MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer

Jonas Pfeiffer[1], **Ivan Vulić**[2], **Iryna Gurevych**[1], **Sebastian Ruder**[3]
[1]Ubiquitous Knowledge Processing Lab, Technical University of Darmstadt
[2]Language Technology Lab, University of Cambridge
[3]DeepMind
pfeiffer@ukp.tu-darmstadt.de

Abstract

The main goal behind state-of-the-art pretrained multilingual models such as multilingual BERT and XLM-R is enabling and bootstrapping NLP applications in *low-resource languages* through zero-shot or few-shot cross-lingual transfer. However, due to limited model capacity, their transfer performance is the weakest exactly on such low-resource languages and languages *unseen* during pretraining. We propose **MAD-X**, an adapter-based framework that enables high portability and parameter-efficient transfer to arbitrary tasks and languages by learning modular language and task representations. In addition, we introduce a novel invertible adapter architecture and a strong baseline method for adapting a pretrained multilingual model to a new language. MAD-X outperforms the state of the art in cross-lingual transfer across a representative set of typologically diverse languages on named entity recognition and causal commonsense reasoning, and achieves competitive results on question answering. Our code and adapters are available at AdapterHub.ml.

1 Introduction

Current deep pretrained multilingual models (Devlin et al., 2019; Conneau and Lample, 2019) achieve state-of-the-art results on cross-lingual transfer, but do not have enough capacity to represent all languages. Evidence for this is the importance of the vocabulary size (Artetxe et al., 2020) and *the curse of multilinguality* (Conneau et al., 2020), a trade-off between language coverage and model capacity. Scaling up a model to cover all of the world's 7,000+ languages is prohibitive. At the same time, limited capacity is an issue even for high-resource languages where state-of-the-art multilingual models underperform their monolingual variants (Eisenschlos et al., 2019; Virtanen et al., 2019; Nozza et al., 2020), and performance decreases further with lower-resource languages covered by the pretrained models. Moreover, the model capacity issue is arguably most severe for languages that were not included in the training data at all, and pretrained models perform poorly on those languages (Ponti et al., 2020b).

In this paper, we propose Multiple ADapters for Cross-lingual transfer (**MAD-X**), a modular framework that leverages a small number of extra parameters to address the fundamental capacity issue that limits pretrained multilingual models. Using a state-of-the-art multilingual model as the foundation, we adapt the model to arbitrary tasks and languages by learning modular language- and task-specific representations *via adapters* (Rebuffi et al., 2017; Houlsby et al., 2019), small bottleneck layers inserted between a model's weights.

In particular, using a recent efficient adapter variant (Pfeiffer et al., 2020a; Rücklé et al., 2020), we train **1)** *language-specific adapter modules* via masked language modelling (MLM) on unlabelled target language data, and **2)** *task-specific adapter modules* via optimising a target task on labelled data in any source language. Task and language adapters are stacked as in Figure 1, allowing us to adapt the pretrained multilingual model also to languages that are not covered in the model's (pre)training data by substituting the target language adapter at inference.

In order to deal with a mismatch between the shared multilingual vocabulary and target language vocabulary, we propose *invertible adapters*, a new type of adapter that is well suited to performing MLM in another language. Our framework goes beyond prior work on using adapters for cross-lingual transfer (Bapna and Firat, 2019; Artetxe et al., 2020) by enabling adaptation to languages *unseen* during pretraining and without learning expensive language-specific token-level embeddings.

We compare MAD-X against state-of-the-art

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 7654–7673,
November 16–20, 2020.

cross-lingual transfer methods on the standard WikiANN NER dataset (Pan et al., 2017; Rahimi et al., 2019) and the XCOPA dataset (Ponti et al., 2020a) for causal commonsense reasoning, relying on a representative set of typologically diverse languages which includes high-resource, low-resource, as well as languages unseen by the pretrained model. MAD-X outperforms the baselines on seen and unseen high-resource and low-resource languages. On the high-resource languages of the challenging XQuAD QA dataset (Artetxe et al., 2020), our framework achieves competitive performance while being more parameter-efficient.

Another contribution of our work is a simple method of adapting a pretrained multilingual model to a new language, which outperforms the standard setting of transferring a model only from labelled source language data.

In sum, our contributions are as follows. **1)** We propose MAD-X, a modular framework that mitigates the curse of multilinguality and adapts a multilingual model to arbitrary tasks and languages. Both code and adapter weights are integrated into the AdapterHub.ml repository (Pfeiffer et al., 2020b).[1] **2)** We propose invertible adapters, a new adapter variant for cross-lingual MLM. **3)** We demonstrate strong performance and robustness of MAD-X across diverse languages and tasks. **4)** We propose a simple and more effective baseline method for adapting a pretrained multilingual model to target languages. **5)** We shed light on the behaviour of current methods on languages that are unseen during multilingual pretraining.

2 Related Work

Cross-lingual Representations Research in modern cross-lingual NLP is increasingly focused on learning general-purpose cross-lingual representations that can be applied to many tasks, first on the word level (Mikolov et al., 2013; Gouws et al., 2015; Glavaš et al., 2019; Ruder et al., 2019; Wang et al., 2020) and later on the full-sentence level (Devlin et al., 2019; Conneau and Lample, 2019; Cao et al., 2020). More recent models such as multilingual BERT (Devlin et al., 2019)—large Transformer (Vaswani et al., 2017) models pretrained on large amounts of multilingual data—have been observed to perform surprisingly well when transferring to other languages (Pires et al., 2019; Wu and Dredze, 2019; Wu et al., 2020) and the current state-of-the-art model, XLM-R is competitive with the performance of monolingual models on the GLUE benchmark (Conneau et al., 2020). Recent studies (Hu et al., 2020), however, indicate that state-of-the-art models such as XLM-R still perform poorly on cross-lingual transfer across many language pairs. The main reason behind such poor performance is the current lack of capacity in the model to represent all languages equally in the vocabulary and representation space (Bapna and Firat, 2019; Artetxe et al., 2020; Conneau et al., 2020).

