Orthogonal Language and Task Adapters in Zero-Shot Cross-Lingual Transfer

Anonymous ACL submission

Abstract

Adapter modules have recently been used for efficient fine-tuning and language specialization of massively multilingual Transformers (MMTs), improving downstream zero-shot cross-lingual transfer. In this work, we propose *orthogonal* language and task adapters (dubbed *orthoadapters*) for cross-lingual transfer. They are trained to encode language- and task-specific information that is complementary (i.e., orthogonal) to the knowledge already stored in the pretrained MMT parameters. Our zero-shot transfer experiments, involving three tasks and 10 diverse languages, **1)** point to the usefulness of orthoadapters in cross-lingual transfer, especially for the most complex NLI task, but also **2)** indicate that the optimal (ortho)adapter configuration highly depends on the task and the target language at hand. We hope that our work will motivate a wider investigation of usefulness of orthogonality constraints in language- and task-specific fine-tuning of pretrained transformers.

1 Introduction

Massively multilingual transformers (MMTs), pretrained on large multilingual corpora via language modeling (LM) objectives (Devlin et al., 2019; Conneau et al., 2020) have overthrown (static) cross-lingual word embeddings (Ruder et al., 2019; Glavaš et al., 2019) as the state-of-the-art paradigm for zero-shot (ZS) cross-lingual transfer. However, MMTs are constrained by the so-called *curse of multilinguality*: the quality of language-specific representations starts decreasing when the number of training languages exceeds the MMT's parameter capacity (Arivazhagan et al., 2019; Conneau et al., 2020). Languages with smallest training corpora are most affected: the largest transfer performance drops occur with those target languages (Lauscher et al., 2020; Wu and Dredze, 2020).

Additional LM training of a full pretrained MMT on monolingual corpora of an underrepresented language is a partial remedy towards satisfactory downstream transfer (Wang et al., 2020; Ponti et al., 2020). However, this approach does not increase the MMT capacity and, consequently, might deteriorate representations for other languages. Adapters (Houlsby et al., 2019; Bapna and Firat, 2019), additional trainable parameters inserted into the MMT's layers, have recently been used for their language and task specialization (Pfeiffer et al., 2020b), offering improved and more efficient ZS cross-lingual transfer. The current adapter-based approaches, however, do not provide any mechanism that would prevent language and task adapters from *capturing redundant information*, that is, from storing knowledge already encoded in the MMT's parameters.

In this work, we advance the idea of augmenting MMT's knowledge through specialized adapter modules. We aim to maximize the injection of *novel* information into both language- and task-specific adapter parameters, that is, we enforce the adapters to encode the information that *complements* the knowledge encoded in MMT's pretrained parameters. To achieve this, we propose to learn *orthogonal adapters* (or *orthoadapters* for short). We augment the training objective[1] with the *orthogonality loss*: it forces the representations produced by the adapters to be *orthogonal* to representations from the corresponding MMT layers, see Figure 1.

Our proof-of-concept ZS transfer experiments on POS-tagging, NER, and natural language inference (XNLI), spanning 10 typologically diverse languages, render language-specific and task-specific orthoadapters viable mechanisms for improving ZS transfer performance. However, we show that the optimal use of orthogonality is also largely task-dependent. We hope that our study will inspire a wider investigation of applicability and usefulness of orthogonality constraints for MMT fine-tuning.

[1]For a language adapter, the training task is masked language modeling on the monolingual corpus of that language.

