Composable Sparse Fine-Tuning for Cross-Lingual Transfer

Alan Ansell[1] **Edoardo Maria Ponti**[1,2] **Anna Korhonen**[1] **Ivan Vulić**[1]
[1]Language Technology Lab, University of Cambridge
[2]Mila - Quebec AI Institute and McGill University

Abstract

Fine-tuning the entire set of parameters of a large pretrained model has become the mainstream approach for transfer learning. To increase its efficiency and prevent catastrophic forgetting and interference, techniques like adapters and sparse fine-tuning have been developed. Adapters are *modular*, as they can be combined to adapt a model towards different facets of knowledge (e.g., dedicated language and/or task adapters). Sparse fine-tuning is *expressive*, as it controls the behavior of all model components. In this work, we introduce a new fine-tuning method with *both* these desirable properties. In particular, we learn sparse, real-valued masks based on a simple variant of the Lottery Ticket Hypothesis. Task-specific masks are obtained from annotated data in a source language, and language-specific masks from masked language modeling in a target language. Both these masks can then be composed with the pretrained model. Unlike adapter-based fine-tuning, this method neither increases the number of parameters at inference time nor alters the original model architecture. Most importantly, it outperforms adapters in zero-shot cross-lingual transfer by a large margin in a series of multilingual benchmarks, including Universal Dependencies, MasakhaNER, and AmericasNLI. Based on an in-depth analysis, we additionally find that sparsity is crucial to prevent both 1) interference between the fine-tunings to be composed and 2) overfitting. We release the code and models at `https://github.com/cambridgeltl/composable-sft`.

1 Introduction

Fine-tuning of pretrained models (Howard and Ruder, 2018; Devlin et al., 2019, *inter alia*) is arguably the dominant paradigm in NLP at present. Originally, "fine-tuning" involved supervised learning of all the parameters of a model pretrained on unlabeled texts. However, given the size of Transformer-based architectures, this approach is often time- and resource- inefficient, and may result in catastrophic forgetting and interference (Wang et al., 2020) during multiple adaptations. To overcome these limitations, two main alternatives have emerged: 1) through *adapters*, new parameters can be added to a pretrained model in the form of extra intermediate layers (Rebuffi et al., 2017; Houlsby et al., 2019) and fine-tuned while keeping all the pretrained parameters fixed; 2) *sparse* fine-tuning (SFT) of a small subset of pretrained model parameters (Guo et al., 2021; Zaken et al., 2021; Xu et al., 2021b, *inter alia*).

Adapters have proven especially useful in multilingual NLP (Bapna and Firat, 2019; Üstün et al., 2020; Pfeiffer et al., 2020b; Vidoni et al., 2020; Pfeiffer et al., 2021b; Ansell et al., 2021) because they exhibit a surprising degree of *modularity*. This ability to disentangle and recombine orthogonal facets of knowledge in original ways (Ponti et al., 2021; Ponti, 2021) allows for separately learning a task adapter from labeled data in a source language and dedicated language adapters from unlabeled data in the source language and target languages. By stacking these components, it is possible to perform zero-shot cross-lingual transfer. Compared to sequentially fine-tuning the full model on both the task and target language, this yields superior performance and efficiency (Pfeiffer et al., 2020b). Notably, achieving coverage over N_T tasks in N_L target languages with the sequential approach requires $N_T N_L$ models to be trained, whereas the modularity of adapters reduces this to $N_T + N_L$.

Meanwhile, the advantage of SFTs over adapters is their *expressivity*: rather than a non-linear transformation of the output of Transformer layers (e.g., using a shallow MLP as with adapters), they can operate directly on a pretrained model's embedding and attention layers. It therefore seems natural to search for a parameter-efficient fine-tuning method that is both modular and expressive.

Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 1778 - 1796

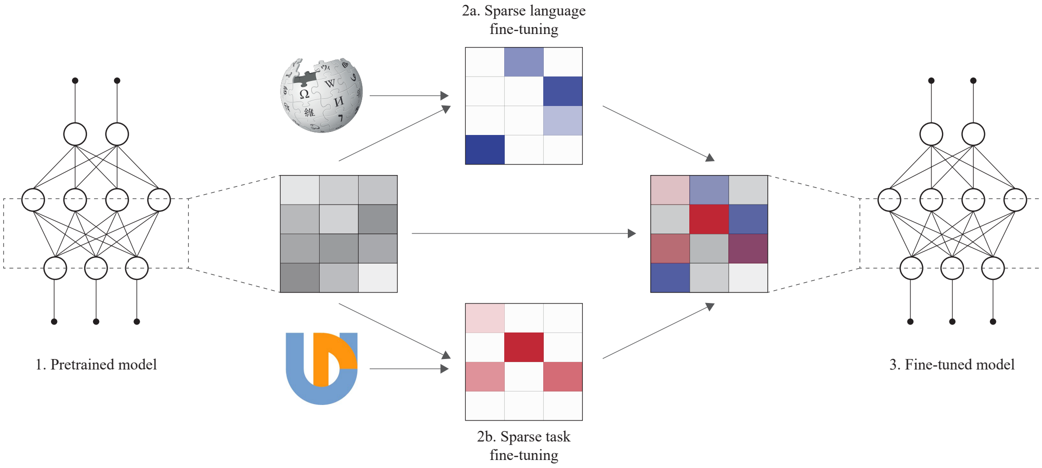


Figure 1: A graphical representation of Lottery Ticket Sparse Fine-Tuning: from the parameters of a pretrained model (gray, left), we generate sparse fine-tunings for task and language knowledge (blue and red, center). Finally, we sum these three components (right) to obtain the adapted/fine-tuned model. Best viewed in color.

To this end, we propose Lottery Ticket Sparse Fine-Tuning (LT-SFT), a simple and general-purpose adaptation technique inspired by the Lottery Ticket Hypothesis (LTH; Frankle and Carbin, 2019; Malach et al., 2020), which was originally conceived for pruning large neural networks. In particular, after fine-tuning a pretrained model for a specific task or language, we select the subset of parameters that change the most. Then, we rewind the model to its pretrained initialization (without setting any value to zero, contrary to the original LTH algorithm). By re-tuning again only the selected subset of parameters, we obtain a sparse fine-tuning in the form of a vector of differences with respect to the pretrained model. Multiple SFTs can be *composed* by simply summing them with the pretrained model. We provide a graphical representation of our method in Figure 1.

We benchmark LT-SFT on a series of multilingual datasets, including Universal Dependencies (Zeman et al., 2020) for part-of-speech tagging and dependency parsing, MasakhaNER (Adelani et al., 2021) for named entity recognition, and AmericasNLI (Ebrahimi et al., 2021) for natural language inference. We evaluate it in a zero-shot cross-lingual transfer setting on 35 typologically and geographically diverse languages that include both languages seen and unseen during masked language modeling of the pretrained model. The results in all transfer tasks indicate that LT-SFT consistently achieves substantial gains over the current state-of-the-art adapter-based method for cross-lingual transfer, MAD-X (Pfeiffer et al., 2020b).

In addition to its superior performance, modularity, and expressivity, LT-SFT offers a series of additional advantages over adapters: 1) the number of parameters remains constant, which prevents the decrease in inference speed observed when adapter layers are added; 2) the neural architecture remains identical to the pretrained model, which makes code development model-independent rather than requiring special modifications for each possible architecture (Pfeiffer et al., 2020a). Finally, 3) we empirically demonstrate that the peak in performance for LT-SFT is consistently found with the same percentage of tunable parameters, whereas the best reduction factor for MAD-X is task-dependent. This makes our method more robust to the choice of hyper-parameters.

In addition, we find that a high level of sparsity in language and task fine-tunings is beneficial to performance, as this makes overlaps less likely and poses a lower risk of creating interference between the knowledge they contain. Moreover, it makes fine-tunings less prone to overfitting due to their constrained capacity. Thus, sparsity is a fundamental ingredient for achieving modularity and composability. These properties in turn allow for systematic generalization to new combinations of tasks and languages in a zero-shot fashion.

2 Background

To establish a broader context for our research, we first provide a succinct overview of current methods for efficient fine-tuning, such as adapters and SFT. We then recapitulate the Lottery Ticket Hypothesis, upon which our newly proposed method is built.

Adapters and Composition. An *adapter* is a component inserted into a Transformer model with the purpose of specializing it for a particular language, task, domain, or modality (Houlsby et al., 2019). Previous work in multilingual NLP has mainly adopted the lightweight yet effective adapter variant of Pfeiffer et al. (2021a). In this setup, only one adapter module, consisting of a successive down-projection and up-projection, is injected per Transformer layer, after the feed-forward sub-layer. The adapter A_b at the b-th Transformer layer performs the following operation:

$$\mathrm{A}_b(\boldsymbol{h}_b, \boldsymbol{r}_b) = U_b \, \mathrm{a}(D_b \boldsymbol{h}_b) + \boldsymbol{r}_b. \tag{1}$$

$\boldsymbol{h}_b$ and $\boldsymbol{r}_b$ are the Transformer hidden state and the residual at layer b, respectively. $D_b \in \mathbb{R}^{m \times h}$ and $U_b \in \mathbb{R}^{h \times m}$ are the down- and up-projections, respectively (h being the Transformer's hidden layer size, and m the adapter's dimension), and $\mathrm{a}(\cdot)$ is a non-linear activation function. The residual connection $\boldsymbol{r}_b$ is the output of the Transformer's feed-forward layer whereas $\boldsymbol{h}_b$ is the output of the subsequent layer normalization. During fine-tuning of a pretrained model with adapters, only the adapter parameters U and D are modified while the pretrained model's parameters are kept fixed.

In the MAD-X adapter composition framework for cross-lingual transfer (Pfeiffer et al., 2020b), a *language adapter* (LA) for a massively multilingual Transformer (MMT) is learned for each source and target language through masked language modeling (MLM), and a *task adapter* (TA) is learned for each target task, where the LA for the source language is inserted during TA training. At inference time, the task adapter and target language adapter are *composed* by stacking one on top of the other. This adapter composition approach has been shown to be highly effective for cross-lingual transfer (Pfeiffer et al., 2020b, 2021b; Ansell et al., 2021), especially for low-resource languages and target languages unseen during MMT pretraining.