Adapters Adapter modules have been originally studied in computer vision tasks where they have been restricted to convolutions and used to adapt a model for multiple domains (Rebuffi et al., 2017, 2018). In NLP, adapters have been mainly used for parameter-efficient and quick fine-tuning of a base pretrained Transformer model to new tasks (Houlsby et al., 2019; Stickland and Murray, 2019) and new domains (Bapna and Firat, 2019), avoiding catastrophic forgetting (McCloskey and Cohen, 1989; Santoro et al., 2016). Bapna and Firat (2019) also use adapters to fine-tune and recover performance of a multilingual NMT model on high-resource languages, but their approach cannot be applied to languages that were not seen during pretraining. Artetxe et al. (2020) employ adapters to transfer a pretrained monolingual model to an unseen language but rely on learning new token-level embeddings, which do not scale to a large number of languages. Pfeiffer et al. (2020a) combine the information stored in multiple adapters for more robust transfer learning between monolingual tasks. In their contemporaneous work, Üstün et al. (2020) generate adapter parameters from language embeddings for multilingual dependency parsing.

3 Multilingual Model Adaptation for Cross-lingual Transfer

Standard Transfer Setup The standard way of performing cross-lingual transfer with a state-of-the-art large multilingual model such as multilingual BERT or XLM-R is 1) to fine-tune it on labelled data of a downstream task in a source language and then 2) apply it directly to perform inference in a target language (Hu et al., 2020). A downside of this setting is that the multilingual initialisation balances *many* languages. It is thus not suited to excel at a specific language at inference time. We propose a simple method to ameliorate this issue by allowing the model to additionally

[1] https://github.com/Adapter-Hub/adapter-transformers

adapt to the particular target language.

Target Language Adaptation Similar to fine-tuning monolingual models on the task domain (Howard and Ruder, 2018), we propose to fine-tune a pretrained multilingual model via MLM on unlabelled data of the target language prior to task-specific fine-tuning in the source language. A disadvantage of this approach is that it no longer allows us to evaluate the same model on multiple target languages as it biases the model to a specific target language. However, this approach might be preferable if we only care about performance in a specific (i.e., fixed) target language. We find that target language adaptation results in improved cross-lingual transfer performance over the standard setting (§6). In other words, it does not result in catastrophic forgetting of the multilingual knowledge already available in the pretrained model that enables the model to transfer to other languages. In fact, experimenting with methods that explicitly try to prevent catastrophic forgetting (Wiese et al., 2017) led to worse performance in our experiments.

Nevertheless, the proposed simple adaptation method inherits the fundamental limitation of the pretrained multilingual model and the standard transfer setup: the model's limited capacity hinders effective adaptation to low-resource and unseen languages. In addition, fine-tuning the full model does not scale well to many tasks or languages.

4 Adapters for Cross-lingual Transfer

Our MAD-X framework addresses these deficiencies and can be used to effectively adapt an existing pretrained multilingual model to other languages. The framework comprises three types of adapters: language, task, and invertible adapters. As in previous work (Rebuffi et al., 2017; Houlsby et al., 2019), adapters are trained while keeping the parameters of the pretrained multilingual model fixed. Our framework thus enables learning language and task-specific transformations in a modular and parameter-efficient way. We show the full framework as part of a standard Transformer model in Figure 1 and describe the three adapter types.

4.1 Language Adapters

For learning language-specific transformations, we employ a recent efficient adapter architecture proposed by Pfeiffer et al. (2020a). Following Houlsby et al. (2019) they define the interior of the adapter to be a simple down- and up-projection combined with a residual connection.[2] The language adapter LA_l at layer l consists of a down-projection $\mathbf{D} \in \mathbb{R}^{h \times d}$ where h is the hidden size of the Transformer model and d is the dimension of the adapter, followed by a ReLU activation and an up-projection $\mathbf{U} \in \mathbb{R}^{d \times h}$ at every layer l:

$$\mathsf{LA}_l(\mathbf{h}_l, \mathbf{r}_l) = \mathbf{U}_l(\mathsf{ReLU}(\mathbf{D}_l(\mathbf{h}_l))) + \mathbf{r}_l \quad (1)$$

where $\mathbf{h}_l$ and $\mathbf{r}_l$ are the Transformer hidden state and the residual at layer l, respectively. The residual connection $\mathbf{r}_l$ is the output of the Transformer's feed-forward layer whereas $\mathbf{h}_l$ is the output of the subsequent layer normalisation (see Figure 1).

We train language adapters on unlabelled data of a language using MLM, which encourages them to learn transformations that make the pretrained multilingual model more suitable for a specific language. During task-specific training with labelled data, we use the language adapter of the corresponding source language, which is kept fixed. In order to perform zero-shot transfer to another language, we

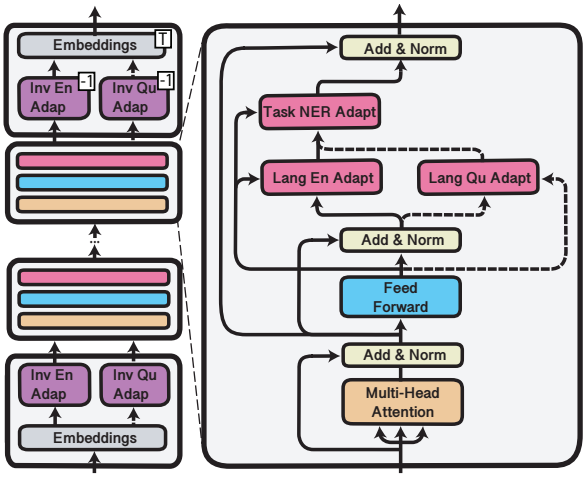
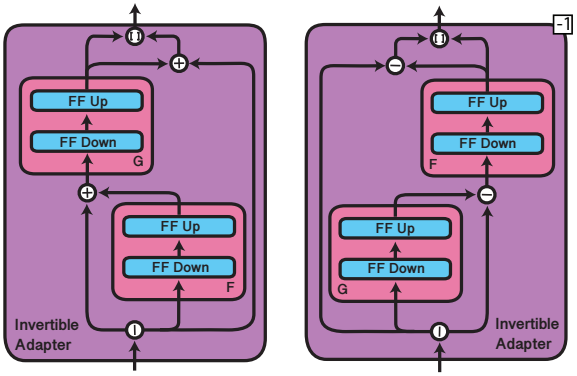


Figure 1: The MAD-X framework inside a Transformer model. Input embeddings are fed into the invertible adapter whose inverse is fed into the tied output embeddings. Language and task adapters are added to each Transformer layer. Language adapters and invertible adapters are trained via masked language modelling (MLM) while the pretrained multilingual model is kept frozen. Task-specific adapters are stacked on top of source language adapters when training on a downstream task such as NER (full lines). During zero-shot cross-lingual transfer, source language adapters are replaced with target language adapters (dashed lines).