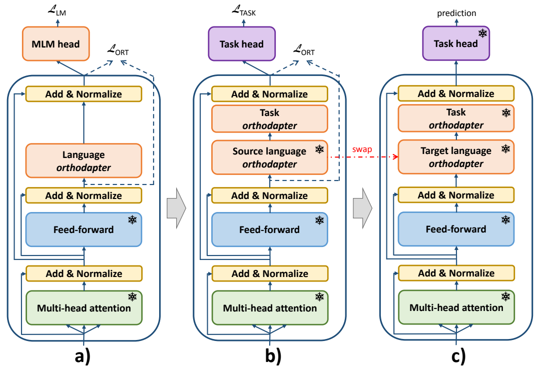

Figure 1: Orthoadapters for zero-shot cross-lingual transfer: **a)** Step 1: training language orthoadapters (independently for each language); **b)** Step 2: training the task orthoadapters on top of the (frozen) source-language orthoadapters on source-language data; **c)** Step 3: swapping the source-language orthoadapters with target language orthoadapters, allowing for task-specific target language inference. Snowflakes denote frozen parameters. For clarity, we show only a single transformer layer; the orthogonal adapter modules are used in all transformer layers of a pretrained MMT.

2 Orthogonal Adapters

Figure 1 provides an illustrative overview of our cross-lingual transfer framework for training and using language and task orthoadapters. We now provide descriptions of its components and steps.

Language and Task Adapters. *Language adapters* are injected into each transformer layer. In prior work (Pfeiffer et al., 2020b), they were trained via masked language modeling (MLM) on the monolingual corpus of the respective language. *Task adapters* are used to store task-specific knowledge during task fine-tuning in the source language.

Adapter Architecture. We adopt the well-performing and lightweight adapter configuration of Pfeiffer et al. (2021), where only one adapter module is injected per transformer layer, after the feed-forward sublayer. The concrete adapter architecture we use is the variant of the so-called bottleneck adapter (Pfeiffer et al., 2021):

$$\mathbf{x}_a = Adapt(\mathbf{x}_h, \mathbf{x}_r) = g\left(\mathbf{x_h}\mathbf{W}_d\right)\mathbf{W}_u + \mathbf{x}_r \quad (1)$$

where $\mathbf{x}_h$ and $\mathbf{x}_r$ are the hidden state and residual representation of the transformer layer, respectively. The parameter matrix $\mathbf{W}_d \in \mathbb{R}^{H \times d}$ down-projects (i.e., compresses) hidden representations to the *adapter size* $d < H$, and $\mathbf{W}_u \in \mathbb{R}^{d \times H}$ up-projects the activated down-projections back to the transformer's hidden size H; g is the non-linear ReLU activation (Nair and Hinton, 2010).

Orthogonality Loss. There is currently no mechanism in adapter-based approaches that would explicitly prevent adapter parameters from learning redundant information, already captured by the pre-trained MMT. Inspired by the idea of orthogonal text representations from prior work on multi-task learning (Romera-Paredes et al., 2012; Liu et al., 2017), we introduce an auxiliary orthogonality loss to adapter-based language and task fine-tuning. It *explicitly* forces the adapters to dedicate their capacity to *new* knowledge, which should be *complementary* (i.e., non-redundant) to the knowledge already encoded in existing MMT parameters.

Let $\mathbf{x}_h^{(i,j)}$ denote the hidden representation in the i-th layer of the MMT for the j-th token in the sequence, input to the adapter. Let $\mathbf{x}_a^{(i,j)}$ be the corresponding output of the same adapter for the same token, as given in Eq. (1). The orthogonality loss of the j-th token in the i-th MMT layer is then simply the square of the cosine similarity between $\mathbf{x}_h^{(i,j)}$ and $\mathbf{x}_a^{(i,j)}$; we then derive the overall orthogonality loss by averaging token-level losses in each layer and then summing layer-level losses:

$$\mathcal{L}_{ORT} = \sum_{i=1}^{N} \frac{1}{T} \sum_{j=1}^{T} \cos\left(\mathbf{x}_h^{(i,j)}, \mathbf{x}_a^{(i,j)}\right)^2 \quad (2)$$

where T is the maximal length of the input token sequence and N is the number of MMT's layers.