Sparse Fine-Tuning. We call $F' = F(\cdot; \boldsymbol{\theta} + \boldsymbol{\phi})$ a *sparse fine-tuning* (SFT) of a pretrained neural model $F(\cdot; \boldsymbol{\theta})$ if $\boldsymbol{\phi}$ is sparse. We sometimes refer to $\boldsymbol{\phi}$ itself as an SFT, or as the SFT's *difference vector*. Previously proposed SFT methods include DiffPruning (Guo et al., 2021), BitFit (Zaken et al., 2021) and ChildTuning (Xu et al., 2021b). DiffPruning simulates sparsity of the difference vector during training by applying a continuous relaxation of a binary mask to it. BitFit on the other hand allows non-zero differences only for bias parameters. ChildTuning selects a subset of fine-tunable parameters by using Fisher information to measure the relevance of each parameter to the task. These methods have been shown to be competitive with full fine-tuning on GLUE (Wang et al., 2019), despite the difference vector $\boldsymbol{\phi}$ having fewer than 0.5% non-zero values.

Lottery Ticket Hypothesis. (LTH; Frankle and Carbin, 2019; Malach et al., 2020) states that each neural model contains a sub-network (a "winning ticket") that, if trained again in isolation, can match or even exceed the performance of the original model. To achieve this, after a pruning stage where some parameters are zero-masked and frozen according to some criterion (e.g., weight magnitude), the remaining parameters are restored to their original values and then re-tuned. This process of pruning and re-training can be iterated multiple times.

The LTH has so far been used mostly for model *compression* through network pruning; to our knowledge, we are the first to use it for pretrained model *adaptation*.

Multi-Source Task Training. Ansell et al. (2021) showed that training task adapters using data from multiple source languages can result in sizable improvements in downstream zero-shot transfer performance even when the total number of training examples is held constant. In their training setup, each batch consisted of examples from a single, randomly selected source language, the language adapter for which is activated for the duration of the training step.

3 Methodology

3.1 Lottery Ticket Sparse Fine-Tuning

Training. In this work, we propose Lottery Ticket Sparse Fine-Tuning (LT-SFT). Similar to the Lottery Ticket algorithm of Frankle and Carbin (2019), our LT-SFT method consists of two phases:

(Phase 1) Pretrained model parameters $\boldsymbol{\theta}^{(0)}$ are fully fine-tuned on the target language or task data $\mathcal{D}$, yielding $\boldsymbol{\theta}^{(1)}$. Parameters are ranked according to some criterion, in our case greatest absolute difference $|\theta_i^{(1)} - \theta_i^{(0)}|$, and the top K are selected for tuning in the next phase: a binary mask $\boldsymbol{\mu}$ is set to have 1 in positions corresponding to these parameters, and 0 elsewhere.

(Phase 2) After resetting the parameters to their

original values $\boldsymbol{\theta}^{(0)}$, the model is again fine-tuned, but this time only the K selected parameters are trainable whereas the others are kept frozen. In practice, we implement this by passing the *masked* gradient $\boldsymbol{\mu} \odot \nabla_{\boldsymbol{\theta}} \mathcal{L}(F(\cdot;\boldsymbol{\theta}), \mathcal{D})$ (where $\odot$ denotes element-wise multiplication and $\mathcal{L}$ a loss function) to the optimizer at each step. From the resulting fine-tuned parameters $\boldsymbol{\theta}^{(2)}$ we can obtain the sparse vector of differences $\boldsymbol{\phi} = \boldsymbol{\theta}^{(2)} - \boldsymbol{\theta}^{(0)}$.

In addition, we experiment with applying a regularization term which discourages parameters from deviating from their pretrained values $\boldsymbol{\theta}^{(0)}$. Specifically, we use L1 regularization of the form $J(\boldsymbol{\theta}) = \frac{\lambda}{N} \sum_i |\theta_i - \theta_i^{(0)}|$.

Composition. Although we often use the term "sparse fine-tuning" to refer to the difference vector $\boldsymbol{\phi}$ itself, an SFT is most accurately conceptualized as a functional which takes as its argument a parameterized function and returns a new function, where some sparse difference vector $\boldsymbol{\phi}$ has been added to the original parameter vector. Suppose we have a language SFT S_L and a task SFT S_T defined by

$$S_L(F(\cdot;\boldsymbol{\theta})) = F(\cdot;\boldsymbol{\theta} + \boldsymbol{\phi}_L)$$
$$S_T(F(\cdot;\boldsymbol{\theta})) = F(\cdot;\boldsymbol{\theta} + \boldsymbol{\phi}_T).$$

Then we have

$$S_L \circ S_T(F(\cdot;\boldsymbol{\theta})) = F(\cdot;\boldsymbol{\theta} + \boldsymbol{\phi}_T + \boldsymbol{\phi}_L).$$

3.2 Zero-Shot Transfer with LT-SFT

We adopt a similar cross-lingual transfer setup to MAD-X (Pfeiffer et al., 2020b, see also §2). We start with an MMT F with pretrained parameters $\boldsymbol{\theta}$ learned through masked language modeling on many languages, such as mBERT (Devlin et al., 2019) or XLM-R (Conneau et al., 2020).

For each language of interest l, we learn a language SFT $\boldsymbol{\phi}_L^{(l)}$ through LT-SFT (also with an MLM objective) on text from language l.

For each task of interest t, we learn a task SFT $\boldsymbol{\phi}_T^{(t)}$ through LT-SFT on annotated data from some source language s. When learning the task SFT, we first adapt to the source language by applying the language SFT for s.[1] The language SFT is removed again after training. That is, we perform LT-SFT on $F(\cdot\,;\boldsymbol{\theta} + \boldsymbol{\phi}_L^{(s)})$ to obtain fine-tuned parameter vector $\boldsymbol{\theta}'$. We then calculate $\boldsymbol{\phi}_T^{(t)} = \boldsymbol{\theta}' - (\boldsymbol{\theta} + \boldsymbol{\phi}_L^{(s)})$. Note that during task training, we also learn a classifier head, which is fully fine-tuned during both phases of LT-SFT adaptation, with the same random initialization applied at the beginning of each phase.

We perform zero-shot adaptation of F to target language l for task t by composing language and task SFTs to obtain $F_{t,l} = F(\cdot\,;\boldsymbol{\theta} + \boldsymbol{\phi}_T^{(t)} + \boldsymbol{\phi}_L^{(l)})$. On top of this, we stack the classifier head learned for t. For a formal algorithm of LT-SFT and the transfer procedure, we refer to Appendix A.

4 Experimental Setup

To evaluate our new method extensively, we benchmark its zero-shot cross-lingual performance on four distinct tasks: part-of-speech tagging (POS), dependency parsing (DP), named entity recognition (NER), and natural language inference (NLI). Table 1 summarizes our experimental setup, including the datasets and languages considered in our experiments. We put emphasis on low-resource languages and languages unseen during MMT pretraining, although we also evaluate on a few high-resource languages. In total, we cover a set of 35 typologically and geographically diverse languages, which makes them representative of cross-lingual variation (Ponti et al., 2019, 2020).

4.1 Baselines and Model Variants

The main baseline is MAD-X, the state-of-the-art adapter-based framework for cross-lingual transfer (Pfeiffer et al., 2020b). We use the "MAD-X 2.0" variant, where the last adapter layers are dropped. Pfeiffer et al. (2021b) found that this improved performance, which we could confirm in our preliminary experiments. Since adapters with the configuration used by Pfeiffer et al. (2020b) are unavailable for many languages in our evaluation, we train our own for all languages. In Appendix D we also provide an evaluation with comparable language adapters from AdapterHub (Pfeiffer et al., 2020a) where available.

We also perform experiments with BitFit (Zaken et al., 2021) to establish a baseline for an existing SFT technique. In addition to the main LT-SFT model variant, on POS and DP we test a RAND-SFT variant as an ablation, where the K parameters to be fine-tuned are selected at random rather than based on an informed criterion.

[1]Adapting to the source language yields substantial improvements in cross-lingual transfer performance with both MAD-X and LT-SFT, with gains of 2-3 points in our preliminary experiments. Paradoxically, our results (see Table 7) and results from previous work (Pfeiffer et al., 2020b; Ansell et al., 2021) suggest that adapting to high-resource *target* languages at inference time does not give similarly large benefits. We think this phenomenon warrants further investigation.

Task	Target Dataset	Source Dataset	MMT	Target Languages
Part-of-Speech Tagging (POS), Dependency Parsing (DP)	Universal Dependencies 2.7 (Zeman et al., 2020)	Universal Dependencies 2.7 (Zeman et al., 2020)	mBERT	Arabic†, Bambara, Buryat, Cantonese, Chinese†, Erzya, Faroese, Japanese†, Livvi, Maltese, Manx, North Sami, Komi Zyrian, Sanskrit, Upper Sorbian, Uyghur
Named Entity Recognition (NER)	MasakhaNER (Adelani et al., 2021)	CoNLL 2003 (Tjong Kim Sang and De Meulder, 2003)	mBERT	Hausa, Igbo, Kinyarwanda, Luganda, Luo, Nigerian-Pidgin, Swahili*, Wolof, Yorùbá*
Natural Language Inference (NLI)	AmericasNLI (Ebrahimi et al., 2021)	MultiNLI (Williams et al., 2018)	XLM-R	Aymara, Asháninka, Bribri, Guarani, Náhuatl, Otomí, Quechua, Rarámuri, Shipibo-Konibo, Wixarika

Table 1: Details of the tasks, datasets, MMTs and languages involved in our zero-shot cross-lingual transfer evaluation. * denotes low-resource languages seen during MMT pretraining; † denotes high-resource languages seen during MMT pretraining; all other languages are low-resource and unseen. The source language is always English. Further details of all the language and data sources used are provided in Appendix B.

