FAD-X: Fusing Adapters for Cross-lingual Transfer to Low-Resource Languages

Jaeseong Lee[1], **Seung-won Hwang**[1*] and **Taesup Kim**[2]
[1]Computer Science and Engineering, Seoul National University
[2]Graduate School of Data Science, Seoul National University
{tbvj5914,seungwonh,taesup.kim}@snu.ac.kr

Abstract

Adapter-based tuning, by adding light-weight adapters to multilingual pretrained language models (mPLMs), selectively updates language-specific parameters to adapt to a new language, instead of finetuning all shared weights. This paper explores an effective way to leverage a public pool of pretrained language adapters, to overcome resource imbalances for low-resource languages (LRLs). Specifically, our research questions are, whether pretrained adapters can be composed, to complement or replace LRL adapters. While composing adapters for multi-task learning setting has been studied, the same question for LRLs has remained largely unanswered. To answer this question, we study how to fuse adapters across languages and tasks, then validate how our proposed fusion adapter, namely FAD-X, can enhance a cross-lingual transfer from pretrained adapters, for well-known named entity recognition and classification benchmarks. [1]

1 Introduction

While fine-tuning the multilingual pretrained language models (mPLMs), such as mBERT (Devlin et al., 2019) or XLM-R (Conneau et al., 2020) has become a de-facto standard to tackle diverse language tasks, task performance in low-resource languages lags behind, due to resource imbalances (Wu and Dredze, 2020).

To overcome this challenge, MAD-X (Pfeiffer et al., 2020) tackles such performance degradation as a capacity issue, and adopts the idea of adapters (Houlsby et al., 2019). For a new language (or a task), they add a few parameters to adapt, while keeping parameters for mPLMs frozen. This approach enables a parameter-efficient adaptation to a new language or task, by tuning only language- and task-specific parameters, which can also be released as pretrained adapters.

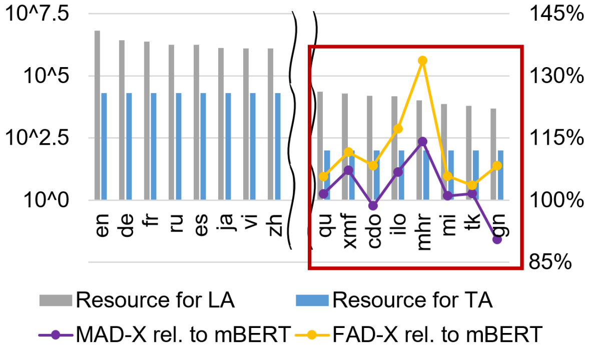

Figure 1: Bar graph: statistics of training resources for language adapters (LAs) and task adapters (TAs), in log scale. Line graph: relative F1 scores (%) of MAD-X and proposed FAD-X, compared to mBERT fine-tuning performance. We target LRLs in the red box, with resources for both LA/TA being orders of magnitude smaller.

However, we argue that a significant resource imbalance yet remains, especially for LRLs. To illustrate, Figure 1 shows 8 highest/lowest resource languages among those with pretrained adapters. The gray bar suggests training resources for LA (Wikipedia articles written in each language) and the blue bar suggests those for TA (WikiAnn in Section 3.2), which are dominated by high-resource languages, especially English. This suggests that pretrained adapters for our target problem of LRLs (shown in the red box), are trained from resources that are multiple orders of magnitude smaller: For example, in Figure 1, resources for TA/LA for gn are up to 20-fold and 1000-fold smaller respectively, which causes a negative transfer of MAD-X, to underperform mBERT baseline (shown in purple line). More significantly, the amount of languages supported by adapters (40+) is much less than that of mBERT (100+), and even more significantly less than 6500+ languages that need to be supported. These observations present two chal-

[*] Corresponding author
[1] Code is available at https://github.com/thnkinbtfly/FAD-X.

Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), pages 57–64
November 20–23, 2022. ©2022 Association for Computational Linguistics

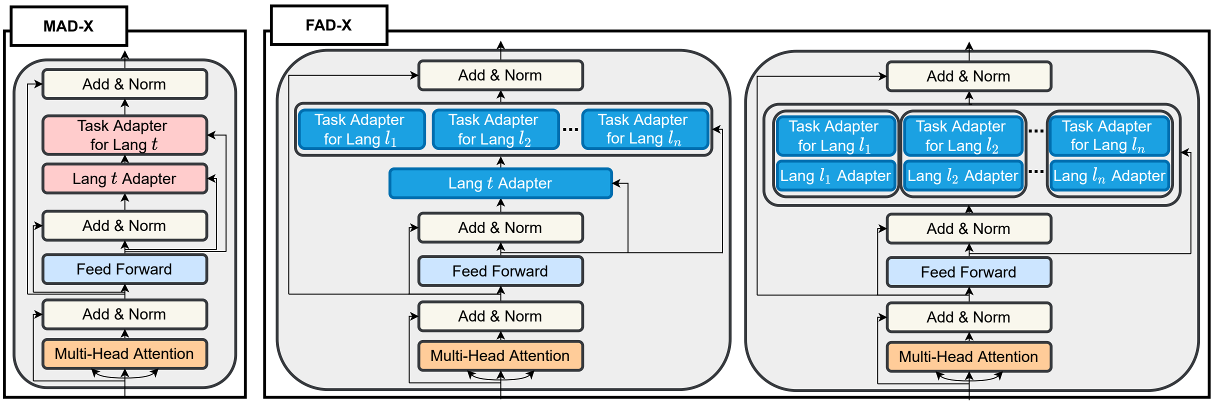

Figure 2: Comparison of FtP (middle) and PtF (right) of FAD-X, and MAD-X (left) architecture.

lenges for LRLs, (a) pretrained LA may not exist, or exist with poor quality, and (b) task-specific resource is also scarce.

In this paper, we propose **F**using multiple **AD**apters for cross-lingual transfer (**FAD-X**), to overcome imbalances, by transferring from both LA and TA resources available for higher-resource languages.

Inspired by multilingual PLM outperforming monolingual PLM for LRLs from a cross-lingual transfer (Wu and Dredze, 2020; Muller et al., 2021; Chau and Smith, 2021), we study whether such a transfer among adapters can be effective. Specifically, we study whether pretrained LAs can be fused to complement LRLs with lower-quality LA, or even to support those with no adapter.

Toward this goal, given the pool of pretrained adapters L and target language t, we propose to utilize pretrained language adapter $LA_{l_i} \in L$, to train task adapter per each language, denoted as TA_{l_i}. We show that fusing such task adapters contributes to overcoming limited training resources, in training TA in the target language (the yellow line in Figure 1 ensures positive transfers in all LRLs with larger gains than MAD-X).

Contributions Our contributions are as follows:

- We devise FAD-X, a method to fuse adapters trained from different languages.
- We propose two designs to fuse language and task adapters, and evaluate the effectiveness on two different tasks; For LRLs, we improve +5.3% F1 on WikiAnn and +16.5% accuracy on Amazon Review dataset, on average.
- We also validate FAD-X, in a more resource-constrained setting, where LA does not exist for the target language.

2 Proposed Method

2.1 Preliminaries

We first briefly review MAD-X (Pfeiffer et al., 2020) architecture (left of Figure 2). For each layer in a given PLM, MAD-X adds two adapters; language adapter (LA) and task adapter (TA). When h is the output of the original transformer layer, MAD-X first alters output as $LA(h)$, and updates the parameters of LA using unlabelled data in language t (Resource for LA in Figure 1), to obtain LA_t. Then, parameters for TA are trained from resource for TA shown in Figure 1, from $TA(LA_t(h))$ to produce TA_t. However, MAD-X suffers when resources for LA/TA are scarce, as shown in the LRLs in the red box in Figure 1.

2.2 FAD-X

To overcome the lack of resources for LA/TA observed for LRLs, we propose FAD-X. Our key idea is fusing task adapters trained with pretrained adapters in other languages.

More formally, given a pool of n pretrained adapters, $L = \{LA_{l_1}, \cdots, LA_{l_n}\}$, our goal is fusing TA_{l_i} trained from each language adapter LA_{l_i}, which can be implemented as one of the following two designs, as also illustrated in Figure 2:

- Fused then Paired (FtP): We first fuse task adapters $F(TA_{l_1}, \cdots, TA_{l_n})$, then pair with target language adapter LA_t, or, $Fuse = F(TA_{l_1}, \cdots, TA_{l_n}) \circ LA_t$.
- Paired then Fused (PtF): Each task adapter TA is paired by language adapter LA used for training, or, $Fuse = F(TA_{l_1} \circ LA_{l_1}, \cdots, TA_{l_n} \circ LA_{l_n})$.

