Distilling Efficient Language-Specific Models for Cross-Lingual Transfer

Alan Ansell[1] **Edoardo Maria Ponti**[2,1] **Anna Korhonen**[1] **Ivan Vulić**[1]
[1]Language Technology Lab, University of Cambridge
[2]University of Edinburgh
aja63@cam.ac.uk

Abstract

Massively multilingual Transformers (MMTs), such as mBERT and XLM-R, are widely used for cross-lingual transfer learning. While these are pretrained to represent hundreds of languages, end users of NLP systems are often interested only in individual languages. For such purposes, the MMTs' language coverage makes them unnecessarily expensive to deploy in terms of model size, inference time, energy, and hardware cost. We thus propose to extract compressed, language-specific models from MMTs which retain the capacity of the original MMTs for cross-lingual transfer. This is achieved by distilling the MMT *bilingually*, i.e., using data from only the source and target language of interest. Specifically, we use a two-phase distillation approach, termed BISTILLATION: (i) the first phase distils a general bilingual model from the MMT, while (ii) the second, task-specific phase sparsely fine-tunes the bilingual 'student' model using a task-tuned variant of the original MMT as its 'teacher'. We evaluate this distillation technique in zero-shot cross-lingual transfer across a number of standard cross-lingual benchmarks. The key results indicate that the distilled models exhibit minimal degradation in target language performance relative to the base MMT despite being significantly smaller and faster. Furthermore, we find that they outperform multilingually distilled models such as DistilmBERT and MiniLMv2 while having a very modest training budget in comparison, even on a per-language basis. We also show that bilingual models distilled from MMTs greatly outperform bilingual models trained from scratch.

1 Introduction

Massively multilingual Transformers (MMTs), pretrained on unlabelled data from hundreds of languages, are a highly effective tool for cross-lingual transfer (Devlin et al., 2019; Conneau et al., 2020; Chung et al., 2020; He et al., 2021). However, they suffer from several limitations as a result of

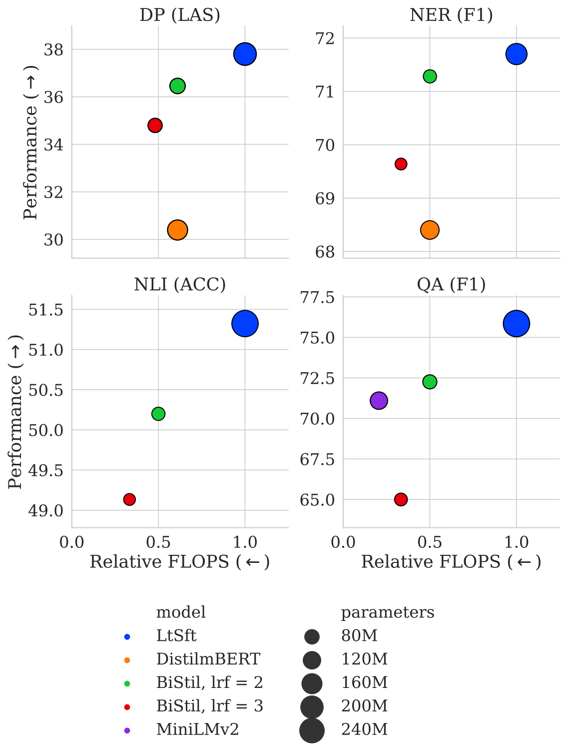

Figure 1: Tradeoff between parameter count, inference FLOPs and averaged performance for our BISTIL models for cross-lingual transfer and several baselines.

their ample language coverage. Firstly, aiming to represent many languages within their parameter budget and dealing with the training signals from different languages might result in negative interference. This is known as the "curse of multilinguality" (Conneau et al., 2020), which impairs the MMT's transfer capabilities (Pfeiffer et al., 2022). Secondly, in practice people are often interested in using or researching NLP systems in just a *single* language. This makes the MMTs *unnecessarily costly* in terms of storage, memory, and compute and thus hard to deploy. This especially impacts communities which speak low-resource languages, which are more likely to have limited access to computational resources (Alabi et al., 2022).

Findings of the Association for Computational Linguistics: ACL 2023, pages 8147–8165
July 9-14, 2023 ©2023 Association for Computational Linguistics

In this work, we address the question: *can we increase the time-efficiency and space-efficiency of MMTs while retaining their performance in cross-lingual transfer?* Knowledge distillation (Hinton et al., 2015) is a family of general methods to achieve the first goal by producing smaller, faster models (Sanh et al., 2019; Jiao et al., 2020, *inter alia*) and has also been applied to MMTs specifically. However, when the distilled MMT is required to cover the same number of languages as the original model, whose capacity is already thinly stretched over hundreds of languages, the "curse of multilinguality" asserts itself, resulting in a significant loss in performance (Sanh et al., 2019).

As a consequence, to achieve the best possible performance with reduced capacity, we depart from the practice of retaining all the languages from the original MMT in the distilled model. Instead, we argue, we should cover only *two* languages, namely the source language and the target language of interest. In fact, distilling just *one* language would fall short of the second goal stated above, namely facilitating cross-lingual transfer, as a monolingually distilled model would be unable to learn from a distinct source language during task-specific fine-tuning. Maintaining cross-lingual transfer capabilities, however, is crucial due to the paucity of labelled task data in many of the world's languages in most tasks (Ponti et al., 2019; Joshi et al., 2020).

In particular, we propose a method for *bilingual distillation* of MMTs, termed BISTILLATION, inspired by the two-phase recipe of Jiao et al. (2020). We start from a *"student"* model, initialized by discarding a subset of layers of the original *"teacher"* MMT, as well as the irrelevant part of its vocabulary. In the first, *"general"* phase of distillation, unlabelled data is used to align the the hidden representations and attention distributions of the student with those of the teacher. In the second, *task-specific* phase, the student is fine-tuned for the task of interest through guidance from a task-adapted variant of the teacher. Rather than fully fine-tuning the student during this second phase, we instead use the parameter-efficient Lottery-Ticket Sparse Fine-Tuning (LT-SFT) method of Ansell et al. (2022). Parameter-efficient task fine-tuning enables a system to support multiple tasks with the same distilled compact model, without unnecessarily creating full model copies per each task.