For both LT-SFT and MAD-X, we also evaluate a task adaptation (TA)-ONLY configuration, where only the task SFT/adapter is applied, without the target language SFT/adapter.

4.2 Language SFT/Adapter Training Setup

MLM Training Data. For all languages in our POS and DP evaluation, we perform MLM language SFT/adapter training on Wikipedia corpora. We also use Wikipedia for all languages in our NER evaluation if available. Where this is not the case, we use the Luo News Dataset (Adelani et al., 2021) for Luo and the JW300 corpus (Agić and Vulić, 2019) for Nigerian Pidgin. The main corpora for the languages in our NLI evaluation are those used by the dataset creators to train their baseline models (Ebrahimi et al., 2021); however, since the sizes of these corpora are restricted due to containing only parallel data, we augment them with data from Wikipedia and the corpora of indigenous Peruvian languages of Bustamante et al. (2020) where available. More details on data sources are provided in Appendix B.

Training Setup and Hyper-parameters. For both SFTs and adapters, we train for the lesser of 100 epochs or 100,000 steps of batch size 8 and maximum sequence length 256, subject to an absolute minimum of 30,000 steps since 100 epochs seemed insufficient for some languages with very small corpora. Model checkpoints are evaluated every 1,000 steps (5,000 for high-resource languages) on a held-out set of 5% of the corpus (1% for high-resource languages), and the one with the smallest loss is selected at the end of training. We use the AdamW optimizer (Loshchilov and Hutter, 2019) with an initial learning rate of $5e$-5 which is linearly reduced to 0 over the course of training.

Following Pfeiffer et al. (2020b), the reduction factor (i.e., the ratio between model hidden size and adapter size) for the adapter baseline was set to 2 for a total of ~7.6M trainable parameters. For comparability, we set the same number of trainable parameters K for our language LT-SFTs. This results in language SFTs with a sparsity of 4.3% for mBERT and 2.8% for XLM-R. Since BITFIT tunes exclusively the bias parameters, its language SFTs have a fixed sparsity of 0.047% for mBERT and 0.030% for XLM-R.

Importantly, during language sparse fine-tuning, we decouple the input and output embedding matrices and fix the parameters of the output matrix; otherwise, we find that the vast majority of the K most changed parameters during full fine-tuning belong to the embedding matrix, seemingly due to its proximity to the model output, which damages downstream performance. We also fix the layer normalization parameters; all other parameters are trainable. For language adaptation, we apply L1 regularization as described in §3.1 with $\lambda = 0.1$. Note that the specified training regime is applied in the same way during both phases of LT-SFT.

For language adapter training in the MAD-X baseline, we use the Pfeiffer configuration (Pfeiffer et al., 2021a) with invertible adapters, special additional sub-components designed for adapting to the vocabulary of the target language, which yields consistent gains.

4.3 Task SFT/Adapter Training Setup

For POS tagging, DP, and NER,[2] we train task SFTs/adapters on the datasets indicated in Table 1 for 10 epochs with batch size 8, except during the first phase of LT-SFT training where we train for only 3 epochs.[3] Model checkpoints are evaluated on the validation set every 250 steps, and the best checkpoint is taken at the end of training, with the selection metric being accuracy for POS, labeled attachment score for DP, and F1-score for NER. Similarly to language fine-tuning, we use an initial learning rate of $5e$-5 which is linearly reduced to 0 over the course of training. For POS and NER we use the standard token-level single-layer multi-class model head. For DP, we use the shallow variant (Glavaš and Vulić, 2021) of the biaffine dependency parser of Dozat and Manning (2017).

For NLI, we employ the same fine-tuning hyper-parameters as Ebrahimi et al. (2021): 5 epochs with batch size 32, with checkpoint evaluation on the validation set every 625 steps, and an initial learning rate of $2e$-5. We apply a two-layer multi-class classification head atop the MMT output corresponding to the `[CLS]` token.

We found that the number of trainable parameters during task adaptation (governed by K for SFTs and reduction factor for adapters) has a large effect on performance: we thus experiment with a range of values. Specifically, we test adapter reduction factors of 32, 16, 8, 4, 2, and 1, and equivalent values of K [4] for SFT.

During task adaptation, we always apply the source language adapter following Pfeiffer et al. (2020b), or source language SFT (see §3.2).

4.4 Multi-Source Training

To validate that task LT-SFT training, like task adapter training in prior work (Ansell et al., 2021), benefits from the presence of multiple source languages in the training data, and to push the boundaries of zero-shot cross lingual transfer, we perform multi-source training experiments on DP and NLI.

We adopt a similar setup to Ansell et al. (2021): we obtain the training set by concatenating the training data for all source languages. We randomly shuffle the training set and train as in the single-source case, except that each batch is composed of examples from a single source language, whose language SFT is applied during the training step.

We prioritize maximizing performance rather than providing a fair comparison against the single-source case, so unlike Ansell et al. (2021), we use the entirety of the training sets. In derogation of this principle, we set a maximum of 15K examples per language for DP to better balance our sample.

For DP, we train our models on the UD treebanks of 11 diverse high-resource languages. For NLI, we train on MultiNLI (Williams et al., 2018) plus the data for all 14 non-English languages in the XNLI dataset (Conneau et al., 2018).

We also evaluate multi-source task SFT training on extractive question answering (QA), as a comparatively generous amount of multilingual data is available for this task. Specifically, we train on English data from SQuAD version 1 (Rajpurkar et al., 2016), all languages from MLQA (Lewis et al., 2020), and those languages from XQuAD (Artetxe et al., 2020) which also appear in MLQA. We evaluate on the languages present in XQuAD but not in MLQA. For QA, we train for 5 epochs with batch size 12 and initial learning rate $3e$-5. Full details of the source languages can be found in Appendix B.

We use an equivalent reduction factor of 1 for all tasks, following the strongest setting from our single-source experiments. Except as stated above, the training configuration and hyper-parameters are the same as for single-source training.

5 Results and Discussion

We report the average test performance of zero-shot cross-lingual transfer for the best reduction factor (or equivalent K) in Table 2. Some patterns emerge across all four tasks: first, LT-SFT consistently outperforms all the baselines. In particular, it surpasses the state-of-the-art MAD-X across all tasks, with gains of 2.5 accuracy in part-of-speech tagging, 2.5 UAS and 3.7 LAS in dependency parsing, 1.8 F1 score in named entity recognition, and 1.9 accuracy in natural language inference. Compared to RAND-SFT, its superior performance demonstrates the importance of selecting "winning tickets" rather than a random subset

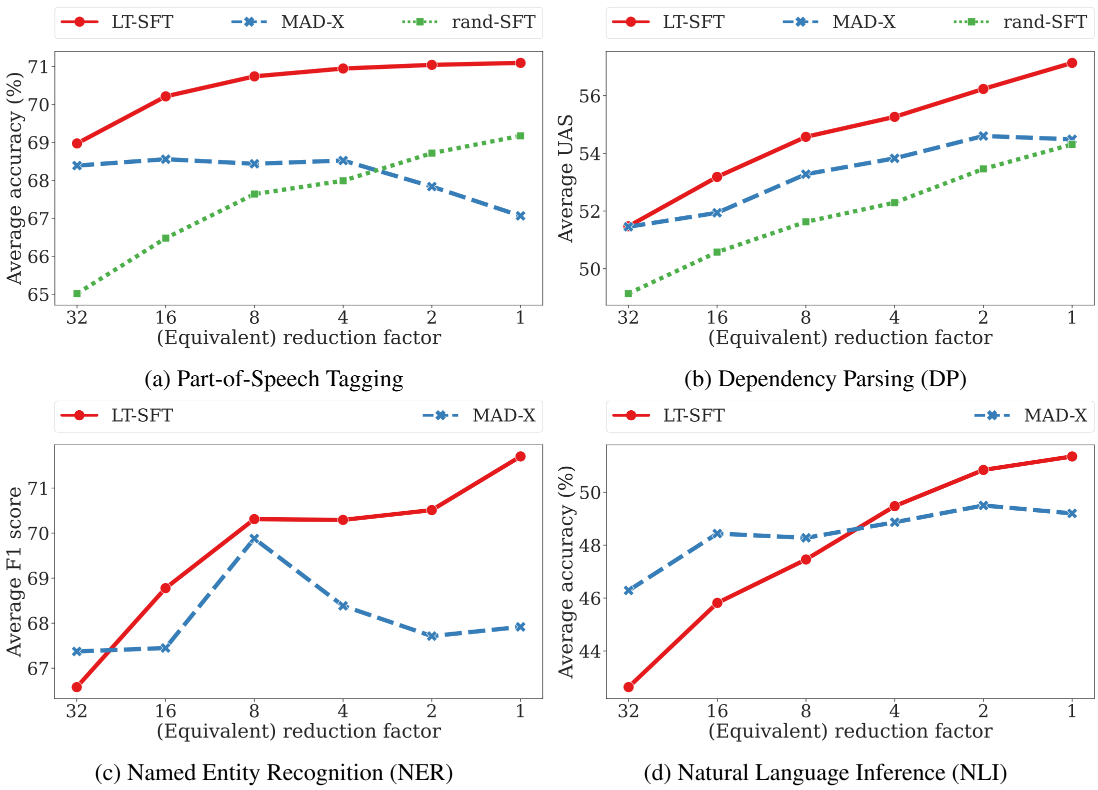
[2] MasakhaNER and CoNLL 2003 datasets respectively use the `DATE` and `MISC` tags which are not used by the other; we replace these with the `O` tag at both train and test time.

[3] This is because full fine-tuning is more prone to overfitting than sparse/adapter fine-tuning. Early stopping somewhat addresses overfitting, but it is insufficient in a cross-lingual setting because the target language performance generally starts to deteriorate faster than the source language performance.

[4] Approximately 442K, 884K, 1.7M, 3.5M, 7.1M, and 14.2M respectively, amounting to sparsity levels of 0.25%, 0.50%, 1.0%, 2.0%, 4.0% and 8.0% for mBERT and 0.16%, 0.32%, 0.63%, 1.3%, 2.6% and 5.1% for XLM-R.