where $F(A_1, \cdots, A_n)$ is formulated as AdapterFusion module (Pfeiffer et al., 2021) as follows:

$$s_i = \text{softmax}(h^T Q \otimes A_i(h)^T K) \quad (1)$$

$$z_i = A_i(h)^T V, i \in 1, \cdots, n \quad (2)$$

$$F(A_1, \cdots, A_n)(h) = \sum_i s_i z_i \quad (3)$$

In the above equation, $\otimes$ denotes the dot product, and Q, K, and V represent the learnable query, key, and value matrices. With the proposed architecture, we can fully utilize other available pretrained adapters.

3 Experiments

3.1 Setup

Datasets We used two datasets to confirm the effect of our proposed method, FAD-X. **WikiAnn** (Pan et al., 2017) is a multilingual dataset for named-entity recognition (NER). We use the split with balanced labels (Rahimi et al., 2019) which covers 176 languages. The size of the dataset highly differs over languages; As Figure 1 shows, high-resource languages may have up to 20,000 examples for training, while low-resource languages usually have only 100 examples. The **Multilingual Amazon Reviews Corpus Dataset** (Keung et al., 2020) contains reviews of items where the user can give one to five stars to each record. There are 200,000, 5,000, and 5,000 reviews in train, validation, and test sets for each language, respectively. We simulate LRLs by random sampling 1% of the train datasets, which corresponds to 2,000 examples.

Languages For experiments conducted with WikiAnn dataset, we select LRLs used in (Pfeiffer et al., 2020) as target LRLs. We set L by collecting one HRL per each language family. For the experiment with Amazon Reviews dataset, we set L as all languages except for the simulated target LRL. We further describe the selected languages in the Appendix.

Methods For given language t, we compare three methods.

- $Fuse(L)$: Fusion of adapters pretrained on languages L, following our proposed method FAD-X.
- $S(t)$: A baseline which stacks TA_t with LA_t, following a state-of-the-art method, MAD-X.
- $S(t)$ w/ param+: A baseline which uses adapters with same additional parameters as $Fuse(L)$.

Experimental Settings To train TA_l for WikiAnn in each language l, we use batch size of 16, learning rate of 2e-5, and train for 100 epochs then select best checkpoint based on the validation F1 score. We conduct each experiment 5 times and report the average test F1 score. We use multilingual BERT (Devlin et al., 2019) with 104 languages for this experiment. To train on Amazon Reviews dataset, we use multilingual BERT and XLM-R (Conneau et al., 2020) as the base models, and use batch size of 32, learning rate of 1e-5. We train for 15 epochs following (Keung et al., 2020). All experiments are run 5 times and we report the average test accuracy.

Scenarios We consider two possible scenarios:

- $LA_t \in L$. We conjecture that, with knowledge transfer from adapters trained in other languages, fused adapters outperform using LA_t only.
- $LA_t \notin L$ (no adapter). LA_t is proxied by that of some l_i in L, which we select the HRL in same language family, or English if isolated.

3.2 Analysis on WikiAnn

$LA_t \in L$: Combining LA_t with others in L was complementary for all target languages (Table 1).

	qu	cdo	ilo	xmf	mhr	mi	tk	gn	avg
mPLM (Pfeiffer et al., 2020)	71.80	48.30	80.20	63.20	61.70	87.10	69.20	62.90	68.05
S(t) (Pfeiffer et al., 2020)	72.90	51.80	79.10	67.50	70.40	88.00	70.30	56.90	69.61
S(t)	70.22	53.00	81.27	69.11	71.09	86.95	68.63	62.61	70.36
S(t) w/ param+	67.46	56.33	80.37	70.50	69.75	90.12	67.86	62.88	70.66
Fuse(L)	**75.88***	**53.90**	**86.88**	**74.08**	**82.49**	**92.19***	**71.67**	**68.11***	**75.65**

Table 1: $LA_t \in L$ results on WikiAnn. w/ param+: add the same number of parameters as in Fuse(L). *: Use PtF architecture, based on Table 3.

	qu	cdo	ilo	xmf	mhr	mi	tk	gn	avg
S(t)	70.22	53.00	81.27	69.11	71.09	86.95	68.63	62.61	70.36
Fuse(L-LA$_t$)	81.01	50.35	85.75	71.06	66.84	92.69	71.34	74.18	**74.15**
Fuse(L-LA$_t$) w/ ml	76.01	51.55	84.73	65.09	66.68	92.00	70.53	71.43	72.25