We evaluate our efficient *"bistilled"* models on a range of downstream tasks from several benchmarks for multilingual NLP, including dependency parsing from Universal Dependencies (UD; Zeman et al., 2020), named entity recognition from MasakhaNER (Adelani et al., 2021), natural language inference from AmericasNLI (Ebrahimi et al., 2022), and QA from XQuAD (Artetxe et al., 2020). We evaluate the model performance as well as its space efficiency (measured in terms of parameter count) and time efficiency (measured in terms of FLOPs and inference time). We compare it against highly relevant baselines: bilingual models pretrained from scratch and two existing multilingual distilled models, DistilmBERT (Sanh et al., 2019) and MiniLMv2 (Wang et al., 2021a).

We find that while our bilingually distilled models are twice or thrice smaller and faster than the original MMT, their performance is only slightly degraded, as illustrated in Figure 1. Our method outperforms the baselines by sizable margins, showing the advantages of (i) bilingual as opposed to multilingual distillation, and (ii) distilling models from MMTs rather than training them from scratch. We hope that our endeavour will benefit end-users of multilingual models, and potential users under-served by currently available technologies, by making NLP systems more accessible. The code and models are publicly available at `https://github.com/AlanAnsell/bistil`.

2 Background

2.1 Cross-Lingual Transfer with MMTs

Prominent examples of MMTs include mBERT (Devlin et al., 2019), XLM-R (Conneau et al., 2020) and mDeBERTa (He et al., 2021), among others.

Pires et al. (2019) and Wu and Dredze (2019) showed that mBERT is surprisingly effective at *zero-shot cross-lingual transfer*. Zero-shot cross-lingual transfer is a useful paradigm when there is little or no training data available for the task of interest in the target language, but there is training data available in some other *source* language. In the simplest form of zero-shot cross-lingual transfer, the model is trained on source language data and is then used without modification for inference on target language data. While this generally works quite well for high-resource languages, transfer performance degrades for low-resource languages, especially those under-represented or fully unseen by the MMT during its pretraining (Lauscher et al., 2020; Pfeiffer et al., 2020; Ansell et al., 2021; Adelani et al., 2021; Ebrahimi et al., 2022).

2.2 Modular Adaptation of MMTs

Because MMTs divide their capacity among many languages, they may often perform sub-optimally with respect to a single source or target language. Furthermore, we are sometimes interested in a target language not covered by the MMT. A naive solution to these problems is to prepare the MMT with continued pretraining on the target language before proceeding to task fine-tuning. While this can improve performance, Pfeiffer et al. (2020) show that a more effective approach is to perform this continued pretraining in a parameter-efficient manner, specifically with the use of *adapters* (Rebuffi et al., 2017; Houlsby et al., 2019). The resulting language-specific adapter is known as a *language adapter*. When the task fine-tuning is also learned in the form of an adapter (*task adapter*), Pfeiffer et al. demonstrate that zero-shot transfer can be achieved by composing arbitrary language and task adapter pairs.

Ansell et al. (2022) extend this idea to a new parameter-fine tuning method, *sparse fine-tuning* (SFT). An SFT of a model is where only a sparse subset of its pre-trained parameters are fine-tuned, i.e. an SFT of a pretrained model F with parameters $\boldsymbol{\theta}$ can be written as $F(\cdot\,;\,\boldsymbol{\theta}+\boldsymbol{\phi})$, where the *difference vector* $\boldsymbol{\phi}$ is sparse (Sung et al., 2021). Language and task SFTs with difference vectors $\boldsymbol{\phi}_L$ and $\boldsymbol{\phi}_T$ respectively are composed through addition, i.e. yielding $F(\cdot\,;\,\boldsymbol{\theta}+\boldsymbol{\phi}_L+\boldsymbol{\phi}_T)$. SFTs are learned through a procedure called "Lottery Ticket Sparse Fine-Tuning" (LT-SFT), based on the Lottery Ticket algorithm of Frankle and Carbin (2019). The $k\%$ of parameters which undergo the greatest absolute change during an initial full fine-tuning phase are selected as tunable parameters during the second "sparse" phase which yields the final SFT.

As SFT composition exhibited somewhat better zero-shot cross-lingual transfer performance across a range of tasks than adapter composition, and SFTs avoid the inference time slow-down incurred by adapters at inference time, we adopt this parameter-efficient approach throughout this work. However, we note that other modular and parameter-efficient architectures can also be tried in future work (Pfeiffer et al., 2023).

Multi-Source Training. Ansell et al. (2021) show that multi-source task adapter training, where a task adapter is trained using data from several source languages simultaneously, yields large gains in cross-lingual transfer performance as a result of the task adapter learning more language-agnostic representations. Ansell et al. (2022) find similarly large gains from multi-source training of task SFTs. An important aspect of cross-lingual transfer with SFTs is that the source language SFT is applied during task SFT training. This requires each batch during multi-source training to consist of examples from a single source language, for which the relevant language SFT is applied during the corresponding training step.

2.3 Distilling Pretrained Language Models

Knowledge distillation (Buciluă et al., 2006; Hinton et al., 2015) is a technique for compressing a pretrained large "teacher" model into a smaller "student" model by training the student to copy the behavior of the teacher. Whereas during standard pretraining, the model receives a single "hard" label per training example, during distillation the student benefits from the enriched signal provided by the full label distribution predicted by the teacher model. Sanh et al. (2019) use this technique to produce *DistilBERT*, a distilled version of BERT$_{\text{base}}$ (Devlin et al., 2019) with 6 instead of the original 12 layers, and *DistilmBERT*, a corresponding distilled version of multilingual BERT. There has been extensive subsequent work on distillation of pretrained language models, but with less focus on distilling MMTs in particular.

3 BiStillation: Methodology

Overview. We are interested in providing NLP capabilities with limited computational resources in a specific target language T which lacks training data in the tasks of interest. A common paradigm in previous work (Pfeiffer et al., 2020; Ansell et al., 2022) is to use cross-lingual transfer with an MMT in conjunction with parameter-efficient task and language adaptation to support multiple tasks without adding a large number of additional parameters per task, see §2.2. Our goal in this work is to replace the highly general MMT, plus optional language adaptation, with a target language-specific model which maintains the benefits of cross-lingual transfer.