Two-Step Orthoadapter Training. We first train language orthoadapters (part **a** of Figure 1), independently for each language, aiming at extending the language knowledge in the pretrained MMT. We use MLM (cross-entropy loss) as the main training objective $\mathcal{L}_{MLM}$. We then alternately update the parameters of language orthoadapters, first by minimizing $\mathcal{L}_{MLM}$ and then $\mathcal{L}_{ORT}$.[2]

In the second step (part **b** in Figure 1), the goal is to maximize the amount of novel information useful for a concrete downstream task: we train the task orthoadapters on the task-specific training data (POS, NER, XNLI) by alternately minimizing **1)** the task-specific objective $\mathcal{L}_{TASK}$[3] and **2)** the orthogonality loss $\mathcal{L}_{ORT}$. Note, however, that in this case $\mathbf{x}_h$ is, in each transformer layer, first adapted by the source language adapter and then by the task adapter, and $\mathbf{x}_a$ is the output of the task adapter.

[2] We use two independent Adam optimizers (Kingma and Ba, 2015), one for each loss. We also experimented with minimizing the joint loss $\mathcal{L}_{MLM} + \lambda \cdot \mathcal{L}_{ORT}$ but this generally yielded poorer performance over a range of λvalues.

[3] Cross-entropy loss for the whole sequence for NLI; sum of token-level cross-entropy losses for POS and NER.

Zero-Shot Cross-Lingual Transfer then proceeds in the same vein as in prior work (Pfeiffer et al., 2020b). It is conducted by simply replacing the source language orthoadapter with the target language orthoadapter while relying on exactly the same task adapter fine-tuned with the labeled source language data, stacked on top of the language adapters (see part **c** of Figure 1).[4] [5]

3 Experimental Setup

Model Configurations. The decomposition into two adapter types in the two-step procedure (Figure 1) allows us **1)** to use language orthoadapters (L-ORT) instead of *regular non-orthogonal* language adapters (L-NOO); and/or **2)** to replace non-orthogonal task adapters (T-NOO) with task orthoadapters (T-ORT). These choices give rise to four different model variants, where the L-NOO+T-NOO variant is the baseline MAD-X variant.

We also test the usefulness of task orthoadapters in a setup without dedicated language adapters: T-ORT variants are compared to T-NOO variants, and also to standard (computationally more intensive) full fine-tuning of the whole MMT (FULL-FT).

Evaluation Tasks and Data. We evaluate all model variants on standard cross-lingual transfer tasks, relying on established evaluation benchmarks: **1)** sentence-pair classification on XNLI (Conneau et al., 2018); **2)** cross-lingual named entity recognition (NER) on the WikiANN dataset (Pan et al., 2017); **3)** part-of-speech tagging with universal POS tags from the Universal Depenedencies (Nivre et al., 2018) (UD-POS).

In all experiments we rely on the pretrained multilingual XLM-R (Base) model (Conneau et al., 2020).[6] English (EN) is our (resource-rich) source language. For completeness, we also report the results on the EN test data, i.e., without any transfer.

10 Target Languages, with their language codes available in the appendix, span 5 geographical macro-areas (Ponti et al., 2020) and 8 distinct language families. In NER evaluations we include three truly low-resource languages: Quechua, Ilocano, and Meadow Mari.[7]

Training and Evaluation: Technical Details. We rely on AdapterHub.ml (Pfeiffer et al., 2020a) built on top of the Transformers library (Wolf et al., 2020) in all experiments based on the MAD-X framework. For adapter training and configurations we follow the suggestions from prior work (Houlsby et al., 2019; Pfeiffer et al., 2020b). For language orthoadapters, we conduct MLM-ing on the Wikipedia data of each language. For task orthoadapters, we rely on the standard training portions of our task data in English.[8] The full details are available in the Appendix A.

4 Results and Discussion

The results of ZS transfer are summarized in Table 1 (XNLI), Table 2 (UD-POS), and Table 3 (NER). First, a comparison with the FULL-FT variant confirms findings from prior work (Pfeiffer et al., 2021), validating the use of the more efficient adapter-based approach: the ZS scores with adapter-based variants are on a par with or even higher than the scores reported with FULL-FT across the board.