[2] Pfeiffer et al. (2020a) perform an extensive hyperparameter search over adapter positions, activation functions, and residual connections within each Transformer layer. They arrive at an architecture variant that performs on par with that of Houlsby et al. (2019), while being more efficient.



(a) The invertible adapter (b) The inversed adapter

Figure 2: The invertible adapter (a) and its inverse (b). The input is split and transformed by projections F and G, which are coupled in an alternating fashion. | indicates the splitting of the input vector, and [ ] indicates the concatenation of two vectors. $+$ and $-$ indicate element-wise addition and subtraction, respectively.

simply replace the source language adapter with its target language component. For instance, as illustrated in Figure 1, we can simply replace a language-specific adapter trained for English with a language-specific adapter trained for Quechua at inference time. This, however, requires that the underlying multilingual model does not change during fine-tuning on the downstream task. In order to ensure this, we additionally introduce task adapters that capture task-specific knowledge.

4.2 Task Adapters

Task adapters TA_l at layer l have the same architecture as language adapters. They similarly consist of a down-projection $\mathbf{D} \in \mathbb{R}^{h \times d}$, a ReLU activation, followed by an up-projection. They are stacked on top of the language adapters and thus receive the output of the language adapter LA_l as input, together with the residual $\mathbf{r}_l$ of the Transformer's feed-forward layer[3]:

$$\mathsf{TA}_l(\mathbf{h}_l, \mathbf{r}_l) = \mathbf{U}_l(\mathsf{ReLU}(\mathbf{D}_l(\mathsf{LA}_l))) + \mathbf{r}_l \quad (2)$$

The output of the task adapter is then passed to another layer normalisation component. Task adapters are the only parameters that are updated when training on a downstream task (e.g., NER) and aim to capture knowledge that is task-specific but generalises across languages.

[3]Initial experiments showed that this residual connection performs better than one to the output of the language adapter.

4.3 Invertible Adapters

The majority of the "parameter budget" of pretrained multilingual models is spent on token embeddings of the shared multilingual vocabulary. Despite this, they underperform on low-resource languages (Artetxe et al., 2020; Conneau et al., 2020), and are bound to fare even worse for languages not covered by the model's training data.

In order to mitigate this mismatch between multilingual and target language vocabulary, we propose invertible adapters. They are stacked on top of the embedding layer while their respective inverses precede the output embedding layer (see Figure 1). As input and output embeddings are tied in multilingual pretrained models, invertibility allows us to leverage the same set of parameters for adapting both input and output representations. This is crucial as the output embeddings, which get discarded during task-specific fine-tuning might otherwise overfit to the pretraining task.

To ensure this invertibility, we employ Nonlinear Independent Component Estimation (NICE; Dinh et al., 2015). NICE enables the invertibility of arbitrary non-linear functions through a set of coupling operations (Dinh et al., 2015). For the invertible adapter, we split the input embedding vector $\mathbf{e}_i$ of the i-th token into two vectors of equal dimensionality $\mathbf{e}_{1,i}, \mathbf{e}_{2,i} \in \mathbb{R}^{h/2}$.[4] For two arbitrary non-linear function F and G, the forward pass through our invertible adapter $A_{inv}()$ is:

$$\begin{aligned} &\mathbf{o}_1 = F(\mathbf{e}_2) + \mathbf{e}_1; \;\; \mathbf{o}_2 = G(\mathbf{o}_1) + \mathbf{e}_2 \\ &\mathbf{o} = [\mathbf{o}_1, \mathbf{o}_2] \end{aligned} \quad (3)$$

where $\mathbf{o}$ is the output of the invertible adapter A_{inv} and $[\cdot, \cdot]$ indicates concatenation of two vectors.

Correspondingly, the inverted pass through the adapter, thus A^{-1}_{inv}, is computed as follows:

$$\begin{aligned} &\mathbf{e}_2 = \mathbf{o}_2 - G(\mathbf{o}_1); \;\; \mathbf{e}_1 = \mathbf{o}_1 - F(\mathbf{e}_2) \\ &\mathbf{e} = [\mathbf{e}_1, \mathbf{e}_2]. \end{aligned} \quad (4)$$

$\mathbf{e}$ is the output of $A^{-1}_{Inv}()$. For the non-linear transformations F and G, we use similar down- and up-projections as for the language and task adapters:

$$\begin{aligned} F(\mathbf{x}) &= \mathbf{U}_F(\mathsf{ReLU}(\mathbf{D}_F(\mathbf{x}))) \\ G(\mathbf{x}) &= \mathbf{U}_G(\mathsf{ReLU}(\mathbf{D}_G(\mathbf{x}))). \end{aligned} \quad (5)$$

where $\mathbf{D}_F, \mathbf{D}_G \in \mathbb{R}^{\frac{h}{4} \times \frac{h}{2}}$ and $\mathbf{U}_F, \mathbf{U}_G \in \mathbb{R}^{\frac{h}{2} \times \frac{h}{4}}$ and $\mathbf{x}$ is a placeholder for $\mathbf{e}_1, \mathbf{e}_2, \mathbf{o}_1$ and $\mathbf{o}_2$. We

[4]For brevity, we further leave out the dependency on i.

illustrate the complete architecture of the invertible adapter and its inverse in Figure 2.

The invertible adapter has a similar function to the language adapter, but aims to capture token-level language-specific transformations. As such, it is trained together with the language adapters using MLM on unlabelled data of a specific language. During task-specific training we use the fixed invertible adapter of the source language, and replace it with the target-language invertible during zero-shot transfer. Importantly, our invertible adapters are much more parameter-efficient compared to the approach of Artetxe et al. (2020), which learns separate token embeddings for every new language.

An Illustrative Example We provide a brief walk-through example from Figure 1. Assuming English (*En*) as the source language and Quechua (*Qu*) as the target language, we first pretrain invertible adapters A^{En}_{Inv} and A^{Qu}_{Inv}, and language adapters A^{En}_{Lang} and A^{Qu}_{Lang} with MLM for which the output of the last layer is passed through $A^{En\ -1}_{Inv}$. We then train a task adapter for the NER task A^{NER}_{Task} on the English NER training set. During training, embeddings are passed through A^{En}_{Inv}. At every layer of the model the data is first passed through the fixed A^{En}_{Lang} and then into the NER adapter A^{NER}_{Task}. For zero-shot inference, the English invertible and language adapters A^{En}_{Inv} and A^{En}_{Lang} are replaced with their Quechua counterparts A^{Qu}_{Inv} and A^{Qu}_{Lang} while the data is still passed through the NER task adapter A^{NER}_{Task}.