An obvious first attempt would be to simply distil the MMT into a smaller model using only text in the target language. However, this monolingual distillation approach is insufficient, as during task fine-tuning, the monolingually distilled student model no longer "understands" the source language. Indeed, our preliminary experiments confirmed the intuition that this approach is inadequate. This problem can be overcome through *bilingual* distillation, where text from both the source and target language is used to train the student model.[1]

Therefore, our aim is to devise a method for deriving from an MMT M a smaller model $M'_{S,T,\tau}$ to perform a given task τ in the target language T given only training data in the source language S. Our approach is inspired by the two-stage distillation paradigm of Jiao et al. (2020). In the first, "general" phase, a bilingual student model $M'_{S,T}$ is distilled from M using the same unsupervised task (e.g., masked language modeling) that was used for M's pretraining. In the second, "task-specific" phase, $M'_{S,T,\tau}$ is produced by fine-tuning $M'_{S,T}$ using M_τ as its teacher, where M_τ is derived from M by fine-tuning it for task τ. The following sections explain the details of these phases.

3.1 Distillation Method

Let L_T be the number of Transformer layers in the teacher model, indexed from 1 to L_T. The number of student model layers L_S is required to evenly divide L_T. We define the downscaling stride as $s = \frac{L_T}{L_S}$.

Following Jiao et al. (2020), the loss functions of the two distillation phases make use of three components, (i) *attention-based*, (ii) *hidden state-based*, and (iii) *prediction-based*. Attention-based loss is defined as follows:

$$\mathcal{L}_{\text{attn}} = \frac{1}{L_S} \sum_{i=1}^{L_S} \texttt{MSE}(A_i^S, A_{i \cdot s}^T). \quad (1)$$

Here, A_i^S and $A_i^T \in \mathbb{R}^{l \times l}$ refer to the attention distribution[2] of Transformer layer i of the student and teacher model, respectively; l refers to the input sequence length; `MSE()` denotes mean squared error loss.

Hidden state-base loss is defined as follows:

$$\mathcal{L}_{\text{hidden}} = \frac{1}{L_S + 1} \sum_{i=0}^{L_S} \texttt{MSE}(H_i^S, H_{i \cdot s}^T), \quad (2)$$

where H_i^S and $H_i^T \in \mathbb{R}^{l \times d}$ refer to the hidden representations output by Transformer layer i of the student and teacher model, respectively, or the output of the embedding layer when $i = 0$. Note that we assume that the student and teacher share the same hidden dimensionality d.

Finally, the prediction-based loss is defined as

$$\mathcal{L}_{\text{pred}} = \texttt{CE}(\boldsymbol{z}^S, \boldsymbol{z}^T), \quad (3)$$

where $\boldsymbol{z}^S$ and $\boldsymbol{z}^T$ are the label distributions predicted by the student and teacher model, respectively, and `CE` denotes cross-entropy loss.

The intuition behind using attention-based and hidden state-based loss for our purposes is as follows. We (i) require good monolingual performance in the source and target language, but we also (ii) must preserve the existing alignment between these languages in the MMT which would consequently facilitate transfer between them. The intuition is that encouraging the student's intermediate representations to match those of the teacher will help to preserve this alignment.

We next describe how these loss components are employed in each phase of BISTILLATION.

3.2 Stage 1: General Bilingual Distillation

Initialization. We initialize all parameters of the student model by copying those of the teacher model, but retaining only the Transformer layers whose indices are multiples of s.

Vocabulary Reduction. Our distilled models can dispose of the many irrelevant tokens in the base MMT's vocabulary, i.e. those which are not frequently used in either the source or target language of interest, an idea previously proposed by Abdaoui et al. (2020). During initialization, the vocabulary of the student model is selected by retaining only the tokens of the teacher's vocabulary whose unigram probability in either the source or target language corpus is $\geq 10^{-6}$.

Teacher Language Adaptation. As we wish to be able to produce distilled models for languages not covered in the base MMT, and to obtain the best possible performance for languages which are covered, we employ language adaptation of the teacher MMT with language-specific SFTs (Ansell et al., 2022) applied on top of the original MMT during distillation.[3] Since it draws examples from two languages, each with its own language SFT, bilingual

[1] This is similar to the idea of bilingual language adapters proposed by Parović et al. (2022), which obtain superior transfer performance by adapting the MMT to both source and target language simultaneously, removing the need to use different and possibly incompatible language adapters during training and inference.

[2] Here, for ease of implementation within the Huggingface Transformers library (Wolf et al., 2020), we differ from Jiao et al. (2020), who use raw attention scores.

[3] Put simply, additionally applying language-specific SFTs 'skews' the MMT towards those particular languages.

distillation becomes a special case of multi-source training as described in §2.2. At each training step, either the source or target language is selected at random with equal probability; the batch is composed of sequences drawn from the training corpus of the chosen language, and a pretrained SFT for that language is applied to the teacher MMT.

Objective. The overall loss function for this phase is given by the sum of the attention-based and hidden state-based loss. Omitting the prediction-based loss here has the advantage of avoiding the need to evaluate the distribution of tokens predicted by the MLM head, which is costly because of the considerable size of MMTs' embedding matrices.

3.3 Stage 2: Task-Specific Distillation

After a general bilingual model has been distilled from the teacher MMT in Stage 1, it can be fine-tuned for a specific task. We first obtain the teacher for task-specific distillation by applying task-specific LT-SFT to fine-tune the base MMT (i.e., the teacher in the general distillation phase) for the task in question. This teacher's outputs and representations are then used to fine-tune the bilingual student model, again using task LT-SFT at the student's end. The use of parameter-efficient task adaptation here avoids adding a large number of parameters to the system for each task. The objective during this task-specific fine-tuning consists of the sum of all three losses from §3.1: $\mathcal{L}_{\text{attn}}$, $\mathcal{L}_{\text{hidden}}$, and $\mathcal{L}_{\text{pred}}$.