	POS	DP		NER	NLI
	Accuracy	UAS	LAS	F1 score	Accuracy
LT-SFT	**71.1** (1)	**57.1** (1)	**37.8** (1)	**71.7** (1)	**51.4** (1)
RAND-SFT	69.2 (1)	54.3 (1)	33.9 (1)	-	-
MAD-X	68.6 (16)	54.6 (2)	34.1 (1)	69.9 (8)	49.5 (2)
BITFIT	58.1	45.7	23.9	54.9	38.3
LT-SFT TA-ONLY	51.3 (32)	39.1 (1)	19.9 (1)	55.3 (8)	39.9 (4)
MAD-X TA-ONLY	52.1 (32)	38.9 (1)	19.5 (1)	52.4 (32)	41.7 (4)

Table 2: Results of zero-shot cross-lingual transfer evaluation averaged over all languages when best equivalent reduction factor (shown in parentheses after each result) is chosen.



(a) Part-of-Speech Tagging

(b) Dependency Parsing (DP)

(c) Named Entity Recognition (NER)

(d) Natural Language Inference (NLI)

Figure 2: Zero-shot cross-lingual transfer evaluation of Lottery-Ticket Sparse Fine-Tuning (LT-SFT), Random Sparse Fine-Tuning (RAND-SFT), and adapter-based MAD-X over four tasks with varying numbers of trainable parameters during task adaptation. Results are averages over all target languages.

of parameters. Secondly, the results demonstrate the importance of language SFTs/adapters for specializing pretrained models to unseen languages, as they bring about a large increase in performance across the 4 tasks compared to the corresponding settings with task adaptation only (TA-ONLY).

We remark that LT-SFT's zero-shot performance also exceeds translation-based baselines on the AmericasNLI task, achieving an average accuracy of 51.4%, compared with the 48.7% of the 'translate-train' baseline of Ebrahimi et al. (2021).

In Figure 2, we provide a more detailed overview of average cross-lingual model performance across a range of different reduction factors. The results for the LT-SFT and RAND-SFT methods generally improve or stay steady as the number of trainable task parameters increases. On the contrary, there is not such a trend for MAD-X, as lower reduction factors may degrade its results. This makes it easier to choose a good setting for this hyper-parameter when using SFT. Moreover, it is worth stressing again that, contrary to MAD-X, this hyper-parameter does not affect inference time.

BITFIT performs much worse than the other methods which perform language adaptation across all tasks. Bearing in mind the strong trend towards increasing performance with increasing K for the other SFT methods, it seems likely that BITFIT, with two orders of magnitude fewer trainable parameters, lacks the capacity to learn effective task

	el	ro	ru	th	tr
XLM-R Base, full FT	71.1/54.3	78.3/63.7	74.1/57.8	67.1/55.7	67.5/51.1
XLM-R Large, full FT (Artetxe et al., 2020)	79.8/61.7	83.6/69.7	80.1/64.3	74.2/62.8	**75.9/59.3**
XLM-R Base MS, LT-SFT	**81.9/65.5**	**86.3/73.3**	**81.4/64.6**	**82.4/75.2**	75.2/58.6

Table 3: Results of zero-shot cross-lingual transfer evaluation on XQuAD (Artetxe et al., 2020), restricted to languages which do not appear in MLQA (Lewis et al., 2020) (see §4.4) in the format F1/exact match score. "Full FT" denotes full fine-tuning, MS denotes multi-source training, where additional data from MLQA and XQuAD is utilized, LT-SFT denotes Lottery-Ticket Sparse Fine-Tuning.

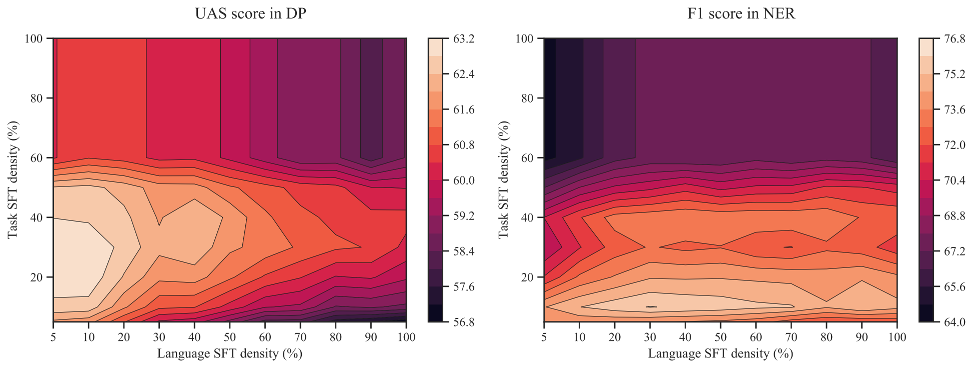


Figure 3: Performance of LT-SFT on DP and NER controlling for the sparsity of task and language fine-tuning. Results are averaged over several selected languages. Denser fine-tunings may interfere with each other and consequently degrade the model performance.

	DP UAS	DP LAS	NLI Accuracy
SINGLE SOURCE	57.1	37.8	51.4
MULTI-SOURCE	**64.3**	**47.6**	**53.1**

Table 4: Results of zero-shot cross-lingual transfer evaluation of single- vs. multi-source LT-SFT task training averaged over all target languages.

and language SFTs.

For additional results at the level of individual languages and an analysis of the efficacy of language adaptation for high- versus low- resource target languages, we refer the reader to Appendix C.

5.1 Multi-Source Training

As shown in Table 4, multi-source LT-SFT training brings about a large improvement in zero-shot cross-lingual transfer performance on DP, and a modest improvement for NLI. This may be a result of the fact that the training set for NLI contains a relatively small number of non-English examples compared to the DP training set. Also, the AmericasNLI target languages generally have a lower degree of genealogical relatedness to the source languages compared to the DP target languages.

Table 3 demonstrates that multi-source training is also beneficial to zero-shot cross-lingual transfer for QA on a series of relatively high-resource languages. In particular, LT-SFT multi-source training of XLM-R Base outperforms single-source full fine-tuning of XLM-R Large (a larger model) comfortably, and outperforms XLM-R Base single-source full fine-tuning by a significant margin. The fact that such an improvement occurs despite each of the 6 non-English source languages having more than an order of magnitude less training data than the English data from SQuAD illustrates the disproportionate advantage of multilingual source data.

5.2 Benefits of Sparsity

Finally, we address the following question: is sparsity responsible for preventing the interference of separate fine-tunings when they are composed? To support this hypothesis with empirical evidence, we use LT-SFT to train language[5] and task fine-tunings with different levels of density, i.e. the percentage of non-zero values (from 5% to 100%). We then evaluate all possible combinations of density levels. The results are visualized in the form of a contour plot in Figure 3 for selected combinations of tasks and languages: Buryat, Cantonese, Erzya, Maltese, and Upper Sorbian for DP, and Hausa, Igbo, Luganda, Swahili and Wolof for NER.

[5]To reduce computational cost, we train language fine-tunings for a maximum of 30K steps rather than the 100K of our main experiments.

From Figure 3, it emerges that the performance decreases markedly for SFTs with a density level greater than ~30% of fine-tuned parameters.[6] We speculate that this is due to the fact that sparser fine-tunings have a lower risk of overlapping with each other, thus creating interference between the different facets of knowledge they encapsulate. It must be noted, however, that alternative hypotheses could explain the performance degradation in addition to parameter overlap, such as overfitting as a result of excessive capacity. While we leave the search for conclusive evidence to future work, both of these hypotheses illustrate why enforcing sparsity in adaptation, as we propose in our method, is crucial to achieving modularity.

6 Related Work

Within the framework of the Lottery Ticket Hypothesis, a series of improvements have been suggested to make the original algorithm to find winning tickets (Frankle and Carbin, 2019) more stable: after fine-tuning, Frankle et al. (2019) rewind the parameters to their values after a few iterations rather than their values before training, whereas Renda et al. (2020) also rewind the learning rate. In addition, Zhou et al. (2019) found that 1) different criteria can be used to select weights as an alternative to the magnitude of their change; 2) different rewinding methods are also effective, such as restoring the original sign, but not the value. In future work, we will investigate whether these variants also benefit our method for cross-lingual transfer, where the LTH is used for adaptation rather than pruning.

Whereas the LTH was originally conceived in the vision domain for convolutional architectures, it is also effective for pruning models trained on NLP tasks (Yu et al., 2020), such as neural machine translation, and based on Transformer architectures (Prasanna et al., 2020). Recently, Xu et al. (2021a) adapted the LTH specifically to prune pretrained models after fine-tuning.

To the best of our knowledge, Wortsman et al. (2020) is the only instance where winning tickets were composed in previous work. In their experiment, a set of task-specific masks were linearly combined at inference time, in order to generalize to new tasks in a continuous learning setting.

[6]Note, furthermore, that levels of task fine-tuning density greater than ~60% do not vary in performance. This is because their subsets of parameters include embeddings of tokens never encountered during task training, which are therefore never updated even if trainable.

7 Conclusion and Future Work

We have presented a new method to fine-tune pretrained models that is both modular (like adapters) and expressive (like sparse fine-tuning). This method is based on a variant of the algorithm to find winning tickets under the framework of the Lottery Ticket Hypothesis. We infer a sparse vector of differences with respect to the original model for each individual language (by modeling unlabeled text) and each individual task (with supervised learning). The adaptations for a language and a task can then be composed with the pretrained model to enable zero-shot cross-lingual transfer. Comparing our method with the state-of-the-art baseline in several multilingual tasks, the results have indicated substantial gains across the board in both languages seen and unseen during pretraining (which includes many truly low-resource languages).