Table 2: $LA_t \notin L$ results on WikiAnn. w/ml: use most resource-abundant languages without consideration of language families.

scenario	arch	qu	cdo	ilo	xmf	mhr	mi	tk	gn
Fuse(L)	FtP	66.32	**55.96**	**88.82**	**71.56**	**83.09**	86.13	**77.20**	61.40
	PtF	**72.70**	52.50	86.66	68.56	71.45	**90.23**	73.52	**66.03**
Fuse(L-LA$_t$)	FtP	**72.89**	**56.70**	**91.79**	**73.45**	**72.69**	**90.34**	**75.66**	**69.05**
	PtF	70.24	55.79	88.64	70.06	70.82	**90.70**	71.14	65.93

Table 3: Average val F1 scores in WikiAnn, comparing PtF and FtP designs.

$LA_t \notin L$: Alternatively, we assume LA_t does not exist and fuse only $L - LA_t$. Table 2 shows that such fusion outperforms the baseline on average.

Parameter Efficiency: We investigate whether our improvement comes from an increase of parameters– We add the same number of parameters as Q, K, V in the fusion module to S(t), described in the row named '$S(t)$ w/ param+' in Table 1.

Though such an increase does improve results for some languages, it often negatively impacts the performance as well. This indicates that our fusion model proposes an effective use of increased parameters.

Selection of HRLs for fusion: This section explores an alternative of choosing one HRL in the same family (as discussed in Section 3.1), by selecting the most resourced language (ml) regardless of the family. Row named 'Fuse(L-LA$_t$) w/ ml' in Table 2 reveals the performance of such variant. It is inferior to our original selection, by collecting HRLs from multiple families. This indicates the diversity of fusing multiple language families enhances the cross-lingual transfer.

FtP vs PtF: In Section 2, we proposed two designs to fuse with HRL adapters, FtP and PtF. We investigate which approach is better with validation scores in WikiAnn, revealed in Table 3. Surprisingly, PtF cannot provide better performance than FtP in most scenarios, even though it uses more adapters. The only exceptions are qu, mi, gn.

We investigated whether these exceptions correlate with phonological similarity, which is studied to highly correlate with cross-lingual transfer performance of WikiAnn (Lauscher et al., 2020). This is computed as cosine similarity between URIEL phonology vectors (Littell et al., 2017). Table 4 reports the similarity of each language to closest HRL– Three languages with the lowest scores are shown in **bold**, where qu and gn are "isolated" without a HRL in the same family, and mi is closer to a HRL in another family. Though we leave deeper analysis as a future work, this predicts languages where FtP underperform.

LRL	qu	cdo	ilo	xmf	mhr	mi	tk	gn
sim	**0.80**	0.89	0.85	0.93	0.91	**0.67**	1.00	**0.75**

Table 4: Linguistic similarity between each target LRL and closest HRL.

	ja
mPLM	**73.2**
S(t)	71.7
Fuse(L)	72.7

Table 5: WikiAnn result in resource-abundant scenario.

Importance of resource-imbalanced scenario: Our conjecture is that FAD-X helps MAD-X outperform mPLM baselines, when the resource for LA or TA lags behind. To verify, we evaluate FAD-X when such condition is violated. Table 5 shows that in resource-abundant situations, although fusion complements the adapters, it does not outperform the mPLM.

3.3 Analysis on Amazon Reviews

We further verify previous observations with Amazon Reviews dataset. We perform same analyses, as long as supported by this dataset.

$LA_t \in L$: Similar to WikiAnn results, LAs in L help LA_t, for all target languages (Table 6). On average, we observe 12% increase for mBERT, and 16.8% accuracy increase for XLM-R.

	mBERT							XLM-R					
	en	zh	de	ja	es	fr	avg	en	zh	de	ja	es	avg
mPLM	44.94	41.91	45.30	39.92	45.10	44.11	43.55	52.42	48.09	52.94	49.36	51.70	50.90
S(t)	36.61	34.06	37.62	31.67	35.40	35.03	35.06	35.60	38.19	36.40	38.51	34.02	36.55
S(t) param+	45.32	42.48	44.91	39.40	44.77	44.49	43.56	48.68	45.74	48.81	46.53	48.15	47.58
Fuse(L)	**49.34**	**45.18**	**41.98**	**48.98**	**48.82**	**48.48**	**47.13**	**54.72**	**50.95**	**51.40**	**54.20**	**55.48**	**53.35**