4 Experimental Setup

We largely adopt the evaluation framework of Ansell et al. (2022) for direct comparability with their LT-SFT method, which they apply to undistilled MMTs, and which we apply for task-specific fine-tuning of bilingually distilled MMTs. Specifically, we evaluate zero-shot cross-lingual transfer performance on four representative tasks: dependency parsing, named entity recognition, natural language inference, and QA. While the prior work focused only on low-resource languages, our method is also highly relevant to high-resource languages: the XQuAD QA task (Artetxe et al., 2020) provides additional insight into high-resource target language performance. Table 1 summarizes the experimental setup, including the datasets and languages considered in our experiments. In total, we cover a set of 44 typologically and geographically diverse languages, which makes them representative of cross-lingual variation (Ponti et al., 2020).

We experiment with three different MMTs as shown in Table 1: mBERT (Devlin et al., 2019), XLM-R$_{\text{base}}$ (Conneau et al., 2020), and mDeBERTa$_{\text{base}}$ (He et al., 2021).

4.1 Baselines and Model Variants

We refer to our main method as BiStil. We compare it with several relevant approaches. First, the LtSft method (Ansell et al., 2022), a state-of-the-art cross-lingual transfer approach, uses LT-SFT with language adaptation on the base MMT. LtSft can be seen as an upper bound for BiStil, allowing us to measure how much the performance suffers as a result of replacing the MMT with its bilingually distilled variant.

For each task except NLI,[4] we also compare against a multilingually distilled MMT, i.e. with all pretraining languages used for distillation as well. For DP and NER, where mBERT is the base MMT, the distilled MMT is DistilmBERT (Sanh et al., 2019), which is similarly based on mBERT. For QA, where BiStil uses mDeBERTa as the base MMT, no directly comparable multilingually distilled MMT is available, so we opt for a loose comparison with MiniLMv2 (Wang et al., 2021a), distilled from XLM-R$_{\text{large}}$, which has achieved strong results on cross-lingual transfer in high-resource languages. We perform task-specific fine-tuning with LT-SFT on DistilmBERT and MiniLMv2 in the same way as for the the undistilled MMTs in the LtSft setting. For DP and NER we also perform language adaptation of DistilmBERT.[5]

We also consider Scratch, a setting where we train bilingual models from scratch instead of distilling them from a pretrained MMT. We then apply the same LT-SFT task fine-tuning method as for the other baselines. This comparison allows us to evaluate the benefit of distilling efficient bilingual models from the MMT rather than pretraining the same-sized bilingual models from scratch.

We refer to our main method, with the task-specific distillation stage as described in §3.3, as

[4] There does not seem to be a multilingually distilled MMT that would provide a suitable comparison on the AmericasNLI task, as MiniLMv2 and other models distilled without an MLM head do not support adaptation to unseen languages through continued pretraining. DistilmBERT on the other hand is based on a generally weaker MMT than XLM-R, which is used as BiStil's base MMT for NLI.

[5] MiniLMv2 does not support language adaptation (see Footnote 4), but this is not as important for the high-resource languages used in XQuAD (Ansell et al., 2022).

Task	Target Dataset	Source Dataset	MMT	Target Languages
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., 2022)	MultiNLI (Williams et al., 2018)	XLM-R	Aymara, Asháninka, Bribri, Guarani, Náhuatl, Otomí, Quechua, Rarámuri, Shipibo-Konibo, Wixarika
Question Answering (QA)	XQuAD (Artetxe et al., 2020)	SQuAD v1.1 (Rajpurkar et al., 2016)	mDeBERTa	Arabic†, Chinese†, German†, Greek†, Hindi†, Romanian†, Russian†, Spanish†, Thai†, Turkish†, Vietnamese†

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

MMT	Distillation	LRF	DRF	#L	D	#V	#P
mBERT	none	-	-	12	768	120K	178M
	D'mBERT	2	-	6	768	120K	135M
	BiStil*	2	-	6	768	31K	67M
		3	-	4	768	31K	53M
XLM-R$_{base}$	none	-	-	12	768	250K	278M
	BiStil*	2	-	6	768	28K	65M
		3	-	4	768	28K	51M
XLM-R$_{large}$	none	-	-	24	1024	250K	560M
	MiniLMv2	2	2.67	12	384	250K	118M
mDeBERTa	none	-	-	12	768	250K	278M
	BiStil*	2	-	6	768	41K	75M
		3	-	4	768	41K	60M

Table 2: Dimensions of models *before* and *after* distillation. LRF = Layer Reduction Factor; DRF = hidden Dimension Reduction Factor; #L = number of Transformer Layers; D = hidden Dimension; #V = Vocabulary size; #P = total number of model Parameters; D'mBERT = DistilmBERT. * - because of its vocabulary reduction procedure, BiStil can produce models of slightly different sizes for different languages; the vocabulary sizes and numbers of parameters shown are averages over all BiStil models trained.

BiStil-TF (TF = *teacher forcing*). We also carry out an ablation focused on the second phase of BiStillation: here, we consider performing task-specific fine-tuning without the assistance of a teacher, i.e. in the same manner as LtSft. We refer to this variant as BiStil-ST (ST = *self-taught*).

Table 2 provides details of the model sizes, before and after distillation using the above methods, demonstrating the benefits of BiStillation with respect to model compactness.

4.2 Distillation/Adaptation Training Setup

We always perform language adaptation of the teacher model during both phases of BiStillation and during LtSft except for mDeBERTa and MiniLMv2[6]. For language adaptation of MMTs we use the pretrained language SFTs of Ansell et al. (2022), and we train our own for DistilmBERT. Similarly, for the LtSft baseline, and for task adaptation of the teacher in the BiStil-TF configuration, we use their pretrained single-source task SFTs or train our own when necessary. When training/distilling our own models or SFTs, we generally choose hyperparameters which match those used to train their SFTs in the original work. See Appendix A for full training details and hyperparameters of all models in our comparison, and Appendix B for details of the training corpora.

We experiment with two layer reduction factors (LRF) for BiStillation, 2 (a reduction from 12 to 6 layers) and 3 (12 to 4 layers). Whereas the BiStil setting initializes the model from the teacher (see §3.2), the Scratch setting initializes it randomly.