In future work, our method offers several potential extensions. In addition to the variants to the Lottery Ticket algorithm surveyed in §6, given the importance of sparsity for modularity (§5.2), we plan to experiment with additional algorithms previously applied to pruning that can identify and fine-tune a subset of the model parameters, such as DiffPruning (Guo et al., 2021) and ChildTuning (Xu et al., 2021b). Finally, given its simplicity and generality, our method is suited for many other applications of transfer learning in addition to cross-lingual transfer, such as multimodal learning, debiasing, and domain adaptation. The code and models are available online at `https://github.com/cambridgeltl/composable-sft`.

Acknowledgements

Alan wishes to thank David and Claudia Harding for their generous support via the Harding Distinguished Postgraduate Scholarship Programme. Anna and Ivan are supported by the ERC PoC Grant MultiConvAI (no. 957356) and a Huawei research donation. We would like to thank Chiara Ponti for the graphic illustration. We also thank the anonymous reviewers for their helpful suggestions.

References

David Ifeoluwa Adelani, Jade 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 Muhammad, Chris Chinenye Emezue, Joyce Nakatumba-

Nabende, Perez Ogayo, Anuoluwapo Aremu, Catherine Gitau, Derguene Mbaye, Jesujoba Alabi, Seid Muhie Yimam, Tajuddeen Gwadabe, Ignatius Ezeani, Rubungo Andre Niyongabo, Jonathan Mukiibi, Verrah Otiende, Iroro Orife, Davis David, Samba Ngom, Tosin 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. *arXiv preprint*.

Željko Agić and Ivan Vulić. 2019. JW300: A wide-coverage parallel corpus for low-resource languages. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3204–3210, Florence, Italy. Association for Computational Linguistics.

Alan Ansell, Edoardo Maria Ponti, Jonas Pfeiffer, Sebastian Ruder, Goran Glavaš, Ivan Vulić, and Anna Korhonen. 2021. 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)*, pages 1538–1548, Hong Kong, China. Association for Computational Linguistics.

David Brambila. 1976. *Diccionario Raramuri-Castellano: Tarahumar*.

Gina Bustamante, Arturo Oncevay, and Roberto Zariquiey. 2020. No data to crawl? monolingual corpus creation from PDF files of truly low-resource languages in Peru. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 2914–2923, Marseille, France. European Language Resources Association.

Luis Chiruzzo, Pedro Amarilla, Adolfo Ríos, and Gustavo Giménez Lugo. 2020. Development of a Guarani - Spanish parallel corpus. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 2629–2633, Marseille, France. European Language Resources Association.

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

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

Rubén Cushimariano Romano and Richer C. Sebastián Q. 2008. Ñaantsipeta asháninkaki birakochaki. diccionario asháninka-castellano. versión preliminar. http://www.lengamer.org/publicaciones/diccionarios/.

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

Timothy Dozat and Christopher D. Manning. 2017. Deep biaffine attention for neural dependency parsing. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

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.

Isaac Feldman and Rolando Coto-Solano. 2020. Neural machine translation models with back-translation for the extremely low-resource indigenous language Bribri. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3965–3976, Barcelona, Spain (Online). International Committee on Computational Linguistics.

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

Jonathan Frankle, Gintare Karolina Dziugaite, Daniel M Roy, and Michael Carbin. 2019. Stabilizing the lottery ticket hypothesis. *arXiv preprint arXiv:1903.01611*.

Ana-Paula Galarreta, Andrés Melgar, and Arturo Oncevay. 2017. Corpus creation and initial SMT experiments between Spanish and Shipibo-konibo. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 238–244, Varna, Bulgaria. INCOMA Ltd.

Goran Glavaš and Ivan Vulić. 2021. Is supervised syntactic parsing beneficial for language understanding tasks? an empirical investigation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3090–3104, Online. Association for Computational Linguistics.

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

Ximena Gutierrez-Vasques, Gerardo Sierra, and Isaac Hernandez Pompa. 2016. Axolotl: a web accessible parallel corpus for Spanish-Nahuatl. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4210–4214, Portorož, Slovenia. European Language Resources Association (ELRA).

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-Efficient Transfer Learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. 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.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Manuel Mager, Diónico Carrillo, and Ivan Meza. 2018. Probabilistic finite-state morphological segmenter for wixarika (huichol) language. *Journal of Intelligent & Fuzzy Systems*, 34(5):3081–3087.

Eran Malach, Gilad Yehudai, Shai Shalev-Schwartz, and Ohad Shamir. 2020. Proving the lottery ticket hypothesis: Pruning is all you need. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 6682–6691. PMLR.

Elena Mihas. 2011. *Añaani katonkosatzi parenini, El idioma del alto Perené*. Milwaukee, WI: Clarks Graphics.

John E Ortega, Richard Alexander Castro-Mamani, and Jaime Rafael Montoya Samame. 2020. Overcoming resistance: The normalization of an Amazonian tribal language. In *Proceedings of the 3rd Workshop on Technologies for MT of Low Resource Languages*, pages 1–13, Suzhou, China. Association for Computational Linguistics.

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

Jonas Pfeiffer, 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*, pages 46–54, Online. Association for Computational Linguistics.

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)*.

Edoardo Ponti, Ivan Vulić, Ryan Cotterell, Marinela Parovic, Roi Reichart, and Anna Korhonen. 2021. Parameter space factorization for zero-shot learning across tasks and languages. *Transactions of the Association for Computational Linguistics*, 9(0):410–428.

Edoardo Maria Ponti. 2021. *Inductive Bias and Modular Design for Sample-Efficient Neural Language Learning*. Ph.D. thesis, University of Cambridge.

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.

Edoardo Maria Ponti, Helen O'Horan, Yevgeni Berzak, Ivan Vulić, Roi Reichart, Thierry Poibeau, Ekaterina Shutova, and Anna Korhonen. 2019. Modeling language variation and universals: A survey on typological linguistics for natural language processing. *Computational Linguistics*, 45(3):559–601.

Sai Prasanna, Anna Rogers, and Anna Rumshisky. 2020. When BERT Plays the Lottery, All Tickets Are Winning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3208–3229, Online. Association for Computational Linguistics.

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*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. 2017. Learning Multiple Visual Domains with Residual Adapters. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Alex Renda, Jonathan Frankle, and Michael Carbin. 2020. Comparing rewinding and fine-tuning in neural network pruning. In *International Conference on Learning Representations*.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2214–2218, Istanbul, Turkey. European Language Resources Association (ELRA).

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

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.

Marko Vidoni, Ivan Vulić, and Goran Glavaš. 2020. Orthogonal language and task adapters in zero-shot cross-lingual transfer. *CoRR*, abs/2012.06460.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*.

Zirui Wang, Zachary C. Lipton, and Yulia Tsvetkov. 2020. On negative interference in multilingual models: Findings and a meta-learning treatment. In *Proceedings of EMNLP 2020*, pages 4438–4450.

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.

Mitchell Wortsman, Vivek Ramanujan, Rosanne Liu, Aniruddha Kembhavi, Mohammad Rastegari, Jason Yosinski, and Ali Farhadi. 2020. Supermasks in superposition. In *Advances in Neural Information Processing Systems*, volume 33, pages 15173–15184. Curran Associates, Inc.

Dongkuan Xu, Ian En-Hsu Yen, Jinxi Zhao, and Zhibin Xiao. 2021a. Rethinking network pruning – under the pre-train and fine-tune paradigm. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2376–2382, Online. Association for Computational Linguistics.

Runxin Xu, Fuli Luo, Zhiyuan Zhang, Chuanqi Tan, Baobao Chang, Songfang Huang, and Fei Huang. 2021b. Raise a child in large language model: Towards effective and generalizable fine-tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9514–9528, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Haonan Yu, Sergey Edunov, Yuandong Tian, and Ari S. Morcos. 2020. Playing the lottery with rewards and multiple languages: lottery tickets in rl and nlp. In *International Conference on Learning Representations*.

Elad Ben Zaken, Shauli Ravfogel, and Yoav Goldberg. 2021. BitFit: Simple Parameter-efficient Fine-tuning for Transformer-based Masked Language-models. *CoRR*, abs/2106.10199.

Daniel Zeman, Joakim Nivre, Mitchell Abrams, Elia Ackermann, Noëmi Aepli, Hamid Aghaei, Željko Agić, Amir Ahmadi, Lars Ahrenberg, Chika Kennedy Ajede, Gabrielė Aleksandravičiūtė,

