70	13.11	14.65	14.95	13.58
Avg.	14.74	12.97	12.82	12.85	12.56	13.62	13.46	12.94

Table 7: Per-language and average perplexity results for the $k = 4$ and $k = 16$ X-ELM experiments (original XGLM and $k = 1$ dense model included for comparison). Lower numbers are better. Each X-ELM setting is trained on 10.5B tokens. *TF-IDF ensemble uses more parameters for inference than other evaluations.

a 10.5B token compute budget. We find that both choices of k underperform the $k = 8$ setting.

Downstream Evaluation on Individual Languages Tables 9, 10, and 11 detail the per-language results for XNLI, XStoryCloze, and PAWS-X, respectively.

Lang.	TF-IDF Ens. top-1	m=2	m=4	m=8
AR	14.00	14.12	14.05	14.05
BG	10.27	10.27	10.27	10.27
DE	12.95	13.09	13.07	13.04
EL	9.03	9.03	8.99	9.00
EN	12.68	12.50	12.48	12.47
ES	13.54	13.40	13.39	13.37
FR	10.79	10.92	10.88	10.88
HI	14.36	13.47	13.62	13.62
JA	11.36	11.35	11.37	11.37
KO	7.61	7.53	7.53	7.53
RU	11.83	11.90	11.90	11.90
SW	19.04	18.67	18.67	18.67
TR	13.41	13.58	13.58	13.58
UR	13.26	13.52	13.52	13.52
VI	10.56	10.41	10.41	10.42
ZH	12.61	12.84	12.84	12.87
Avg.	12.33	12.29	12.29	12.28
AZ	–	736.49	724.97	722.10
HE	–	749.12	719.68	719.05
PL	–	177.31	175.27	174.83
SV	–	95.33	94.37	94.14

Table 8: Perplexity scores of the different inference methods on the TF-IDF X-ELMs trained with 21B tokens. **Top-1** chooses a single expert per language, with no routing mechanism, whereas **m=2,4,8** ensembles TF-IDF experts.

Model	AR	BG	DE	EL	EN	ES	FR	HI	RU	SW	TH*	TR	UR	VI	ZH
Zero-shot															
XGLM (1.7B)	46.8	45.7	44.1	42.5	51.5	36.5	47.2	45.9	47.3	43.6	44.9	42.5	43.5	43.9	46.9
Dense	47.9	45.0	45.3	45.2	51.1	37.2	45.9	44.5	44.5	39.6	44.3	44.8	43.1	41.6	44.6
Typ. (TRG)	46.2	44.9	43.9	45.4	52.0	36.0	47.2	43.5	41.9	40.6	–	44.2	41.9	44.4	46.3
TF-IDF (Top-1)	47.3	45.1	42.9	47.1	51.5	36.3	45.6	43.1	40.6	38.7	–	45.0	43.2	41.8	44.6
TF-IDF (Ens.)	48.6	47.2	46.2	43.1	53.0	37.0	47.5	45.7	45.6	40.0	45.8	44.1	44.2	42.6	46.6
Few-shot															
XGLM (1.7B)	42.0	44.2	43.4	43.4	47.2	38.1	45.5	40.4	43.1	41.4	41.9	38.0	39.7	42.2	44.3
Dense	43.4	42.2	43.6	41.9	45.9	36.7	42.3	42.2	40.8	40.0	43.2	39.9	40.3	41.0	43.5
Typ. (TRG)	42.8	43.0	42.6	43.0	47.3	38.5	45.4	38.9	39.9	41.7	–	41.0	39.6	42.9	43.4
Typ. (EN)	42.2	42.6	44.0	42.6	47.3	38.5	42.9	42.1	42.8	40.9	44.5	41.1	40.0	42.1	44.9
TF-IDF (Top-1)	43.1	43.6	43.2	41.7	47.5	38.2	45.3	42.1	40.5	41.9	–	41.1	41.4	42.1	44.1
TF-IDF (Ens.)	43.0	43.3	44.3	43.3	47.8	37.7	44.2	43.2	42.3	41.4	44.4	41.8	41.0	42.7	44.9

Table 9: Individual language accuracy on XNLI. *TH (Thai) is an unseen language for the X-ELM models.

Model	AR	EN	ES	EU*	HI	ID*	MY*	RU	SW	TE*	ZH
Zero-shot											
XGLM (1.7B)	53.3	63.1	57.3	56.4	55.0	59.3	54.0	60.0	60.1	57.0	55.5
Dense	50.5	60.7	56.1	52.1	52.0	55.4	53.4	58.6	58.6	55.5	56.2
Typ. (TRG)	52.3	62.7	57.5	–	52.7	–	–	60.2	60.3	–	58.8
TF-IDF (Top-1)	52.1	62.1	58.1	53.2	55.2	57.7	52.6	59.6	60.5	57.3	57.0
TF-IDF (Ens.)	51.9	60.4	57.8	54.0	55.4	58.5	52.0	59.5	60.2	57.1	57.0
Few-shot											
XGLM (1.7B)	48.6	58.2	53.2	51.7	50.4	52.1	51.5	52.5	56.0	56.5	53.7
Dense	50.2	59.0	54.6	51.3	51.6	53.5	52.9	56.9	57.8	54.2	55.2
Typ. (TRG)	50.3	60.1	55.0	–	52.0	–	–	57.4	58.0	–	56.0
Typ. (EN)	48.8	60.1	55.0	–	52.2	–	–	53.7	57.4	–	55.2
TF-IDF (Top-1)	49.3	59.5	54.5	51.4	52.4	55.2	52.9	55.4	58.0	56.1	56.1
TF-IDF (Ens.)	49.4	59.0	53.8	51.1	52.5	54.5	52.0	55.1	57.8	55.0	55.4

Table 10: Individual language accuracy on XStoryCloze (and EN StoryCloze). *Unseen languages for the X-ELM models.

Model	DE	EN	ES	FR	JA	KO	ZH
Zero-shot							
XGLM (1.7B)	44.5	47.9	51.8	45.2	53.8	49.6	47.0
Dense	49.4	47.5	50.7	47.5	48.8	47.2	48.0
Typ. (TRG)	47.9	47.9	53.0	45.5	55.4	53.6	45.7
TF-IDF (Top-1)	47.4	46.9	55.0	45.9	54.9	49.4	50.8
TF-IDF (Ens.)	49.1	47.1	52.1	47.2	53.6	50.0	50.4
Few-shot							
XGLM (1.7B)	56.3	50.5	55.4	55.2	55.6	53.0	55.7
Dense	56.0	54.9	55.8	55.2	54.9	53.8	53.0
Typ. (TRG)	56.5	53.4	55.8	55.1	55.6	55.9	55.4
Typ. (EN)	56.0	53.4	55.8	55.4	55.5	54.7	55.1
TF-IDF (Top-1)	56.6	54.2	55.7	54.9	55.6	55.7	55.7
TF-IDF (Ens.)	53.8	54.9	54.8	53.6	55.3	55.2	54.4

Table 11: Individual language accuracy on PAWS-X.