5 Results and Discussion

The results in terms of task performance are summarized in Tables 3-6. As expected, LtSft on the undistilled MMTs performs best across all tasks. However, BiStil-TF with reduction factor 2 is not much worse, with a degradation in performance not exceeding 1.3 points relative to LtSft on DP, NER and NLI. The larger gap of 3.4 EM points on QA is likely a result of the fact that the base MMT is much more thoroughly pretrained on the high-resource languages found in XQuAD than on the lower-resource languages found in the datasets for the other tasks. It is therefore harder for BiDis-

[6]See Footnote 4 for MiniLMv2; mDeBERTa could in theory support language adaptation but its pretraining code was not made publicly available in time to be used in this work.

	ar	bm	bxr	fo	gv	hsb	ja	kpv	mt	myv	olo	sa	sme	ug	yue	zh	avg	avgΔ
LTSFT	53.6	16.5	25.9	55.5	42.4	60.5	19.7	27.2	55.4	45.3	47.8	25.2	42.1	16.7	34.0	37.0	37.8	-
DISTILMBERT	47.7	9.9	19.5	49.1	31.7	53.2	16.2	20.0	43.0	34.9	37.6	17.7	31.4	11.4	28.9	33.9	30.4	-7.4
SCRATCH, LRF = 2	16.9	4.9	6.7	27.8	9.1	15.2	6.7	5.6	16.1	12.7	11.1	3.5	9.3	3.9	11.5	14.6	11.0	-26.8
BISTIL-ST, LRF = 2	50.9	15.8	24.1	53.7	38.3	57.1	18.7	**23.9**	52.2	**43.7**	**46.5**	25.2	39.8	13.3	31.8	34.8	35.6	-2.2
BISTIL-ST, LRF = 3	48.2	16.1	23.4	52.1	35.0	55.1	18.1	22.2	49.9	40.3	41.3	22.2	37.6	13.3	30.7	33.4	33.7	-4.1
BISTIL-TF, LRF = 2	**53.2**	**16.4**	**24.6**	**54.8**	**39.1**	**59.0**	**19.0**	23.8	**54.1**	43.5	46.0	**26.9**	**40.7**	13.1	**32.7**	**36.4**	**36.5**	**-1.3**
BISTIL-TF, LRF = 3	49.7	**16.4**	24.4	52.7	36.8	57.1	18.2	21.0	52.2	41.0	43.3	25.1	38.1	**14.5**	31.3	34.9	34.8	-3.0

Table 3: DP; LAS scores. The results with the smallest gap to the upper-bound LTSFT model are in **bold**.

	hau	ibo	kin	lug	luo	pcm	swa	wol	yor	avg	avgΔ
LTSFT	83.5	76.7	67.4	67.9	54.7	74.6	79.4	66.3	74.8	71.7	-
DISTILMBERT	81.1	73.2	65.3	63.4	50.0	69.2	77.7	64.4	71.2	68.4	-3.3
BISTIL-ST, LRF = 2	**81.3**	74.1	65.9	66.7	53.5	72.1	77.1	64.6	72.8	69.8	-1.9
BISTIL-ST, LRF = 3	80.3	74.0	63.1	64.6	54.7	69.6	76.9	68.0	70.5	69.1	-2.6
BISTIL-TF, LRF = 2	81.0	**74.8**	**67.5**	**67.3**	55.0	**72.9**	**78.4**	**69.0**	**75.7**	**71.3**	**-0.4**
BISTIL-TF, LRF = 3	79.6	**74.8**	64.6	64.5	**56.7**	70.6	77.2	66.1	72.8	69.6	-2.1

Table 4: NER; F1 scores.

	aym	bzd	cni	gn	hch	nah	oto	quy	shp	tar	avg	avgΔ
LTSFT	58.1	44.4	47.9	63.5	42.8	52.4	48.5	62.0	50.3	43.3	51.3	-
BISTIL-TF, LRF = 2	**58.9**	**45.7**	46.4	**62.9**	**44.3**	50.8	**44.0**	58.7	**47.2**	**43.1**	**50.2**	**-1.1**
BISTIL-TF, LRF = 3	57.7	43.6	**48.1**	60.9	41.3	**51.4**	42.6	**59.9**	45.5	40.3	49.1	-2.2

Table 5: NLI accuracy (%)

	ar	de	el	es	hi	ro	ru	th	tr	vi	zh	avg	avgΔ
LTSFT	56.5	64.7	61.2	62.4	57.8	69.0	61.8	56.0	56.4	57.1	60.8	60.3	-
MINILMV2	50.4	59.4	54.4	57.9	52.9	64.5	57.6	50.3	51.3	**53.8**	55.0	55.2	-5.1
BISTIL-TF, LRF = 2	**53.5**	**62.2**	**55.4**	**59.8**	**54.5**	**66.2**	**58.3**	**54.4**	**53.1**	53.4	**55.7**	**57.0**	**-3.4**
BISTIL-TF, LRF = 3	44.3	55.0	44.1	55.2	46.1	59.5	51.3	42.4	48.3	44.6	50.9	49.3	-11.1

Table 6: XQuAD; Exact Match scores.

TIL to achieve the base MMT's depth of knowledge of the target language during its relatively short distillation training time. BISTIL-TF, LRF = 2 nevertheless outperforms MiniLMv2 on QA by 1.7 EM points, despite MiniLMv2 receiving 320 times more training than each BIDISTIL model, or roughly 6 times more per language[7].

Furthermore, BISTIL-TF, LRF = 2 significantly outperforms DISTILMBERT, with a 6.1 LAS gap on DP and 2.9 F1 gap on NER. BISTIL, LRF = 2 produces models roughly half the size of DISTILMBERT and that, once again, are trained for vastly less time[8].

Training bilingual models from SCRATCH performs poorly, lagging behind the other methods by more than 20 points on DP.[9] One crucial weakness of SCRATCH, besides its reduced monolingual performance, is a lack of alignment between its representations of the source and target languages, severely impairing cross-lingual transfer. This highlights the advantage of distilling a bilingual model from an MMT within which cross-lingual alignment is already present.