Ika Alfina, Lene Antonsen, Katya Aplonova, Angelina Aquino, Carolina Aragon, Maria Jesus Aranzabe, Ĥórunn Arnardóttir, Gashaw Arutie, Jessica Naraiswari Arwidarasti, Masayuki Asahara, Luma Ateyah, Furkan Atmaca, Mohammed Attia, Aitziber Atutxa, Liesbeth Augustinus, Elena Badmaeva, Keerthana Balasubramani, Miguel Ballesteros, Esha Banerjee, Sebastian Bank, Verginica Barbu Mititelu, Victoria Basmov, Colin Batchelor, John Bauer, Seyyit Talha Bedir, Kepa Bengoetxea, Gözde Berk, Yevgeni Berzak, Irshad Ahmad Bhat, Riyaz Ahmad Bhat, Erica Biagetti, Eckhard Bick, Agnė Bielinskienė, Kristín Bjarnadóttir, Rogier Blokland, Victoria Bobicev, Loïc Boizou, Emanuel Borges Völker, Carl Börstell, Cristina Bosco, Gosse Bouma, Sam Bowman, Adriane Boyd, Kristina Brokaitė, Aljoscha Burchardt, Marie Candito, Bernard Caron, Gauthier Caron, Tatiana Cavalcanti, Gülşen Cebiroğlu Eryiğit, Flavio Massimiliano Cecchini, Giuseppe G. A. Celano, Slavomír Čéplö, Savas Cetin, Özlem Çetinoğlu, Fabricio Chalub, Ethan Chi, Yongseok Cho, Jinho Choi, Jayeol Chun, Alessandra T. Cignarella, Silvie Cinková, Aurélie Collomb, Çağrı Çöltekin, Miriam Connor, Marine Courtin, Elizabeth Davidson, Marie-Catherine de Marneffe, Valeria de Paiva, Mehmet Oguz Derin, Elvis de Souza, Arantza Diaz de Ilarraza, Carly Dickerson, Arawinda Dinakaramani, Bamba Dione, Peter Dirix, Kaja Dobrovoljc, Timothy Dozat, Kira Droganova, Puneet Dwivedi, Hanne Eckhoff, Marhaba Eli, Ali Elkahky, Binyam Ephrem, Olga Erina, Tomaž Erjavec, Aline Etienne, Wograine Evelyn, Sidney Facundes, Richárd Farkas, Marília Fernanda, Hector Fernandez Alcalde, Jennifer Foster, Cláudia Freitas, Kazunori Fujita, Katarína Gajdošová, Daniel Galbraith, Marcos Garcia, Moa Gärdenfors, Sebastian Garza, Fabrício Ferraz Gerardi, Kim Gerdes, Filip Ginter, Iakes Goenaga, Koldo Gojenola, Memduh Gökırmak, Yoav Goldberg, Xavier Gómez Guinovart, Berta González Saavedra, Bernadeta Griciūtė, Matias Grioni, Loïc Grobol, Normunds Grūzītis, Bruno Guillaume, Céline Guillot-Barbance, Tunga Güngör, Nizar Habash, Hinrik Hafsteinsson, Jan Hajič, Jan Hajič jr., Mika Hämäläinen, Linh Hà Mỹ, Na-Rae Han, Muhammad Yudistira Hanifmuti, Sam Hardwick, Kim Harris, Dag Haug, Johannes Heinecke, Oliver Hellwig, Felix Hennig, Barbora Hladká, Jaroslava Hlaváčová, Florinel Hociung, Petter Hohle, Eva Huber, Jena Hwang, Takumi Ikeda, Anton Karl Ingason, Radu Ion, Elena Irimia, Ọlájídé Ishola, Tomáš Jelínek, Anders Johannsen, Hildur Jónsdóttir, Fredrik Jørgensen, Markus Juutinen, Sarveswaran K, Hüner Kaşıkara, Andre Kaasen, Nadezhda Kabaeva, Sylvain Kahane, Hiroshi Kanayama, Jenna Kanerva, Boris Katz, Tolga Kayadelen, Jessica Kenney, Václava Kettnerová, Jesse Kirchner, Elena Klementieva, Arne Köhn, Abdullatif Köksal, Kamil Kopacewicz, Timo Korkiakangas, Natalia Kotsyba, Jolanta Kovalevskaitė, Simon Krek, Parameswari Krishnamurthy, Sookyoung Kwak, Veronika Laippala, Lucia Lam, Lorenzo Lambertino, Tatiana Lando, Septina Dian Larasati, Alexei Lavrentiev, John Lee, Phương Lê Hồng, Alessandro Lenci, Saran Lertpradit, Herman Leung, Maria Levina, Cheuk Ying Li, Josie Li, Keying Li, Yuan Li, KyungTae Lim, Krister Lindén, Nikola Ljubešić, Olga Loginova, Andry Luthfi, Mikko Luukko, Olga Lyashevskaya, Teresa Lynn, Vivien Macketanz, Aibek Makazhanov, Michael Mandl, Christopher Manning, Ruli Manurung, Cătălina Mărănduc, David Mareček, Katrin Marheinecke, Héctor Martínez Alonso, André Martins, Jan Mašek, Hiroshi Matsuda, Yuji Matsumoto, Ryan McDonald, Sarah McGuinness, Gustavo Mendonça, Niko Miekka, Karina Mischenkova, Margarita Misirpashayeva, Anna Missilä, Cătălin Mititelu, Maria Mitrofan, Yusuke Miyao, AmirHossein Mojiri Foroushani, Amirsaeid Moloodi, Simonetta Montemagni, Amir More, Laura Moreno Romero, Keiko Sophie Mori, Shinsuke Mori, Tomohiko Morioka, Shigeki Moro, Bjartur Mortensen, Bohdan Moskalevskyi, Kadri Muischnek, Robert Munro, Yugo Murawaki, Kaili Müürisep, Pinkey Nainwani, Mariam Nakhlé, Juan Ignacio Navarro Horñiacek, Anna Nedoluzhko, Gunta Nešpore-Bērzkalne, Lương Nguyễn Thị, Huyền Nguyễn Thị Minh, Yoshihiro Nikaido, Vitaly Nikolaev, Rattima Nitisaroj, Alireza Nourian, Hanna Nurmi, Stina Ojala, Atul Kr. Ojha, Adédayọ‘ Olúòkun, Mai Omura, Emeka Onwuegbuzia, Petya Osenova, Robert Östling, Lilja Øvrelid, Şaziye Betül Özateş, Arzucan Özgür, Balkız Öztürk Başaran, Niko Partanen, Elena Pascual, Marco Passarotti, Agnieszka Patejuk, Guilherme Paulino-Passos, Angelika Peljak-Łapińska, Siyao Peng, Cenel-Augusto Perez, Natalia Perkova, Guy Perrier, Slav Petrov, Daria Petrova, Jason Phelan, Jussi Piitulainen, Tommi A Pirinen, Emily Pitler, Barbara Plank, Thierry Poibeau, Larisa Ponomareva, Martin Popel, Lauma Pretkalniņa, Sophie Prévost, Prokopis Prokopidis, Adam Przepiórkowski, Tiina Puolakainen, Sampo Pyysalo, Peng Qi, Andriela Rääbis, Alexandre Rademaker, Taraka Rama, Loganathan Ramasamy, Carlos Ramisch, Fam Rashel, Mohammad Sadegh Rasooli, Vinit Ravishankar, Livy Real, Petru Rebeja, Siva Reddy, Georg Rehm, Ivan Riabov, Michael Rießler, Erika Rimkutė, Larissa Rinaldi, Laura Rituma, Luisa Rocha, Eiríkur Rögnvaldsson, Mykhailo Romanenko, Rudolf Rosa, Valentin Roșca, Davide Rovati, Olga Rudina, Jack Rueter, Kristján Rúnarsson, Shoval Sadde, Pegah Safari, Benoît Sagot, Aleksi Sahala, Shadi Saleh, Alessio Salomoni, Tanja Samardžić, Stephanie Samson, Manuela Sanguinetti, Dage Särg, Baiba Saulīte, Yanin Sawanakunanon, Kevin Scannell, Salvatore Scarlata, Nathan Schneider, Sebastian Schuster, Djamé Seddah, Wolfgang Seeker, Mojgan Seraji, Mo Shen, Atsuko Shimada, Hiroyuki Shirasu, Muh Shohibussirri, Dmitry Sichinava, Einar Freyr Sigurðsson, Aline Silveira, Natalia Silveira, Maria Simi, Radu Simionescu, Katalin Simkó, Mária Šimková, Kiril Simov, Maria Skachedubova, Aaron Smith, Isabela Soares-Bastos, Carolyn Spadine, Steinĥór Steingrímsson, Antonio Stella, Milan Straka, Emmett Strickland, Jana Strnadová, Alane Suhr, Yogi Lesmana Sulestio, Umut Sulubacak, Shingo Suzuki,

Zsolt Szántó, Dima Taji, Yuta Takahashi, Fabio Tamburini, Mary Ann C. Tan, Takaaki Tanaka, Samson Tella, Isabelle Tellier, Guillaume Thomas, Liisi Torga, Marsida Toska, Trond Trosterud, Anna Trukhina, Reut Tsarfaty, Utku Türk, Francis Tyers, Sumire Uematsu, Roman Untilov, Zdeňka Urešová, Larraitz Uria, Hans Uszkoreit, Andrius Utka, Sowmya Vajjala, Daniel van Niekerk, Gertjan van Noord, Viktor Varga, Eric Villemonte de la Clergerie, Veronika Vincze, Aya Wakasa, Joel C. Wallenberg, Lars Wallin, Abigail Walsh, Jing Xian Wang, Jonathan North Washington, Maximilan Wendt, Paul Widmer, Seyi Williams, Mats Wirén, Christian Wittern, Tsegay Woldemariam, Tak-sum Wong, Alina Wróblewska, Mary Yako, Kayo Yamashita, Naoki Yamazaki, Chunxiao Yan, Koichi Yasuoka, Marat M. Yavrumyan, Zhuoran Yu, Zdeněk Žabokrtský, Shorouq Zahra, Amir Zeldes, Hanzhi Zhu, and Anna Zhuravleva. 2020. Universal Dependencies 2.7. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Hattie Zhou, Janice Lan, Rosanne Liu, and Jason Yosinski. 2019. Deconstructing lottery tickets: Zeros, signs, and the supermask. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