5 Experiments

Data We conduct experiments on three tasks: Named entity recognition (NER), question answering (QA), and causal commonsense reasoning (CCR). For NER, we use the WikiANN (Pan et al., 2017) dataset, which was partitioned into train, development, and test portions by Rahimi et al. (2019). For QA, we employ the XQuAD dataset (Artetxe et al., 2020), a cross-lingual version of SQuAD (Rajpurkar et al., 2016). For CCR, we rely on XCOPA (Ponti et al., 2020a), a cross-lingual version of COPA (Roemmele et al., 2011).

Languages The partitioned version of WikiANN covers 176 languages. In order to obtain a comprehensive comparison to state-of-the-art cross-lingual methods under different evaluation conditions, we select languages based on: **a)** variance in data availability (by selecting languages with a range of respective Wikipedia sizes); **b)** their presence in pretrained multilingual models; more precisely, whether data in the particular language was included in the pretraining data of both multilingual BERT and XLM-R or not; and **c)** typological diversity to ensure that different language types and families are covered. In total, we can discern four categories in our language set: **1)** high-resource languages and **2)** low-resource languages covered by the pretrained SOTA multilingual models (i.e., by mBERT and XLM-R); as well as **3)** low-resource languages and **4)** truly low-resource languages not covered by the multilingual models. We select four languages from different language families for each category. We highlight characteristics of the 16 languages from 11 language families in Table 1.

Language	ISO code	Language family	# of Wiki articles	Covered by SOTA?
English	en	Indo-European	6.0M	✓
Japanese	ja	Japonic	1.2M	✓
Chinese	zh	Sino-Tibetan	1.1M	✓
Arabic	ar	Afro-Asiatic	1.0M	✓
Javanese	jv	Austronesian	57k	✓
Swahili	sw	Niger-Congo	56k	✓
Icelandic	is	Indo-European	49k	✓
Burmese	my	Sino-Tibetan	45k	✓
Quechua	qu	Quechua	22k	
Min Dong	cdo	Sino-Tibetan	15k	
Ilokano	ilo	Austronesian	14k	
Mingrelian	xmf	Kartvelian	13k	
Meadow Mari	mhr	Uralic	10k	
Maori	mi	Austronesian	7k	
Turkmen	tk	Turkic	6k	
Guarani	gn	Tupian	4k	

Table 1: Languages in our NER evaluation.

We evaluate on all possible language pairs (i.e., on the Cartesian product), using each language as a source language with every other language (including itself) as a target language. This subsumes both the standard *zero-shot cross-lingual transfer setting* (Hu et al., 2020) as well as the standard *monolingual in-language* setting.

For CCR and QA, we evaluate on the 12 and 11 languages provided in XCOPA and XQuAD respectively, with English as source language. XCOPA contains a typologically diverse selection of languages including two languages (Haitian Creole and Quechua) that are unseen by our main model. XQuAD comprises slightly less typologically diverse languages that are mainly high-resource.

Model	en	ja	zh	ar	jv	sw	is	my	qu	cdo	ilo	xmf	mi	mhr	tk	gn	avg
XLM-R^{Base}	44.2	38.2	40.4	36.4	37.4	42.8	47.1	**26.3**	27.4	18.1	28.8	**35.0**	16.7	**31.7**	20.6	**31.2**	32.6
XLM-R^{Base} MLM-SRC	39.5	45.2	34.7	17.7	34.5	35.3	43.1	20.8	26.6	21.4	28.7	22.4	18.1	25.0	27.6	24.0	29.0
XLM-R^{Base} MLM-TRG	54.8	**47.4**	**54.7**	51.1	38.7	48.1	53.0	20.0	29.3	16.6	27.4	24.7	15.9	26.4	26.5	28.5	35.2
MAD-X^{Base} – LAD – INV	44.5	38.6	40.6	42.8	32.4	43.1	48.6	23.9	22.0	10.6	23.9	27.9	13.2	24.6	18.8	21.9	29.8
MAD-X^{Base} – INV	52.3	46.0	46.2	56.3	**41.6**	48.6	52.4	23.2	32.4	**27.2**	30.8	33.0	**23.5**	29.3	30.4	28.4	37.6
MAD-X^{Base}	**55.0**	46.7	47.3	**58.2**	39.2	**50.4**	**54.5**	24.9	**32.6**	24.2	**33.8**	34.3	16.8	**31.7**	**31.9**	30.4	**38.2**
mBERT	48.6	50.5	50.6	50.9	45.3	48.7	51.2	17.7	31.8	20.7	33.3	26.1	**20.9**	31.3	**34.8**	**30.9**	37.1
MAD-X^{mBERT}	**52.8**	**53.1**	**53.2**	**55.5**	**46.3**	**50.9**	**51.4**	**21.0**	**37.7**	**22.1**	**35.0**	**30.0**	18.6	**31.8**	33.0	25.1	**38.6**
XLM-R^{Large}	47.10	46.52	46.43	45.15	39.21	43.96	48.69	**26.18**	26.39	15.12	22.80	**33.67**	19.86	27.70	29.56	**33.78**	34.6
MAD-X^{Large}	**56.30**	**53.37**	**55.6**	**59.41**	**40.40**	**50.57**	**53.22**	24.55	**33.89**	**26.54**	**30.98**	33.37	**24.31**	**28.03**	**30.82**	26.38	**39.2**

Table 2: NER F1 scores averaged over all 16 target languages when transferring from each source language (i.e. the columns are source languages). The vertical dashed line distinguishes between languages seen in multilingual pretraining and the unseen ones (see also Table 1).

5.1 Baselines

The baseline models are based on different approaches to multilingual model adaptation for cross-lingual transfer, discussed previously in §3.

XLM-R The main model we compare against is XLM-R (Conneau et al., 2020), the current state-of-the-art model for cross-lingual transfer (Hu et al., 2020). It is a Transformer-based model pretrained for 100 languages on large cleaned Common Crawl corpora (Wenzek et al., 2020). For efficiency, we use the XLM-R Base configuration as the basis for most of our experiments. However, we note that the main idea behind the MAD-X framework is not tied to any particular pretrained model, and the framework can be easily adapted to other pretrained multilingual models as we show later in §6 (e.g., multilingual BERT). First, we compare against XLM-R in the standard setting where the entire model is fine-tuned on labelled data of the task in the source language.