Interestingly, when we evaluate the SCRATCH models on their *English* DP performance, we obtain an average UAS/LAS score of 81.8/77.1, which is much more competitive in relative terms with the BISTIL-TF, LRF = 2 English DP score of 91.0/88.2 than the corresponding comparison in average target language DP scores of 29.9/11.0 to 55.5/36.5. This suggests that an even larger factor in SCRATCH's weakness than its poor monolingual performance is a lack of alignment between its representations of the source and target languages,

[7]MiniLMv2 is trained for 1M steps with a batch size of 256 and max sequence length of 512; BIDISTIL for 200K steps with a batch size of 8 and max sequence length of 256.

[8]Sanh et al. (2019) note that their monolingual DistilBERT model was trained on 8 16GB V100 GPUs for approximately 90 hours. Our BISTIL models are trained on a single 10GB RTX 3080 GPU for approximately 9 hours.

[9]As this method is clearly inferior, we opted to reduce computational expense by not repeating it for other tasks.

	cpu ↑	gpu ↑	flops ↓
DISTILMBERT	1.41x	1.03x	0.61x
BISTIL, LRF = 2	1.44x	1.25x	0.61x
BISTIL, LRF = 3	1.71x	1.36x	0.48x

(a) DP efficiency

	cpu ↑	gpu ↑	flops ↓
DISTILMBERT	1.93x	1.94x	0.50x
BISTIL, LRF = 2	1.97x	1.98x	0.50x
BISTIL, LRF = 3	2.97x	2.78x	0.33x

(b) NER efficiency

	cpu ↑	gpu ↑	flops ↓
BISTIL, LRF = 2	2.02x	1.97x	0.50x
BISTIL, LRF = 3	2.89x	2.85x	0.33x

(c) NLI efficiency

	cpu ↑	gpu ↑	flops ↓
MINILMV2	4.25x	3.44x	0.21x
BISTIL, LRF = 2	1.99x	1.85x	0.50x
BISTIL, LRF = 3	2.85x	2.42x	0.33x

(d) QA efficiency

Table 7: Relative inference speed and FLOP cost. Values are given relative to LTSFT without distillation, i.e. a speed reading of "2.00x" means the distilled model can on average process twice as many inference examples per second as the undistilled MMT. Likewise a FLOPs reading of "0.50x" would mean that the distilled model on average requires half as many FLOPs to process an inference example as the undistilled MMT does.

severely impairing cross-lingual transfer. This highlights the advantage of distilling a bilingual model from an MMT within which cross-lingual alignment is already present.

As expected, the performance of BISTIL is somewhat weaker with a larger layer reduction factor of 3, though this is heavily task-dependent. With an LRF of 3, BISTIL-TF still comfortably outperforms DISTILMBERT on DP and NER, and does not fall much behind LRF = 2 for NLI. However, we observe a considerable degradation in performance for LRF = 3 for QA; this may indicate that a 4-layer Transformer struggles to adapt to this particular task, or that for this architecture the modest training time is not sufficient to approach the base MMT's understanding of the source and target languages.

Table 7 presents an analysis of the inference time efficiency. We measure the inference speed both on CPU with batch size 1 and GPU with the same batch size as during task-specific training. We also calculate the number of floating-point operations (FLOPs) per example using fvcore, measured during an inference pass over the test set of the first language in each task.

For NER, NLI and QA, the efficiency results conform quite closely to the intuitive expectation that a model's inference time should scale linearly with its number of layers; that is, BIDISTIL with LRF = 2 is generally around twice as fast as the base MMT. For DP, we observe a seemingly sub-linear scaling which is caused by the very large biaffine parsing head, consisting of ∼23M parameters. The significant cost of applying the model head contributes equally to all models regardless of their degree of distillation. Despite having a moderate LRF of 2, MINILMV2 exhibits impressive speed as a result of the fact that it additionally has a smaller hidden dimension than its teacher (see Table 2), a technique which we do not consider for BIDISTIL, but may be a promising avenue for future work.

We argue that BIDISTIL accomplishes its aim by achieving two- to three-fold reductions in inference time and model size without sacrificing much in the way of raw performance. Its superior performance relative to multilingually distilled models despite its comparatively very modest training budget supports the assertion that specializing multilingual models for a specific transfer pair during distillation helps to avoid performance degradation resulting from the curse of multilinguality.

6 Related Work

One strand of prior work focuses on parameter-efficient adaptation of pretrained MMTs, i.e. adaptation by adding/modifying a small subset of parameters. Adapters (Rebuffi et al., 2017; Houlsby et al., 2019) have been used extensively for this purpose (Üstün et al., 2020), with the MAD-X framework of Pfeiffer et al. (2020) becoming a starting point for several further developments (Vidoni et al., 2020; Wang et al., 2021b; Parović et al., 2022), where a notable theme is adapting MMTs to unseen languages (Ansell et al., 2021; Pfeiffer et al., 2021). Ansell et al. (2022) propose composable sparse fine-tunings as an alternative to adapters.

Pfeiffer et al. (2022) create a modular MMT from scratch, where some parameters are shared among all languages and others are language-specific. This allows the model to dedicate considerable capacity to every language without each language-specific model becoming overly large; thus it is quite similar in its aims to this work.

A variety of approaches have been proposed for general distillation of pretrained language models. The simplest form uses only soft target probabilities predicted by the teacher model as the training signal for the student (Sanh et al., 2019). Other approaches try to align the hidden states and self-attention distributions of the student and teacher (Sun et al., 2020; Jiao et al., 2020) and/or finer-grained aspects of the self-attention mechanism (Wang et al., 2020, 2021a). Mukherjee et al. (2021) initialize the student's embedding matrix with a factorization of the teacher's for better performance when their hidden dimensions differ. Of these, Sanh et al. (2019); Wang et al. (2020, 2021a); Mukherjee et al. (2021) apply their methods to produce distilled versions of MMTs.

Parović et al. (2022) adapt pretrained MMTs to specific transfer pairs with adapters; this approach is similar to ours in spirit, but it is aimed towards improving performance rather than efficiency. Minixhofer et al. (2022) learn to transfer full monolingual models across languages. The only work prior we are aware of which creates purely bilingual models for cross-lingual transfer is that of Tran (2020). This approach starts with a monolingual pretrained source language model, initializes target language embeddings via an alignment procedure, and then continues training the model with the added target embeddings on both languages.