A Algorithm of Cross-Lingual Transfer with LT-SFT

Algorithm 1 Cross-Lingual Transfer with Lottery-Ticket Sparse Fine-Tuning

function $\textsc{LtSft}(\mathcal{D}, \mathcal{L}, \boldsymbol{\theta}^{(0)}, \eta, K)$
  $\boldsymbol{\theta}^{(1)} \leftarrow \boldsymbol{\theta}^{(0)}$
  while not converged **do**
    $\boldsymbol{\theta}^{(1)} \leftarrow \boldsymbol{\theta}^{(1)} - \eta \nabla \mathcal{L}(\boldsymbol{\theta}^{(1)}, \mathcal{D})$
  $\mu_i \leftarrow \begin{cases} 1 & \text{if } \theta_i^{(1)} \in \mathrm{argmax}_{\theta_1, \ldots, \theta_K} |\boldsymbol{\theta}^{(1)} - \boldsymbol{\theta}^{(0)}| \\ 0 & \text{otherwise} \end{cases}$
  $\boldsymbol{\theta}^{(2)} \leftarrow \boldsymbol{\theta}^{(0)}$
  while not converged **do**
    $\boldsymbol{\theta}^{(2)} \leftarrow \boldsymbol{\theta}^{(2)} - \boldsymbol{\mu} \odot \eta \nabla \mathcal{L}(\boldsymbol{\theta}^{(2)}, \mathcal{D})$
  $\boldsymbol{\phi} \leftarrow \boldsymbol{\theta}^{(2)} - \boldsymbol{\theta}^{(0)}$
  return $\boldsymbol{\phi}$
end function

function $\textsc{CrossLingualTransfer}(\mathcal{D}_{\text{src}}, \mathcal{D}_{\text{tar}}, \mathcal{D}_{\text{task}}, \mathcal{L}_{\text{task}}, \boldsymbol{\theta}^{(0)}, \eta, K)$
  $\boldsymbol{\phi}_{\text{src}} \leftarrow \textsc{LtSft}(\mathcal{D}_{\text{src}}, \mathcal{L}_{\text{MLM}}, \boldsymbol{\theta}^{(0)}, \eta, K)$
  $\boldsymbol{\phi}_{\text{task}} \leftarrow \textsc{LtSft}(\mathcal{D}_{\text{task}}, \mathcal{L}_{\text{task}}, \boldsymbol{\theta}^{(0)} + \boldsymbol{\phi}_{\text{src}}, \eta, K)$
  $\boldsymbol{\phi}_{\text{tar}} \leftarrow \textsc{LtSft}(\mathcal{D}_{\text{tar}}, \mathcal{L}_{\text{MLM}}, \boldsymbol{\theta}^{(0)}, \eta, K)$
  return $\boldsymbol{\theta}^{(0)} + \boldsymbol{\phi}_{\text{task}} + \boldsymbol{\phi}_{\text{tar}}$
end function

B Languages

Task	Language	ISO Code	Family	UD Treebank	Corpus source(s)
Source	Arabic[†,‡]	ar	Afro-Asiatic, Semitic		Wikipedia
	Basque[*]	eu	Basque	BDT	
	Bulgarian[†]	bg	Indo-European, Slavic		
	Chinese[†,‡]	zh	Sino-Tibetan		
	Czech[*]	cs	Indo-European, Slavic	PDT	
	English[*,†,‡],	en	Indo-European, Germanic	EWT	
	Estonian[*]	et	Uralic, Finnic	EDT	
	French[*,†]	fr	Indo-European, Romance	GSD	
	German[†,‡]	de	Indo-European, Germanic		
	Greek[*,†]	el	Indo-European, Greek	GDT	
	Hindi[*,†,‡]	hi	Indo-European, Indic	HDTB	
	Korean[*]	ko	Korean	GSD	
	Persian[*]	fa	Indo-European, Iranian	PerDT	
	Russian[†]	ru	Indo-European, Slavic		
	Spanish[†,‡]	es	Indo-European, Romance		
	Swahili[†]	swa	Niger-Congo, Bantoid		
	Thai[†]	th	Tai-Kadai, Kam-Thai		
	Turkish[*,†]	tr	Turkic, Southwestern	BOUN	
	Urdu[†]	ur	Indo-European, Indic		
	Vietnamese[*,‡]	vi	Austro-Asiatic, Viet-Muong	VTB	
POS/DP	Arabic	ar	Afro-Asiatic, Semitic	PUD	Wikipedia
	Bambara	bm	Mande	CRB	
	Buryat	bxr	Mongolic	BDT	
	Cantonese	yue	Sino-Tibetan	HK	
	Chinese	zh	Sino-Tibetan	GSD	
	Erzya	myv	Uralic, Mordvin	JR	
	Faroese	fo	Indo-European, Germanic	FarPaHC	
	Japanese	ja	Japanese	GSD	
	Livvi	olo	Uralic, Finnic	KKPP	
	Maltese	mt	Afro-Asiatic, Semitic	MUDT	
	Manx	gv	Indo-European, Celtic	Cadhan	
	North Sami	sme	Uralic, Sami	Giella	
	Komi Zyrian	kpv	Uralic, Permic	Lattice	
	Sanskrit	sa	Indo-European, Indic	UFAL	
	Upper Sorbian	hsb	Indo-European, Slavic	UFAL	
	Uyghur	ug	Turkic, Southeastern	UDT	
NER	Hausa	hau	Afro-Asiatic, Chadic	N/A	Wikipedia
	Igbo	ibo	Niger-Congo, Volta-Niger		Wikipedia
	Kinyarwanda	kin	Niger-Congo, Bantu		Wikipedia
	Luganda	lug	Niger-Congo, Bantu		Wikipedia
	Luo	luo	Nilo-Saharan		Luo News Dataset (Adelani et al., 2021)
	Nigerian-Pidgin	pcm	English Creole		JW300 (Agić and Vulić, 2019)
	Swahili	swa	Niger-Congo, Bantu		Wikipedia
	Wolof	wol	Niger-Congo, Senegambian		Wikipedia
	Yorùbá	yor	Niger-Congo, Volta-Niger		Wikipedia
NLI	Aymara	aym	Aymaran	N/A	Tiedemann (2012); Wikipedia
	Asháninka	cni	Arawakan		Ortega et al. (2020); Cushimariano Romano and Sebastián Q. (2008); Mihas (2011); Bustamante et al. (2020)
	Bribri	bzd	Chibchan, Talamanca		Feldman and Coto-Solano (2020)
	Guarani	gn	Tupian, Tupi-Guarani		Chiruzzo et al. (2020); Wikipedia
	Náhuatl	nah	Uto-Aztecan, Aztecan		Gutierrez-Vasques et al. (2016); Wikipedia
	Otomí	oto	Oto-Manguean, Otomian		Hñähñu Online Corpus
	Quechua	quy	Quechuan		Agić and Vulić (2019); Wikipedia
	Rarámuri	tar	Uto-Aztecan, Tarahumaran		Brambila (1976)
	Shipibo-Konibo	shp	Panoan		Galarreta et al. (2017); Bustamante et al. (2020)
	Wixarika	hch	Uto-Aztecan, Corachol		Mager et al. (2018)
QA	Greek	el	Indo-European, Greek	N/A	Wikipedia
	Romanian	ro	Indo-European, Romance		
	Russian	ru	Indo-European, Slavic		
	Thai	th	Tai-Kadai, Kam-Tai		
	Turkish	tr	Turkic, Southwestern		

Table 5: Details of the languages and data used for training and evaluation of SFTs and adapters. The corpora of Bustamante et al. (2020) are available at `https://github.com/iapucp/multilingual-data-peru`; all other NLI corpora mentioned are available at `https://github.com/AmericasNLP/americasnlp2021`. * denotes source languages for multi-source DP training; † denotes source languages for multi-source NLI training; ‡ denotes source languages for multi-source QA training. English is the source language in all single-source task training experiments.

C Results by Language

	LT-SFT	RAND-SFT	MAD-X	BITFIT	LT-SFT TA	MAD-X TA
ar	68.7	69.3	70.1	69.8	70.6	**70.8**
bm	**57.0**	55.6	51.0	41.7	34.2	37.2
bxr	**73.2**	71.4	71.9	64.2	59.5	62.0
fo	**87.9**	86.5	85.7	77.3	72.9	74.1
gv	**72.0**	68.4	66.9	44.3	35.4	37.5
hsb	**83.1**	82.4	81.8	77.2	69.2	69.6
ja	53.9	**54.3**	51.1	53.9	54.1	51.2
kpv	**61.8**	56.0	58.5	39.6	37.1	35.8
mt	**80.6**	77.6	73.7	53.6	32.6	30.9
myv	**80.3**	71.5	75.6	54.7	45.7	48.5
olo	**82.3**	81.7	79.7	73.1	62.2	63.4
sa	**65.3**	63.2	60.9	50.3	39.8	45.0
sme	**78.0**	70.4	72.0	50.6	43.3	39.4
ug	59.1	**64.7**	63.7	43.2	34.0	36.8
yue	**66.8**	65.6	**66.8**	66.2	64.5	64.1
zh	67.5	68.0	67.6	**69.2**	65.9	67.6
avg	**71.1**	69.2	68.6	58.1	51.3	52.1

(a) POS accuracy (%)

	LT-SFT	RAND-SFT	MAD-X	BITFIT	LT-SFT TA	MAD-X TA	LT-SFT MS
ar	**70.8/53.6**	68.7/51.6	69.5/51.5	64.0/48.6	68.7/53.0	68.6/52.3	81.5/69.8
bm	**43.1/16.5**	39.3/14.8	39.1/13.6	33.3/8.1	30.0/7.8	29.9/6.8	46.4/20.6
bxr	**49.2/25.9**	48.3/24.1	48.3/24.0	44.9/19.7	40.7/17.3	41.0/18.0	60.2/35.4
fo	**68.2/55.5**	65.7/53.1	66.3/52.5	57.7/43.4	54.3/39.8	53.6/38.5	67.2/55.6
gv	60.0/**42.4**	59.0/39.1	**61.2**/37.0	43.3/14.7	28.1/5.0	26.4/5.4	66.1/52.0
hsb	**73.7**/60.5	72.1/58.7	72.1/**61.1**	61.7/47.7	55.4/42.1	53.5/40.9	87.0/79.5
ja	**36.9/19.7**	34.8/18.9	33.0/18.9	34.4/18.8	36.0/19.3	33.8/18.3	44.0/26.9
kpv	**50.5/27.2**	45.1/20.7	47.3/22.6	35.8/11.3	24.7/7.5	25.4/7.1	57.1/35.9
mt	**74.6/55.4**	68.9/48.8	69.4/50.8	51.0/25.0	29.2/5.7	28.9/5.0	81.0/67.9
myv	**65.9/45.3**	59.8/36.3	59.6/35.7	42.2/17.2	32.1/11.7	30.3/10.4	73.8/57.4
olo	**66.4/47.8**	64.5/43.1	60.9/42.0	52.4/29.3	42.2/20.0	42.5/18.3	74.9/62.4
sa	**49.5/25.2**	48.9/20.8	46.8/19.5	42.8/13.9	32.5/8.7	36.0/9.9	62.1/39.5
sme	**58.0/42.1**	49.9/29.6	50.6/29.0	31.7/10.7	23.2/7.0	22.3/6.6	63.4/50.7
ug	36.4/16.7	37.3/15.8	**42.1/19.2**	35.3/13.5	21.9/7.7	23.5/8.4	56.3/35.9
yue	**51.1/34.0**	48.7/31.2	48.8/31.8	44.5/27.0	47.4/30.0	47.0/29.4	52.1/36.3
zh	**59.8**/37.0	58.2/35.6	58.5/**37.2**	55.9/33.7	58.4/36.3	59.1/36.9	55.3/35.9
avg	**57.1/37.8**	54.3/33.9	54.6/34.1	45.7/23.9	39.1/19.9	38.9/19.5	64.3/47.6