Regular vs Orthogonal Language Adapters. First, the usefulness of language orthoadapters (L-ORT variants) does depend on the task at hand and its complexity. As an encouraging finding, we observe consistent gains in cross-lingual NLI:[9] at least +1 accuracy point on 4/5 target languages with the L-ORT+T-NOO variant. This variant also yields highest average ZS performance, and slight (but statistically insignificant) gains on EN NLI. The picture is less clear for UD-POS and NER: L-ORT+T-NOO does have a slight edge over the baseline L-NOO+T-NOO variant in UD-POS, but this seems to be due to large gains in Chinese. In a similar vein, while L-ORT+T-NOO is the best performing variant in NER on average, the gains over L-NOO+T-NOO are slight, and inconsistent across

[4] See (Pfeiffer et al., 2020b,a) for further technical details.

[5] Besides applications in cross-lingual transfer, adapters have also been recently investigated in the NMT literature (Vilar, 2018; Bapna and Firat, 2019; Philip et al., 2020); this application scenario is beyond the scope of this work.

[6] XLM-R showed state-of-the-art zero-shot performance in a recent comparative empirical study of Hu et al. (2020), and even stronger results when combined with the adapter-based MAD-X framework (Pfeiffer et al., 2020b).

[7] The selection of target languages has been guided by several (sometimes clashing) criteria: **C1)** typological diversity; **C2)** availability in the standard evaluation benchmarks; **C3)** computational tractability; **C4)** evaluation also on truly low-resource languages. Given that the main computational bottleneck is MLM-ing for learning language adapters, we have started from the subset of languages represented in our evaluation datasets (C2) for which pretrained language adapters (regular, non-orthogonal) are already available online (C3) (Pfeiffer et al., 2020a), also respecting C1 and C4.

[8] We select task (ortho)adapters solely based on the performance on the source language (i.e, English) dev set.

[9] XNLI is arguably the most complex (reasoning) task in our evaluation and, unlike UD-POS and NER, requires successful high-level semantic modeling and ZS transfer.

Variant	EN	AR	HI	SW	TR	ZH	AVGz
FULL-FT	83.67	72.01	68.64	63.77	71.75	73.11	69.86
T-NOO	84.05	69.51	68.26	64.48	71.73	72.25	69.25
T-ORT	84.25	69.97	68.84	63.35	70.85	71.95	68.99
L-NOO+T-NOO	84.59	71.17	70.61	67.68	71.75	72.29	70.70
L-NOO+T-ORT	84.79	68.88	69.71	66.34	70.47	71.49	69.38
L-ORT+T-NOO	84.73	72.25	69.28	69.10	72.89	73.61	71.43
L-ORT+T-ORT	84.35	69.73	68.56	67.92	71.23	71.53	69.79

Table 1: Accuracy scores ($\times 100\%$) of zero-shot transfer for the natural language inference task on the XNLI dataset. See §3 for the descriptions of different model variants. EN is the source language in all experiments. The scores in the AVG*z* column denote the average performance of zero-shot transfer (i.e., without English results).

Variant	EN	AR	ET	HI	TR	ZH	AVGz
FULL-FT	95.94	66.42	84.68	70.38	74.01	35.59	66.22
T-NOO	95.59	64.35	84.43	71.06	72.68	31.47	64.80
T-ORT	95.65	65.28	85.17	70.42	72.93	40.03	66.77
L-NOO+T-NOO	95.59	65.77	85.81	69.62	74.17	19.25	62.92
L-NOO+T-ORT	95.66	67.15	85.67	71.57	74.08	31.68	66.03
L-ORT+T-NOO	95.63	66.62	84.26	68.93	72.02	29.50	64.27
L-ORT+T-ORT	95.63	67.40	84.22	67.26	71.21	31.79	64.38

Table 2: F_1 scores ($\times 100\%$) of zero-shot transfer in the UD-POS task.