7 Conclusions

While MMTs are an effective tool for cross-lingual transfer, their broad language coverage makes them unnecessarily costly to deploy in the frequently-encountered situation where capability is required in only a single, often low-resource, language. We have proposed BiStillation, a method of training more efficient models suited to this scenario which works by distilling an MMT using only the source-target language pair of interest. We show that this approach produces models that offer an excellent trade off between target language performance, efficiency, and model compactness. The 'bistilled' models exhibit only a slight decrease in performance relative to their base MMTs whilst achieving considerable reduction in both model size and inference time. Their results also compare favorably to those of multilingually distilled MMTs despite receiving substantially less training even on a per-language basis.

Limitations

While the results of our experiments seem sufficient to validate the concept and our general approach to bilingual distillation, we have not carried out a detailed systematic analysis of alternative implementations of the various aspects of our methods, such as different student model initializations, distillation objectives and hyperparameter settings. Furthermore, our BiStil models are likely undertrained due to limited computational resources. Consequently, we do not claim our specific implementation of bilingual distillation to be optimal or even close to optimal. Areas that warrant further investigation toward realizing the full potential of this approach include the use of hidden dimension reduction, which yielded impressive speed gains for MiniLMv2 in our experiments, and other innovations in distillation such as progressive knowledge transfer (Mukherjee et al., 2021).

With the exception of improved efficiency, our BiStil models inherit the limitations of the MMTs from which they are distilled; notably, there is a discrepancy between the performance on high- and low-resource languages resulting from the distribution of data used during MMT pretraining.

In this work, we have only considered English as the source language; some target languages may benefit from other transfer sources. Future work may also consider the use of multi-source transfer, which would entail distilling with more than two languages. Here the challenge would be optimizing the balance of model capacity allocated to source languages versus the target language.

Acknowledgements

Alan wishes to thank David and Claudia Harding for their generous support via the Harding Distinguished Postgraduate Scholarship Programme. Ivan Vulić is supported by a personal Royal Society University Research Fellowship *'Inclusive and Sustainable Language Technology for a Truly Multilingual World'* (no 221137; 2022–).

References

Amine Abdaoui, Camille Pradel, and Grégoire Sigel. 2020. Load what you need: Smaller versions of mutililingual BERT. In *Proceedings of SustaiNLP: Workshop on Simple and Efficient Natural Language Processing*, pages 119–123, Online. Association for Computational Linguistics.

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 H. Muhammad, Chris Chinenye Emezue, Joyce Nakatumba-Nabende, Perez Ogayo, Aremu Anuoluwapo, Catherine Gitau, Derguene Mbaye, Jesujoba Alabi, Seid Muhie Yimam, Tajuddeen Rabiu 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. *Transactions of the Association for Computational Linguistics*, 9:1116–1131.

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.

Jesujoba O. Alabi, David Ifeoluwa Adelani, Marius Mosbach, and Dietrich Klakow. 2022. Adapting pre-trained language models to African languages via multilingual adaptive fine-tuning. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4336–4349, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

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

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.

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

Cristian Buciluă, Rich Caruana, and Alexandru Niculescu-Mizil. 2006. Model compression. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, page 535–541, New York, NY, USA. Association for Computing Machinery.

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.

Hyung Won Chung, Thibault Fevry, Henry Tsai, Melvin Johnson, and Sebastian Ruder. 2020. Rethinking embedding coupling in pre-trained language models. In *International Conference on Learning Representations*.

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.

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 Meza Ruiz, Gustavo Giménez-Lugo, Elisabeth Mager, Graham Neubig, Alexis Palmer, Rolando Coto-Solano, Thang Vu, and Katharina Kann. 2022. AmericasNLI: Evaluating zero-shot natural language understanding of pretrained multilingual models in truly low-resource languages. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6279–6299, Dublin, Ireland. Association for Computational Linguistics.

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*.

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.

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).

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network.

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.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2020. TinyBERT: Distilling BERT for natural language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4163–4174, Online. Association for Computational Linguistics.

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

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

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

Benjamin Minixhofer, Fabian Paischer, and Navid Rekabsaz. 2022. WECHSEL: Effective initialization of subword embeddings for cross-lingual transfer of monolingual language models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3992–4006, Seattle, United States. Association for Computational Linguistics.

Subhabrata Mukherjee, Ahmed Hassan Awadallah, and Jianfeng Gao. 2021. Xtremedistiltransformers: Task transfer for task-agnostic distillation.

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.

Marinela Parović, Goran Glavaš, Ivan Vulić, and Anna Korhonen. 2022. BAD-X: Bilingual adapters improve zero-shot cross-lingual transfer. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational*

Linguistics: Human Language Technologies, pages 1791–1799, Seattle, United States. Association for Computational Linguistics.

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

Jonas Pfeiffer, Sebastian Ruder, Ivan Vulić, and Edoardo Maria Ponti. 2023. Modular deep learning. *CoRR*, abs/2302.11529.

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

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

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy. Association for Computational Linguistics.

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

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.

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.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter.

Zhiqing Sun, Hongkun Yu, Xiaodan Song, Renjie Liu, Yiming Yang, and Denny Zhou. 2020. MobileBERT: a compact task-agnostic BERT for resource-limited devices. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2158–2170, Online. Association for Computational Linguistics.

Yi-Lin Sung, Varun Nair, and Colin Raffel. 2021. Training neural networks with fixed sparse masks. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 24193–24205.

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.

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

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

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

Wenhui Wang, Hangbo Bao, Shaohan Huang, Li Dong, and Furu Wei. 2021a. MiniLMv2: Multi-head self-attention relation distillation for compressing pretrained transformers. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2140–2151, Online. Association for Computational Linguistics.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Advances in Neural Information Processing Systems*, volume 33, pages 5776–5788. Curran Associates, Inc.

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

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.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Shijie Wu and Mark Dredze. 2019. Beto, bentz, becas: The surprising cross-lingual effectiveness of BERT. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 833–844, Hong Kong, China. Association for Computational Linguistics.

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, Hórunn 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, Adédayọ̀ 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, Steinþór 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.

A Training Details and Hyperparameters

As we evaluate over many languages and tasks, we carry out a single run per (task, language, configuration) triple.