XLM-R^{Base} MLM-SRC; XLM-R^{Base} MLM-TRG In §3, we have proposed target language adaptation as a simple method to adapt pretrained multilingual models for better cross-lingual generalisation on the downstream task. As a sanity check, we also compare against adapting to the source language data; we expect it to improve in-language performance but not to help with transfer. In particular, we fine-tune XLM-R with MLM on unlabelled source language (XLM-R^{Base} MLM-SRC) and target language data (XLM-R^{Base} MLM-TRG) prior to task-specific fine-tuning.

5.2 MAD-X: Experimental Setup

For the **MAD-X** framework, unless noted otherwise, we rely on the XLM-R Base architecture; we evaluate the full MAD-X, MAD-X without invertible adapters (–INV), and also MAD-X without language and invertible adapters (– LAD – INV). We use the Transformers library (Wolf et al., 2020) for all our experiments. For fine-tuning via MLM on unlabelled data, we train on the Wikipedia data of the corresponding language for 250,000 steps, with a batch size of 64 and a learning rate of $5e-5$ and $1e-4$ for XLM-R (also for the -SRC and -TRG variants) and adapters, respectively. We train models on NER data for 100 epochs with a batch size of 16 and 8 for high-resource and low-resource languages, respectively, and a learning rate of $5e-5$ and $1e-4$ for XLM-R and adapters, respectively. We choose the best checkpoint for evaluation based on validation performance. Following Pfeiffer et al. (2020a), we learn language adapters, invertible adapters, and task adapters with dimensionalities of 384, 192 (384 for both directions), and 48, respectively. XLM-R Base has a hidden layer size of 768, so these adapter sizes correspond to reductions of 2, 2, and 16.

For NER, we conduct five runs of fine-tuning on the WikiAnn training set of the source language—except for XLM-R^{Base} MLM-TRG for which we conduct one run for efficiency purposes for every source language–target language combination. For QA, we conduct three runs of fine-tuning on the English SQuAD training set, evaluate on all XQuAD target languages, and report mean F_1 and exact match (EM) scores. For CCR, we conduct three runs of fine-tuning on the respective English training set, evaluate on all XCOPA target languages, and report accuracy scores.

6 Results and Discussion

Named Entity Recognition As our main summary of results, we average the cross-lingual transfer results of each method for each source language

across all 16 target languages on the NER dataset. We show the aggregated results in Table 2. Moreover, in the appendix we report the detailed results for all methods across each single language pair, as well as a comparison of methods on the most common setting with English as source language.

In general, we observe that XLM-R performance is indeed lowest for unseen languages (the right half of the table after the vertical dashed line). XLM-R^{Base} MLM-SRC performs worse than XLM-R, which indicates that source-language fine-tuning is not useful for cross-lingual transfer in general.[5] On the other hand, XLM-R^{Base} MLM-TRG is a stronger transfer method than XLM-R on average, yielding gains in 9/16 target languages. However, its gains seem to vanish for low-resource languages. Further, there is another disadvantage, outlined in §3: XLM-R^{Base} MLM-TRG requires fine-tuning the full large pretrained model separately for each target language in consideration, which can be prohibitively expensive.

MAD-X without language and invertible adapters performs on par with XLM-R for almost all languages present in the pretraining data (left half of the table). This mirrors findings in the monolingual setting where task adapters have been observed to achieve performance similar to regular fine-tuning while being more parameter-efficient (Houlsby et al., 2019). However, looking at unseen languages, the performance of MAD-X that only uses task adapters deteriorates significantly compared to XLM-R. This shows that task adapters alone are not expressive enough to bridge the discrepancy when adapting to an unseen language.

Adding language adapters to MAD-X improves its performance across the board, and their usefulness is especially pronounced for low-resource languages. Language adapters help capture the characteristics of the target language and consequently provide boosts for unseen languages. Even for high-resource languages, the addition of language-specific parameters yields substantial improvements. Finally, invertible adapters provide further gains and generally outperform only using task and language adapters: for instance, we observe gains with MAD-X over MAD-X –INV on 13/16 target languages. Overall, the full MAD-X framework improves upon XLM-R by more than 5 F_1 points on average.

[5] However, there are some examples (e.g., JA, TK) where it does yield slight gains over the standard XLM-R transfer.

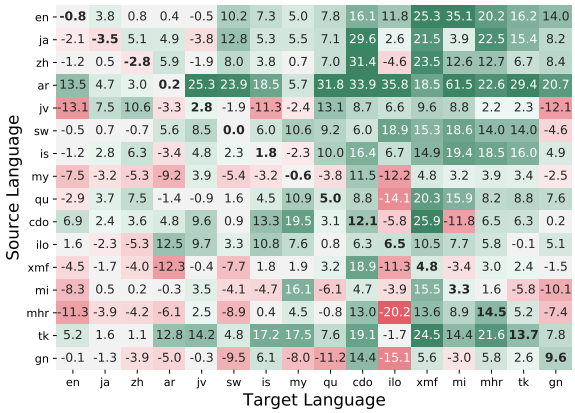


Figure 3: Relative F_1 improvement of MAD-X^{Base} over XLM-R^{Base} in cross-lingual NER transfer.

To demonstrate that our framework is model-agnostic, we also employ two other strong multilingual models, XLM-R^{Large} and mBERT as foundation for MAD-X and show the results in Table 2. MAD-X shows consistent improvements even over stronger base pretrained models.

For a more fine-grained impression of the performance of MAD-X in different languages, we show its relative performance against XLM-R in the standard setting in Figure 3. We observe the largest differences in performance when transferring from high-resource to low-resource and unseen languages (top-right quadrant of Figure 3), which is arguably the most natural setup for cross-lingual transfer. In particular, we observe strong gains when transferring from Arabic, whose script might not be well represented in XLM-R's vocabulary. We also detect strong performance in the in-language monolingual setting (diagonal) for the subset of low-resource languages. This indicates that MAD-X may help bridge the perceived weakness of multilingual versus monolingual models. Finally, MAD-X performs competitively even when the target language is high-resource.[6]

Causal Commonsense Reasoning We show results on transferring from English to each target language on XCOPA in Table 3. For brevity, we only show the results of the best fine-tuning set-

[6] In the appendix, we also plot relative performance of the full MAD-X method (with all three adapter types) versus XLM-R^{Base} MLM-TRG across all language pairs. The scores lead to similar conclusions as before: the largest benefits of MAD-X are observed for the set of low-resource target languages (i.e., the right half of the heatmap). The scores also again confirm that the proposed XLM-R^{Base} MLM-TRG transfer baseline is more competitive than the standard XLM-R transfer across a substantial number of language pairs.