Variant	EN	AR	ET	HI	ILO	MHR	QU	SW	TR	ZH	AVGz
FULL-FT	82.98	30.85	65.63	58.87	62.04	39.83	60.43	61.26	65.98	8.77	50.41
T-NOO	83.41	46.19	70.86	67.14	60.75	39.33	58.01	60.79	72.60	23.07	55.42
T-ORT	82.87	46.71	70.27	66.16	59.03	45.30	58.72	61.26	72.93	19.69	55.56
L-NOO+T-NOO	82.33	41.29	75.04	64.84	63.11	54.55	70.64	71.90	71.63	16.68	58.85
L-NOO+T-ORT	82.86	38.74	74.48	64.21	69.72	53.01	61.02	72.78	70.65	14.41	57.67
L-ORT+T-NOO	82.44	40.62	74.27	66.74	77.21	50.78	65.31	72.98	70.80	12.16	58.99
L-ORT+T-ORT	82.63	35.82	72.81	61.83	70.64	52.31	64.98	73.84	69.41	12.25	57.10

Table 3: F_1 scores ($\times 100\%$) of zero-shot transfer in the NER task on the WikiAnn dataset.

languages (e.g., large gains on ILO, some on AR and SW, but some decrease on QU and MHR). We again speculate that this is mostly due to the nature and complexity of the task at hand.[10]

Orthogonal Task Adapters display a different behavior, but we can again largely relate it to the properties of the evaluation tasks. First, task orthoadapters seem detrimental for XNLI (compare L-NOO+T-ORT vs. L-NOO+T-NOO as well as L-ORT+T-ORT vs. L-ORT+T-NOO in Table 1), and also yield no real benefits in the simpler setup (T-ORT vs. T-NOO). The two main objectives – (i) MLM for the original MMT pretraining and language adapter training, and (ii) cross-entropy loss for the whole sequence for NLI – might be structurally too different for the orthogonality loss to capture any additional task-related information.[11] However, task orthoadapters seem to be useful UD-POS, with substantial gains reported on 3/5 languages – Arabic, Chinese, Hindi, all of which have non-Latin scripts. Combining them with language orthoadapters, however, does deteriorate the performance. The overall trend is even more complex with NER: while there are clear hints that using orthoadapters is useful for some languages and some model variants, there is still a substantial variance in the results.[12]

5 Conclusion

We investigated how orthogonality constraints impact zero-shot (ZS) cross-lingual transfer via massively multilingual transformers (MMTs, e.g., XLM-R) for three standard tasks: NLI, POS, and NER. Relying on the standard adapter-based transfer techniques, we introduced the idea of orthogonal language and task adapters (or orthoadapters): we explicitly enforce the information stored in the parameters of the orthoadapters to be orthogonal to the information already stored in the pretrained MMT. In general, our results suggest that explicitly controlling for the information that gets captured in the orthoadapters can have a positive impact on ZS transfer via MMTs. The use of orthogonality, however, seems to be language- and task-dependent, warranting further investigations in future work. The code will be available at: `[URL]`.

[10] In order to perform cross-lingual transfer for NLI, the underlying MMT must capture and leverage more language-specific nuances than for sequence labeling tasks such as POS-tagging and NER. By enforcing the capture of non-redundant information in the additional language-specific adapters, we allow the model to store additional and, more importantly, *novel* target language information. While the same information is available also for NER and POS tagging, they require 'shallower' language-specific knowledge (Lauscher et al., 2020); this is why more complex target language-specific knowledge captured in orthoadapters (compared to regular non-orthogonal language adapters) does not make a difference.

[11] In fact, we speculate that the orthogonal loss might have emphasized this discrepancy between the objectives.

[12] We partially attribute it to the documented volatility of WikiAnn for low-resource languages (Pfeiffer et al., 2020b).