A.1 Language Distillation/Adaptation

The following are constant across all language distillation/SFT training: we use a batch size of 8 and a maximum sequence length of 256; 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 linear decay without any warm-up.

During LT-SFT training of DistilmBERT's language SFTs, the dense and sparse fine-tuning phases each last the lesser of 100,000 steps or 200 epochs, but at least 30,000 steps if 200 epochs is less. The initial learning rate is $5 \cdot 10^{-5}$. The SFT density is set to 4%.[10]

When distilling bilingual models or learning them from scratch, training lasts 200,000 steps (to equal the total length of the two phases of LT-SFT training). The initial learning rate is 10^{-4}. The model architecture and hyperparameters are identical to the teacher MMT's other than a reduction in the number of layers and the use of vocabulary reduction as described in §3.2.

A.2 Task Distillation/Adaptation

For DP and NER, we train task SFTs for 3 epochs in the dense phase of LT-SFT and 10 epochs in the sparse phase, evaluating the model checkpoint on the validation set at the end of each epoch, and taking the best checkpoint at the end of training. The selection metric is labeled attachment score for DP and F1-score for NER. The initial learning rate is $5 \cdot 10^{-5}$ with linear decay. For 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 train for 5 epochs with batch size 32, with checkpoint evaluation on the validation set every 625 steps,

[10]This is similar but not identical to the density used by Ansell et al. (2022), who use a very specific number of trainable parameters for comparability to their baseline; we prefer to use a round number.

and an initial learning rate of $2 \cdot 10^{-5}$. We apply a two-layer multi-class classification head atop the model output corresponding to the `[CLS]` token. For QA, we train for 5 epochs with a batch size of 12, with checkpoint evaluation every 2000 steps and an initial learning rate of $3 \cdot 10^{-5}$. The single-layer model head independently predicts the start and end positions of the answer span, and at inference time the span whose endpoints have the largest sum of logits is selected.

We set the density of our task SFTs to 8%, which Ansell et al. (2022) found to offer the best task performance in all their experiments.

B Languages

Task	Language	ISO Code	Family	UD Treebank	Corpus source(s)
Source	English	en	Indo-European, Germanic	EWT	Wikipedia
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	Arabic	ar	Afro-Asiatic, Semitic	N/A	Wikipedia
	Chinese	zh	Sino-Tibetan		
	German	de	Indo-European, Germanic		
	Greek	el	Indo-European, Greek		
	Hindi	hi	Indo-European, Indic		
	Romanian	ro	Indo-European, Romance		
	Russian	ru	Indo-European, Slavic		
	Thai	th	Tai-Kadai, Kam-Tai		
	Turkish	tr	Turkic, Southwestern		
	Vietnamese	vi	Austro-Asiatic, Viet-Muong		

Table 8: 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.

C Additional Results

	ar	bm	bxr	fo	gv	hsb	ja	kpv	mt	myv	olo	sa	sme	ug	yue	zh	avg	avgΔ
LtSft	70.8	43.1	49.2	68.2	60.0	73.7	36.9	50.5	74.6	65.9	66.4	49.5	58.0	36.4	51.1	59.8	57.1	-
DistilmBert	65.7	34.4	42.3	63.0	52.8	67.6	32.1	42.2	65.4	58.6	59.6	44.1	51.2	29.2	47.0	56.1	50.7	-6.4
Scratch, lrf = 2	38.5	26.6	24.8	44.9	35.4	33.5	18.6	23.4	42.9	31.5	30.2	23.0	26.1	12.3	30.8	35.6	29.9	-27.2
BiStil-ST, lrf = 2	68.0	41.6	45.7	66.3	56.6	70.9	34.1	**48.2**	71.0	**64.5**	**64.3**	48.9	**57.6**	**34.5**	49.4	56.7	54.9	-2.2
BiStil-ST, lrf = 3	65.5	42.5	45.9	64.1	52.7	68.1	33.2	46.5	68.0	62.0	61.5	46.9	55.1	32.4	48.6	55.3	53.0	-4.1
BiStil-TF, lrf = 2	**70.3**	43.4	46.8	**67.1**	**57.7**	**72.4**	**34.5**	47.6	**72.7**	64.2	62.6	**50.5**	57.4	32.3	**49.8**	**58.6**	**55.5**	**-1.6**
BiStil-TF, lrf = 3	67.0	**43.9**	**47.6**	65.1	54.2	70.0	33.4	44.2	69.7	62.3	61.8	49.2	55.1	33.3	48.9	56.5	53.9	-3.3

Table 9: DP UAS score

	ar	de	el	es	hi	ro	ru	th	tr	vi	zh	avg	avgΔ
LTSFT	73.0	80.5	78.6	80.6	74.3	82.4	77.8	69.7	72.2	76.5	68.9	75.9	-
MINILMV2	66.4	75.5	72.4	76.6	69.6	78.3	74.0	63.8	67.6	**73.3**	**64.6**	71.1	-4.8
BISTIL-TF, LRF = 2	**69.4**	**77.4**	**73.8**	**77.6**	**69.7**	**79.1**	**75.0**	**66.7**	**68.8**	72.8	64.5	**72.3**	**-3.6**
BISTIL-TF, LRF = 3	62.4	70.7	63.3	74.7	61.4	73.4	68.7	54.3	62.9	63.0	60.4	65.0	-10.9

Table 10: XQuAD F1 score

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
8

☒ A2. Did you discuss any potential risks of your work?
We do not consider there to be any significant apparent risks arising from the work.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
1

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

B ☑ Did you use or create scientific artifacts?

4,5

☑ B1. Did you cite the creators of artifacts you used?
Throughout.

☒ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
The license terms are easily accessible using the links provided and our usage was clearly in keeping with the terms.

☒ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
All use of existing artifacts was clearly in keeping with their intended purpose.

☒ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
We did not use sensitive data in our experiments.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
4

☒ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Did not include # of examples for space reasons, though this information is easily accessible. Would include in camera-ready.

C ☑ Did you run computational experiments?

4,5

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
5. Total computational budget is not stated as it is difficult to calculate exactly, but time for core experimental method is provided.

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

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
4, A1.

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
5, A1.

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
3,4

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

Left blank.

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

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

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

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

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