(b) DP UAS/LAS

	LT-SFT	MAD-X	BITFIT	LT-SFT TA	MAD-X TA
hau	**83.5**	83.4	50.2	46.5	44.0
ibo	**76.7**	71.7	57.2	56.8	54.5
kin	**67.4**	65.3	56.0	52.9	50.2
lug	**67.9**	67.0	50.9	53.8	53.3
luo	**54.7**	52.2	35.6	37.7	33.0
pcm	**74.6**	72.1	66.8	74.4	71.0
swa	**79.4**	77.6	67.4	69.5	69.6
wol	**66.3**	65.6	45.0	37.1	29.8
yor	**74.8**	74.0	64.7	69.3	66.6
avg	**71.7**	69.9	54.9	55.3	52.4

(c) NER F1-score

	LT-SFT	MAD-X	BITFIT	LT-SFT TA	MAD-X TA	LT-SFT MS
aym	**57.9**	51.6	40.8	38.3	40.7	59.9
bzd	**44.4**	44.0	36.7	37.1	38.3	46.3
cni	**47.9**	47.6	34.5	40.9	44.1	50.3
gn	**63.5**	58.8	46.4	44.8	43.3	69.1
hch	**42.9**	41.5	36.3	38.4	40.7	44.4
nah	52.7	**53.7**	38.8	41.6	44.2	53.8
oto	**48.5**	46.8	39.8	39.7	40.8	43.3
quy	**62.0**	58.3	34.5	38.3	41.5	68.4
shp	**50.3**	48.9	38.8	42.1	44.4	53.2
tar	43.5	**43.9**	36.7	37.6	38.8	42.5
avg	**51.4**	49.5	38.3	39.9	41.7	53.1

(d) NLI accuracy (%)

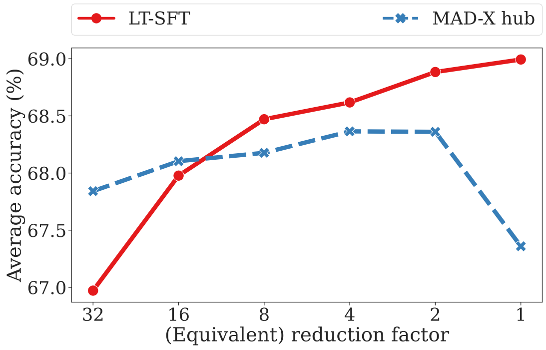
Table 6: Results achieved by various zero-shot cross-lingual transfer methods across all tasks for each language. For each (method, task) pair, the (equivalent) reduction factor with the best mean score is selected as shown in Table 2. LT-SFT MS denotes LT-SFT with multi-source training. **Bold** denotes best-performing method per language, excluding LT-SFT MS as its larger, more diverse dataset gives it an unfair advantage.

	POS (accuracy)				DP (UAS)				NER (F1)		
	ar	ja	zh	avg.	ar	ja	zh	avg.	swa	yor	avg.
LT-SFT	68.7	53.9	67.5	63.4	70.8	36.9	59.8	55.9	79.4	74.8	77.1
RAND-SFT	69.3	54.3	68.0	63.9	68.7	34.8	58.2	53.9	-	-	-
MAD-X	70.1	51.1	67.6	62.9	69.5	33.0	58.5	53.7	77.6	74.0	75.8
BITFIT	69.8	53.9	69.2	64.3	64.0	34.3	55.9	51.4	67.4	64.7	66.0
LT-SFT TA-ONLY	70.6	54.1	65.9	63.5	68.7	36.0	58.4	54.4	69.5	69.3	69.4
MAD-X TA-ONLY	70.8	51.2	67.6	63.2	68.6	33.8	59.1	53.8	69.6	66.6	68.1

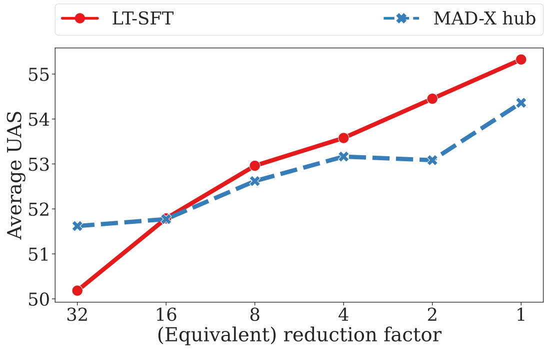
Table 7: Results for zero-shot cross-lingual transfer evaluation of the seen languages included in the POS, DP and NER evaluations. For each method/metric pair, the best equivalent reduction factor from Table 2 is used.

Arabic, Japanese and Chinese, which were included in the POS/DP evaluation, can be considered high-resource languages; Swahili and Yorùbá, on the other hand, were included in the NER evaluation and are arguably resource-poor. In keeping with previous work, we find that language adaptation benefits seen languages less than unseen languages and—among the former—resource-rich languages less than resource-poor languages. This agrees with the intuition that lower-resource languages have greater scope for improvement through language adaptation due to the fact that they receive less signal during MMT pretraining. Interestingly, BITFIT performs much more competitively on the high-resource languages than low-resource and unseen languages, suggesting that its lack of capacity is more problematic for language adaptation rather than for task fine-tuning.

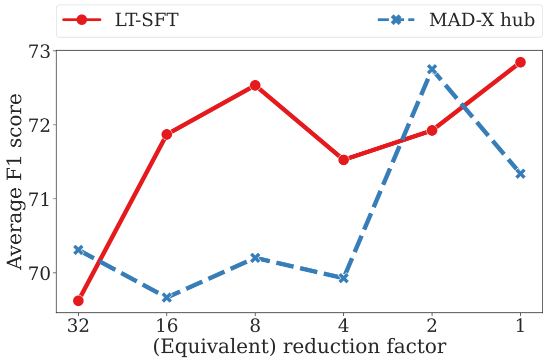
D MAD-X Results with AdapterHub Adapters



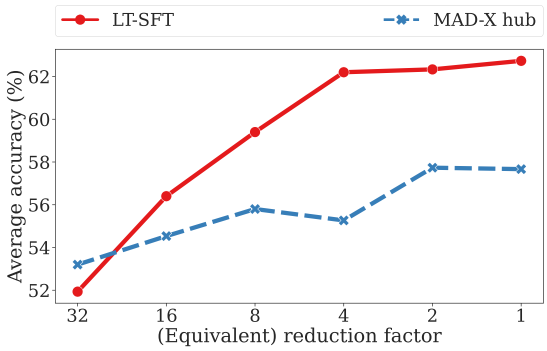
(a) Part-of-Speech Tagging
(ar, bxr, ja, kpv, mt, myv, sme, ug, yue, zh)



(b) Dependency Parsing (DP)
(ar, bxr, ja, kpv, mt, myv, sme, ug, yue, zh)



(c) Named Entity Recognition (NER) (swa, wol)



(d) Natural Language Inference (NLI) (gn, quy)

Figure 4: Zero-shot cross-lingual transfer evaluation of Lottery-Ticket Sparse Fine-Tuning (LT-SFT) and MAD-X when pretrained language adapters from AdapterHub (Pfeiffer et al., 2020a) are used during task training and evaluation. These adapters are trained for 250,000 steps with a batch size of 64, as opposed to the 100,000 steps of batch size 8 used in our experiments. LT-SFT nevertheless maintains an edge in performance across all tasks. Since AdapterHub adapters are only available for some of the languages in our evaluation, the results shown are averaged over only the languages for which they are available, indicated in the subfigure captions.

E Parameter Overlap between Languages

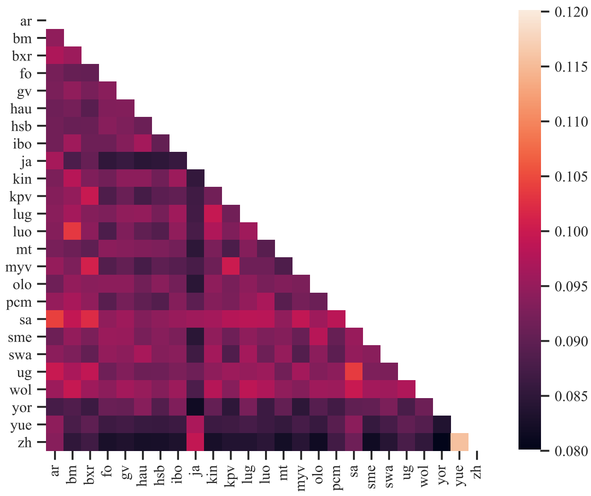


Figure 5: Percentage of parameters selected for the sparse fine-tuning of both languages in a pair.

In order to understand whether similar languages also share similar sub-networks, we plot the pairwise overlap (in percentage) between parameter subsets of language SFTs in Figure 5. Except for a single instance (Mandarin Chinese and Cantonese) where the high overlap reflects the fact that both languages are genealogically related, we find that the overlap is small for most language pairs. The explanation, we believe, is two-fold. Firstly, most of the languages in the multilingual datasets considered in our experiments belong to separate genera and families. Therefore, a lack of correlation in parameter subsets is expected. Secondly, for a pretrained model, there exist multiple parameter subsets ("winning tickets") with comparable performance (Prasanna et al., 2020). The Lottery Ticket algorithm selects randomly among these equally valid subsets. Hence, a lack of overlap does not necessarily imply the reliance on disjoint sub-networks.