References

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George F. Foster, Colin Cherry, Wolfgang Macherey, Zhifeng Chen, and Yonghui Wu. 2019. Massively multilingual neural machine translation in the wild: Findings and challenges. *CoRR*, abs/1907.05019.

Ankur Bapna and Orhan Firat. 2019. Simple, scalable adaptation for neural machine translation. In *Proceedings of EMNLP-IJCNLP 2019*, pages 1538–1548.

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 ACL 2020*, pages 8440–8451.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of EMNLP 2018*, pages 2475–2485.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT 2019*, pages 4171–4186.

Goran Glavaš, Robert Litschko, Sebastian Ruder, and Ivan Vulić. 2019. How to (properly) evaluate cross-lingual word embeddings: On strong baselines, comparative analyses, and some misconceptions. In *Proceedings of ACL 2019*, pages 710–721.

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 ICML 2019*, pages 2790–2799.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. XTREME: A massively multilingual multi-task benchmark for evaluating cross-lingual generalization. In *Proceedings of ICML 2020*.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of ICLR 2015*.

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 EMNLP 2020*, pages 4483–4499.

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2017. Adversarial multi-task learning for text classification. In *Proceedings of ACL 2017*, pages 1–10.

Vinod Nair and Geoffrey E. Hinton. 2010. Rectified linear units improve restricted Boltzmann machines. In *Proceedings of ICML 2010*, pages 807–814.

Joakim Nivre, Mitchell Abrams, Željko Agić, Lars Ahrenberg, Lene Antonsen, Maria Jesus Aranzabe, Gashaw Arutie, Masayuki Asahara, Luma Ateyah, Mohammed Attia, et al. 2018. Universal Dependencies 2.2.

Xiaoman Pan, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, and Heng Ji. 2017. Cross-lingual name tagging and linking for 282 languages. In *Proceedings of ACL 2017*, pages 1946–1958.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. 2021. AdapterFusion: Non-destructive task composition for transfer learning. In *Proceedings of EACL 2021*, pages 487–503.

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 EMNLP 2020: System Demonstrations*, pages 46–54.

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 EMNLP 2020*, pages 7654–7673.

Jerin Philip, Alexandre Berard, Matthias Gallé, and Laurent Besacier. 2020. Monolingual adapters for zero-shot neural machine translation. In *Proceedings of EMNLP 2020*, pages 4465–4470.

Edoardo M. Ponti, Goran Glavaš, Olga Majewska, Qianchu Liu, Ivan Vulić, and Anna Korhonen. 2020. XCOPA: A multilingual dataset for causal commonsense reasoning. In *Proceedings of EMNLP 2020*, pages 2362–2376.

Afshin Rahimi, Yuan Li, and Trevor Cohn. 2019. Massively multilingual transfer for NER. In *Proceedings of ACL 2019*, pages 151–164.

Bernardino Romera-Paredes, Andreas Argyriou, Nadia Berthouze, and Massimiliano Pontil. 2012. Exploiting unrelated tasks in multi-task learning. In *Proceedings of AISTATS 2012*, pages 951–959.

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

David Vilar. 2018. Learning hidden unit contribution for adapting neural machine translation models. In *Proceedings of NAACL-HLT 2018*, pages 500–505.

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

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 EMNLP 2020: System Demonstrations*, pages 38–45.

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

A Training Details

A.1 Task Training Details

We processed the data for all tasks using the preprocessing pipeline provided with the XTREME benchmark (Hu et al., 2020).[13]

XNLI. In NLI training (i.e., for XNLI transfer) were trained for 30 epochs with the batch size of 32. Maximum sequence length was 128 input tokens. Gradient norms were clipped to 1.0.

UD-POS. We trained for 50 epochs with batch size of 16. Maximum sequence length was 128 input tokens. Gradient norms were clipped to 1.0.

NER. We trained for 100 epochs with the batch size of 16. Maximum sequence length was 128 input tokens. Gradient norms were clipped to 1.0. As prior work, we use the data splits of Rahimi et al. (2019).