Model	en	et	ht	id	it	qu	sw	ta	th	tr	vi	zh	avg
XLM-R^{Base}	66.8	58.0	51.4	65.0	60.2	51.2	52.0	58.4	62.0	56.6	65.6	**68.8**	59.7
XLM-R^{Base} MLM-TRG	66.8	59.4	50.0	**71.0**	61.6	46.0	**58.8**	60.0	**63.2**	**62.2**	**67.6**	67.4	61.2
MAD-X^{Base}	**68.3**	**61.3**	**53.7**	65.8	**63.0**	**52.5**	56.3	**61.9**	61.8	60.3	66.1	67.6	**61.5**

Table 3: Accuracy scores of all models on the XCOPA test sets when transferring from English. Models are first fine-tuned on SIQA and then on the COPA training set.

	en	ar	de	el	es	hi	ru	th	tr	vi	zh	avg
XLM-R^{Base}	83.6 / 72.1	66.8 / 49.1	**74.4 / 60.1**	73.0 / **55.7**	76.4 / **58.3**	68.2 / 51.7	**74.3 / 58.1**	66.5 / **56.7**	68.3 / 52.8	73.7 / 53.8	51.3 / 42.0	70.6 / 55.5
XLM-R^{Base} MLM-TRG	**84.7 / 72.6**	**67.0 / 49.2**	73.7 / 58.8	**73.2** / 55.7	**76.6 / 58.3**	**69.8 / 53.6**	**74.3** / 57.9	67.0 / 55.8	**68.6 / 53.0**	**75.5 / 54.9**	52.2 / **43.1**	**71.1 / 55.7**
MAD-X^{Base} – INV	83.3 / 72.1	64.0 / 47.1	72.0 / 55.8	71.0 / 52.9	74.6 / 55.5	67.3 / 51.0	72.1 / 55.1	64.1 / 51.8	66.2 / 49.6	73.0 / 53.6	50.9 / 40.6	67.0 / 53.2
MAD-X^{Base}	83.5 / **72.6**	65.5 / 48.2	72.9 / 56.0	72.9 / 54.6	75.9 / 56.9	68.2 / 51.3	73.1 / 56.7	**67.8** / 55.9	67.0 / 49.8	73.7 / 53.3	**52.7** / 42.8	70.3 / 54.4

Table 4: F_1 / EM scores on XQuAD with English as the source language for each target language.

ting from Ponti et al. (2020a)—fine-tuning first on SIQA (Sap et al., 2019) and on the English COPA training set—and report other possible settings in the appendix. Target language adaptation outperforms XLM-R^{Base} while MAD-X^{Base} achieves the best scores. It shows gains in particular for the two unseen languages, Haitian Creole (ht) and Quechua (qu). Performance on the other languages is also generally competitive or better.

Question Answering The results on XQuAD when transferring from English to each target language are provided in Table 4. The main finding is that MAD-X achieves similar performance to the XLM-R baseline. As before, invertible adapters generally improve performance and target language adaptation improves upon the baseline setting. We note that all languages included in XQuAD can be considered high-resource, with more than 100k Wikipedia articles each (cf. Wikipedia sizes of NER languages in Table 1). The corresponding setting can be found in the top-left quadrant in Figure 3 where relative differences are comparable.

These and XCOPA results demonstrate that, while MAD-X excels at transfer to unseen and low-resource languages, it achieves competitive performance even for high-resource languages and on more challenging tasks. These evaluations also hint at the modularity of the adapter-based MAD-X approach, which holds promise of quick adaptation to more tasks: we use exactly the same language-specific adapters in NER, CCR, and QA for languages such as English and Mandarin Chinese that appear in all three evaluation language samples.

7 Further Analysis

Impact of Invertible Adapters We also analyse the relative performance difference of MAD-X

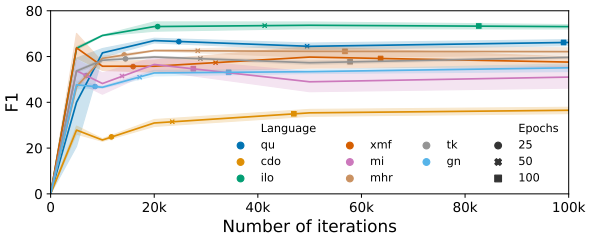


Figure 4: Cross-lingual NER performance of MAD-X transferring from English to the target languages with invertible and language adapters trained on target language data for different numbers of iterations. Shaded regions denote variance in F_1 scores across 5 runs.

Model		+ Params	% Model
MAD-X^{Base}		8.25M	3.05
MAD-X^{Base}	– INV	7.96M	2.94
MAD-X^{Base}	– LAD – INV	0.88M	0.32

Table 5: Number of parameters added to XLM-R Base, and as a fraction of its parameter budget (270M).

with and without invertible adapters for each source language–target language pair on the NER data set (see Section D in the appendix). Invertible adapters improve performance for many transfer pairs, and particularly when transferring to low-resource languages. Performance is only consistently lower with a single low-resource language as source (Maori), likely due to variation in the data.

Sample Efficiency The main adaptation bottleneck of MAD-X is training language adapters and invertible adapters. However, due to the modularity of MAD-X, once trained, these adapters have an advantage of being directly reusable (i.e., "plug-and-playable") across different tasks (see the discussion in §6). To estimate the sample efficiency of adapter training, we measure NER performance on

several low-resource target languages (when transferring from English as the source) conditioned on the number of training iterations. The results are given in Figure 4. They reveal that we can achieve strong performance for the low-resource languages already at 20k training iterations, and longer training offers modest increase in performance.

Moreover, in Table 5 we present the number of parameters added to the original XLM-R Base model per language for each MAD-X variant. The full MAD-X model for NER receives an additional set of 8.25M adapter parameters for every language, which makes up only 3.05% of the original model.

8 Conclusion

We have proposed MAD-X, a general modular framework for transfer across tasks and languages. It leverages a small number of additional parameters to mitigate the capacity issue which fundamentally hinders current multilingual models. MAD-X is model-agnostic and can be adapted to any current pre-trained multilingual model as foundation. We have shown that it is particularly useful for adapting to languages not covered by the multilingual model's training model, while also achieving competitive performance on high-resource languages.