Table 6: $LA_t \in L$ results on Amazon Multi Review dataset with simulated low-resource scenario.

	mBERT							XLM-R					
	en	zh	de	ja	es	fr	avg	en	zh	de	ja	es	avg
S(t)	36.61	34.06	37.62	31.67	35.40	35.03	35.06	35.60	38.19	36.40	38.51	34.02	36.55
Fuse(L-LA$_t$)	**49.23**	**45.44**	**42.28**	**48.88**	**48.74**	**48.06**	**47.10**	**54.79**	**50.84**	**51.48**	**54.11**	**55.06**	**53.26**

Table 7: $LA_t \notin L$ results on Amazon Multi Review dataset with simulated low-resource scenario.

$LA_t \notin L$: LAs in $L - LA_t$ could substitute LA_t (Table 7), which is consistent with WikiAnn results.

Parameter Efficiency: Again, we examine whether the parameter increment is the main cause for the enhanced performance. By comparing last two rows of Table 6 we can observe that, although more parameters could lead to better performance, FAD-X could utilize the given parameters more efficiently.

FtP vs PtF: We investigate whether FtP outperform PtF consistently over various train data sizes, with mBERT. We additionally build train sets by randomly sampling 0.1% and 10% of the original train datasets. Table 8 shows that, FtP generally outperforms PtF over diverse train data sizes.

4 Related Work

Adapters Adapters proposed for domain adaptations in computer vision tasks (Rebuffi et al., 2017, 2018), have been successful for language tasks, as a parameter-efficient alternative to fine-tuning PLMs, specifically for task (Houlsby et al., 2019) and domain adaptation (Bapna and Firat, 2019), avoiding catastrophic forgetting (Santoro et al., 2016). The closest work to ours is, AdapterFusion (Pfeiffer et al., 2021) combines the representations from several task adapters for monolingual target tasks. Our distinction is enabling a cross-lingual transfer across multiple language and task adapters.

Cross-lingual transfer A de-facto cross-lingual transfer is finetuning PLMs: mBERT (Devlin et al., 2019), XLM-R (Conneau et al., 2020), or mT5 (Xue et al., 2021), while MAD-X (Pfeiffer et al., 2020), leveraging three types of adapters: language, task, and invertible adapters, have been its parameter-efficient alternative. Our contribution is observing the weaknesses of MAD-X for LRLs, and presenting a fusion to overcome such weaknesses.

5 Conclusion

We proposed FAD-X, fusing multiple pretrained adapters, for a cross-lingual transfer to LRLs, overcoming the imbalances in resources for LA/TA. We validate the effectiveness of our approach, for LRLs with no pretrained adapter or that trained with limited resources.

References

Ankur Bapna and Orhan Firat. 2019. Simple, Scalable Adaptation for Neural Machine Translation. In *Proceedings of the 2019 Conference on Empirical*

		Fuse(L)						Fuse(L-LA$_t$)					
data size	arch	en	zh	ja	es	de	fr	en	zh	ja	es	de	fr
0.1%	FtP	**45.08**	**41.63**	**38.74**	**44.07**	**43.63**	**44.30**	**44.72**	**41.62**	**38.81**	**44.02**	**43.22**	**43.85**
	PtF	43.03	39.76	36.25	42.01	41.36	42.72	43.38	39.91	36.14	42.44	42.61	42.37
1%	FtP	**48.99**	**44.49**	**43.42**	**48.58**	**48.47**	**48.17**	**48.59**	**44.55**	**43.83**	**48.33**	**47.92**	**47.99**
	PtF	48.01	43.50	42.20	47.56	47.20	47.92	47.96	44.31	42.79	47.52	47.90	47.81
10%	FtP	**52.58**	**47.58**	**48.24**	**52.58**	**52.70**	**52.31**	**52.78**	**47.60**	**48.06**	**52.17**	**52.99**	**51.86**
	PtF	52.01	47.02	47.38	51.60	51.92	51.40	52.68	47.54	47.39	51.45	52.76	51.38

Table 8: Average val accuracy on Amazon Reviews with mBERT, comparing PtF with FtP over diverse data sizes.

Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 1538–1548, Hong Kong, China. Association for Computational Linguistics.

Ethan C. Chau and Noah A. Smith. 2021. Specializing Multilingual Language Models: An Empirical Study. In *Proceedings of the 1st Workshop on Multilingual Representation Learning*, pages 51–61, Punta Cana, Dominican Republic. Association for Computational Linguistics.

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.

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.

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*, pages 2790–2799. PMLR.

Phillip Keung, Yichao Lu, György Szarvas, and Noah A. Smith. 2020. The Multilingual Amazon Reviews Corpus. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4563–4568, 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.

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

Benjamin Muller, Antonios Anastasopoulos, Benoît Sagot, and Djamé Seddah. 2021. When Being Unseen from mBERT is just the Beginning: Handling New Languages With Multilingual Language Models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 448–462, Online. Association for Computational Linguistics.

Xiaoman Pan, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, and Heng Ji. 2017. Cross-lingual Name Tagging and Linking for 282 Languages. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1946–1958, Vancouver, Canada. Association for Computational Linguistics.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. 2021. AdapterFusion: Non-Destructive Task Composition for Transfer Learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 487–503, Online. Association for Computational Linguistics.

Jonas Pfeiffer, 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.

Afshin Rahimi, Yuan Li, and Trevor Cohn. 2019. Massively Multilingual Transfer for NER. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 151–164, Florence, Italy. 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.

Sylvestre-Alvise Rebuffi, Andrea Vedaldi, and Hakan Bilen. 2018. Efficient Parametrization of Multi-domain Deep Neural Networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8119–8127, Salt Lake City, UT. IEEE.

Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. 2016. Meta-learning with memory-augmented neural networks. In *Proceedings of the 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1842–1850, New York, New York, USA. PMLR.

Shijie Wu and Mark Dredze. 2020. Are All Languages Created Equal in Multilingual BERT? In *Proceedings of the 5th Workshop on Representation Learning for NLP*, pages 120–130, Online. Association for Computational Linguistics.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A Massively Multilingual Pre-trained Text-to-Text Transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

	lang	language family	R for TA	R for LA
HRLs	**English (en)**	Indo-European	20K	6.24M
	Vietnamese (vi)	Austroasiatic	20K	1.26M
	Chinese (zh)	Sino-Tibetan	20K	1.18M
	Arabic (ar)	Afro-Asiatic	20K	1.10M
	Indonesian (id)	Austronesian	20K	0.56M
	Finnish (fi)	Uralic	20K	0.50M
	Turkish (tr)	Turkic	20K	0.39M
	Georgian (ka)	Kartvelian	10K	0.15M
	German (de)	Indo-European	20K	2.53M
	French (fr)	Indo-European	20K	2.30M
	Russian (ru)	Indo-European	20K	1.70M
	Spanish (es)	Indo-European	20K	1.66M
	Japanese (ja)	Japonic	20K	1.25M
LRLs	Quechua (qu)	Quechua	0.1K	22k
	Min Dong (cdo)	Sino-Tibetan	0.1K	15k
	Ilokano (ilo)	Austronesian	0.1K	14k
	Mingrelian (xmf)	Kartvelian	0.1K	13k
	Meadow Mari (mhr)	Uralic	0.1K	10k
	Maori (mi)	Austronesian	0.1K	7k
	Turkmen (tk)	Turkic	0.1K	6k
	Guarani (gn)	Tupian	0.1K	4k

Table 9: Languages we used for WikiAnn experiments. Bolded HRLs are the languages used for fusion. Underlined HRLs are used as a comparison in Section 3.2.

A Appendix

A.1 Language Selection

For experiments conducted with WikiAnn dataset, we investigate all unseen languages used in (Pfeiffer et al., 2020), which lack resource for task adapters and language adapter, revealed in the bottom of Table 9. To select languages to fusion with, we choose one HRL per each language family, which are bolded in Table 9. For experiment with alternative selection (Section 3.2), we choose languages with most abundant resources, without consideration of diverse language families, which are underlined in Table 9. Note that all languages we deal with have pretrained language adapters available in Adapter-Hub[2]. For the experiment with Amazon Reviews dataset, we consider all languages available, except French, whose language adapter was not provided on Adapter-Hub that fits on XLM-R.

[2] https://adapterhub.ml