A.2 Experimental Setup without Language Adapters

For the "non-MAD-X" experimental setup (i.e., the setup without language adapters, see §3), we relied on our own implementation of the adapter module. The bottleneck size for the task adapter was set to $d = 64$. For (X)NLI, we searched the following learning rate grid: $[2e-5, 5e-5, 7e-5]$; for UD-POS and WikiAnn the corresponding learning rate grid was $[5e-5, 1e-4]$. For task orthoadapters, we searched the following additional learning rate grid for the orthogonal loss optimizer: $[1e-6, 1e-7, 1e-8]$.

A.3 Full Experimental Setup

For the more complex multi-adapter setup based on the MAD-X framework (i.e., with both language and task adapters, see Figure 1), we utilized the *Adapter-Transformers* library and the underlying *AdapterHub* service (Pfeiffer et al., 2020a).

Task Adapters and Orthoadapters. We followed the recommendation from the original paper (Pfeiffer et al., 2020b). We utilized the *Pfeiffer* configuration[14] found in the Adapter-Transformers library with the adapter dimensionality of 48. Due to the computational constraints, the learning rate grid search took into account best settings observed in the baseline experiments. For XNLI our main learning rate was set at a well-performing $5e-5$. For UD-POS and WikiAnn, due to more instability, we tested the learning rate grid of $[5e-5, 1e-4]$. For the task orthoadapters, we used the same learning rate for the orthogonality loss as for the non-MAD-X setup: $[1e-6, 1e-7, 1e-8]$.

Regular Language Adapters. We utilized the pretrained language adapters readily available via the AdapterHub service (Pfeiffer et al., 2020a). These language adapters have the dimensionality of 384. They were trained (while the rest of the model was frozen) by executing the MLM-ing for 250.000 iterations on the Wikipedia data in the target language.

Orthogonal Language Adapters. We started from the MLM-ing training script for training language adapters provided by the Adapter-Transformers library and trained language orthoadapters on the Wikipedia data, relying on the setup of AdapterHub's regular language adapters (dimensionality 384, 250,000 iterations). Due to computational constraints we reduced the maximum sequence length of the input to 128 tokens, while the batch size was 8. Finally, for the main optimizer and orthogonality loss optimizer we used the learning rates of $1e-4$ and $1e-7$, respectively.

A.4 Statistical Significance Testing

Statistical significance ($p < 0.05$) is reported following the recommended statistical significance tests for each task, see: `https://arxiv.org/pdf/1809.01448.pdf`.

[13] `https://github.com/google-research/xtreme`

[14] Pfeiffer et al. (2021) found this configuration to perform on a par with the configuration proposed by Houlsby et al. (2019), who inject two adapter modules per transformer layer (the other one after the multi-head attention sublayer), while being more efficient to train.

Language	Family	Type	ISO 639	Tasks
English	IE: Germanic	fusional	EN	XNLI, UD-POS, NER
Arabic	Semitic	introflexive	AR	XNLI, UD-POS, NER
Estonian	Uralic: Finnic	agglutinative	ET	UD-POS, NER
Hindi	IE: Indo-Aryan	fusional	HI	XNLI, UD-POS, NER
Ilocano	Austronesian	agglutinative	ILO	NER
Meadow Mari	Uralic: Mari	agglutinative	MHR	NER
Quechua	Quechuan	agglutinative	QU	NER
Kiswahili	Niger-Congo: Bantu	agglutinative	SW	XNLI, NER
Turkish	Turkic	agglutinative	TR	XNLI, UD-POS, NER
Mandarin Chinese	Sino-Tibetan	isolating	ZH	XNLI, UD-POS, NER

Table 4: Target languages used in the main experiments along with their corresponding language family (IE=Indo-European), morphological type, and ISO 639-1 code (or ISO 639-2 for Ilocano; or ISO 639-3 for Meadow Mari). We use English (EN) as the source language in all experiments. EN is a fusional language (IE: Germanic).