In future work, we will apply MAD-X to other pre-trained models, and employ adapters that are particularly suited for languages with certain properties (e.g. with different scripts). We will also evaluate on additional tasks, and investigate leveraging pre-trained language adapters from related languages for improved transfer to truly low-resource languages with limited monolingual data.

Acknowledgments

Jonas Pfeiffer is supported by the LOEWE initiative (Hesse, Germany) within the emergenCITY center. The work of Ivan Vulić is supported by the ERC Consolidator Grant LEXICAL: Lexical Acquisition Across Languages (no 648909). We thank Laura Rimell for feedback on a draft.

We would like to thank Isabel Pfeiffer for the logo illustrations.

References

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020. On the cross-lingual transferability of monolingual representations. In *Proceedings of the 58th Conference of the Association for Computational Linguistics, ACL 2020, Virtual Conference, July 6-8, 2020*, pages 4623–4637.

Ankur Bapna and Orhan Firat. 2019. Simple, scalable adaptation for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 1538–1548.

Steven Cao, Nikita Kitaev, and Dan Klein. 2020. Multilingual Alignment of Contextual Word Representations. In *8th International Conference on Learning Representations, ICLR 2020, Virtual Conference, April 26 - May 1, 2020*.

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 Conference of the Association for Computational Linguistics, ACL 2020, Virtual Conference, July 6-8, 2020*, pages 8440–8451.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 7057–7067.

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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Laurent Dinh, David Krueger, and Yoshua Bengio. 2015. NICE: non-linear independent components estimation. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Workshop Track Proceedings*.

Julian Eisenschlos, Sebastian Ruder, Piotr Czapla, Marcin Kardas, Sylvain Gugger, and Jeremy Howard. 2019. Multifit: Efficient multi-lingual language model fine-tuning. 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 2019, Hong Kong, China, November 3-7, 2019*, pages 5701–5706.

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 the 57th Conference of the Association*

for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers, pages 710–721.

Stephan Gouws, Yoshua Bengio, and Greg Corrado. 2015. Bilbowa: Fast bilingual distributed representations without word alignments. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, pages 748–756.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzkebski, 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, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, pages 2790–2799.

Jeremy Howard and Sebastian Ruder. 2018. Universal Language Model Fine-tuning for Text Classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 328–339.

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 the 37th International Conference on Machine Learning, ICML 2020, 12-18 July 2020, Virtual Conference*.

Michael McCloskey and Neal J. Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of Learning and Motivation*, volume 24, pages 109–165.

Tomas Mikolov, Quoc V. Le, and Ilya Sutskever. 2013. Exploiting similarities among languages for machine translation. *arXiv preprint*.

Debora Nozza, Federico Bianchi, and Dirk Hovy. 2020. What the [mask]? making sense of language-specific BERT models. *arXiv preprint*.

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, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1946–1958.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. 2020a. AdapterFusion: Non-destructive task composition for transfer learning. *arXiv preprint*.

Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. 2020b. Adapterhub: A framework for adapting transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (System Demonstrations), EMNLP 2020, Virtual Conference, 2020*.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual bert? In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4996–5001.

Edoardo Maria Ponti, Goran Glavaš, Olga Majewska, Qianchu Liu, Ivan Vulić, and Anna Korhonen. 2020a. XCOPA: A Multilingual Dataset for Causal Commonsense Reasoning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Virtual Conference*.

Edoardo Maria Ponti, Ivan Vulić, Ryan Cotterell, Marinela Parovic, Roi Reichart, and Anna Korhonen. 2020b. Parameter space factorization for zero-shot learning across tasks and languages. *Transactions of the Association for Computational Linguistics 2020*.

Afshin Rahimi, Yuan Li, and Trevor Cohn. 2019. Massively multilingual transfer for NER. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 151–164.

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, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 2383–2392.

Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. 2017. Learning multiple visual domains with residual adapters. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 506–516.

Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. 2018. Efficient parametrization of multi-domain deep neural networks. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 8119–8127.

Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S. Gordon. 2011. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *Logical Formalizations of Commonsense Reasoning, Papers from the 2011 AAAI Spring Symposium, Technical Report SS-11-06, Stanford, California, USA, March 21-23, 2011*.

Andreas Rücklé, Gregor Geigle, Max Glockner, Tilman Beck, Jonas Pfeiffer, Nils Reimers, and Iryna

Gurevych. 2020. AdapterDrop: On the Efficiency of Adapters in Transformers. *arXiv preprint*.

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

Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy P. Lillicrap. 2016. One-shot learning with memory-augmented neural networks. *arXiv preprint*.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan LeBras, and Yejin Choi. 2019. Socialiqa: Commonsense reasoning about social interactions. *arXiv preprint*.

Asa Cooper Stickland and Iain Murray. 2019. BERT and PALs: Projected Attention Layers for Efficient Adaptation in Multi-Task Learning. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, pages 5986–5995.

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 2020, Virtual Conference*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 5998–6008.

Antti Virtanen, Jenna Kanerva, Rami Ilo, Jouni Luoma, Juhani Luotolahti, Tapio Salakoski, Filip Ginter, and Sampo Pyysalo. 2019. Multilingual is not enough: BERT for Finnish. *arXiv preprint*.

Zirui Wang, Jiateng Xie, Ruochen Xu, Yiming Yang, Graham Neubig, and Jaime Carbonell. 2020. Cross-lingual Alignment vs Joint Training: A Comparative Study and A Simple Unified Framework. In *8th International Conference on Learning Representations, ICLR 2020, Virtual Conference, April 26 - May 1, 2020*.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. 2020. Ccnet: Extracting high quality monolingual datasets from web crawl data. In *Proceedings of The 12th Language Resources and Evaluation Conference, LREC 2020, Marseille, France, May 11-16, 2020*, pages 4003–4012.

Georg Wiese, Dirk Weissenborn, and Mariana L. Neves. 2017. Neural Domain Adaptation for Biomedical Question Answering. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017), Vancouver, Canada, August 3-4, 2017*, pages 281–289.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi andArt Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2020. HuggingFace's Transformers: State-of-the-art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (System Demonstrations), EMNLP 2020, Virtual Conference, 2020*.

Shijie Wu, Alexis Conneau, Haoran Li, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Emerging cross-lingual structure in pretrained language models. In *Proceedings of the 58th Conference of the Association for Computational Linguistics, ACL 2020, Virtual Conference, July 6-8, 2020*, pages 6022–6034.

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 2019, Hong Kong, China, November 3-7, 2019*, pages 833–844.

A Evaluation data

- Named Entity Recognition (NER). Data: WikiANN (Rahimi et al., 2019). Available online at:
  `www.amazon.com/clouddrive/share/d3KGCRCIYwhKJF0H3eWA26hjg2ZCRhjpEQtDL70FSBN`.
- Causal Commonsense Reasoning (CCR). Data: XCOPA (Ponti et al., 2020a). Available online at:
  `github.com/cambridgeltl/xcopa`
- Question Answering (QA). Data: XQuAD (Artetxe et al., 2020). Available online at:
  `github.com/deepmind/xquad`

B NER zero-shot results from English

We show the F1 scores when transferring from English to the other languages averaged over five runs in Table 6.

C NER results per language pair

We show the F1 scores on the NER dataset across all combinations of source and target language for all of our comparison methods in Figures 5 (XLM-R^{Base}), 6 (XLM-R^{Base} MLM-SRC), 7 (XLM-R^{Base} MLM-TRG), 8 (MAD-X^{Base} –

	en	ja	zh	ar	jv	sw	is	my	qu	cdo	ilo	xmf	mi	mhr	tk	gn	avg
mBERT	84.8	**26.7**	**38.5**	38.7	**57.8**	66.0	65.7	42.9	54.9	14.20	63.5	31.1	21.8	46.0	47.2	45.4	44.0
XLM-R	83.0	15.2	19.6	41.3	56.1	63.5	67.2	46.9	58.3	20.47	61.3	32.2	15.9	41.8	43.4	41.0	41.6
XLM-R^{Base} MLM-SRC	84.2	8.45	11.0	27.3	44.8	57.9	59.0	35.6	52.5	21.4	60.3	22.7	22.7	38.1	44.0	41.7	36.5
XLM-R^{Base} MLM-TRG	84.2	9.30	15.5	**44.5**	50.2	**77.7**	71.7	**55.5**	**68.7**	**47.6**	**84.7**	**60.3**	43.6	56.3	56.4	50.6	52.8
MAD-X – LAD – inv	82.0	15.6	20.3	41.0	54.4	66.4	67.8	48.8	57.8	16.9	59.9	36.9	14.3	44.3	41.9	42.9	41.9
MAD-X – INV	82.2	16.8	20.7	36.9	54.1	68.7	71.5	50.0	59.6	39.2	69.9	54.9	48.3	58.1	53.1	52.8	50.3
MAD-X	82.3	19.0	20.5	41.8	55.7	73.8	**74.5**	51.9	66.1	36.5	73.1	57.6	**51.0**	**62.1**	**59.7**	**55.1**	**53.2**

Table 6: NER F1 scores for zero-shot transfer from English.

LAD – INV), 9 (MAD-X^{Base} – INV), 10 (MAD-X^{Base}), 11 (mBERT), 12 (MAD-X^{mBERT}) , 13 (XLM-R^{Large}), and 14 (MAD-X^{mBERT}). Each score is averaged over five runs.

D Relative improvement of MAD-X over baselines in cross-lingual NER transfer

We show the heatmaps which depict relative F1 improvements of the full MAD-X^{Base} framework in the cross-lingual NER transfer task over: (a) the baseline model XLM-R^{Base} MLM-TRG (Figure 15) and (b) the MAD-X^{Base} variant without invertible adapters: MAD-X^{Base} –INV (Figure 16).

The heatmap which depicts relative F1 improvements of the full MAD-X^{mBERT} framework over mBERT can be found in Figure 17.

Finally, the heatmap which depicts relative F1 improvements of the full MAD-X^{Large} framework over XLM-R^{Large} can be found in Figure 18.

E XCOPA results for all settings

We show the results on XCOPA for all fine-tuning settings in Table 7.

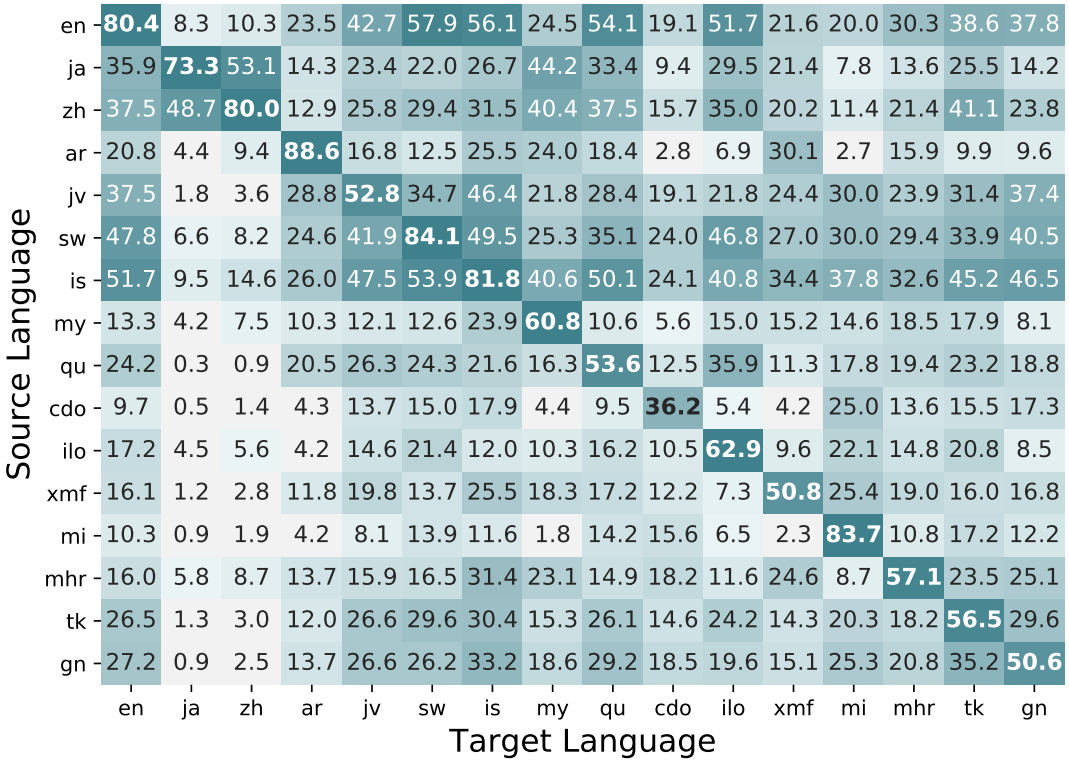


Figure 5: Mean F1 scores of XLM-R^{Base} in the standard setting (XLM-R^{Base}) for cross-lingual transfer on NER.

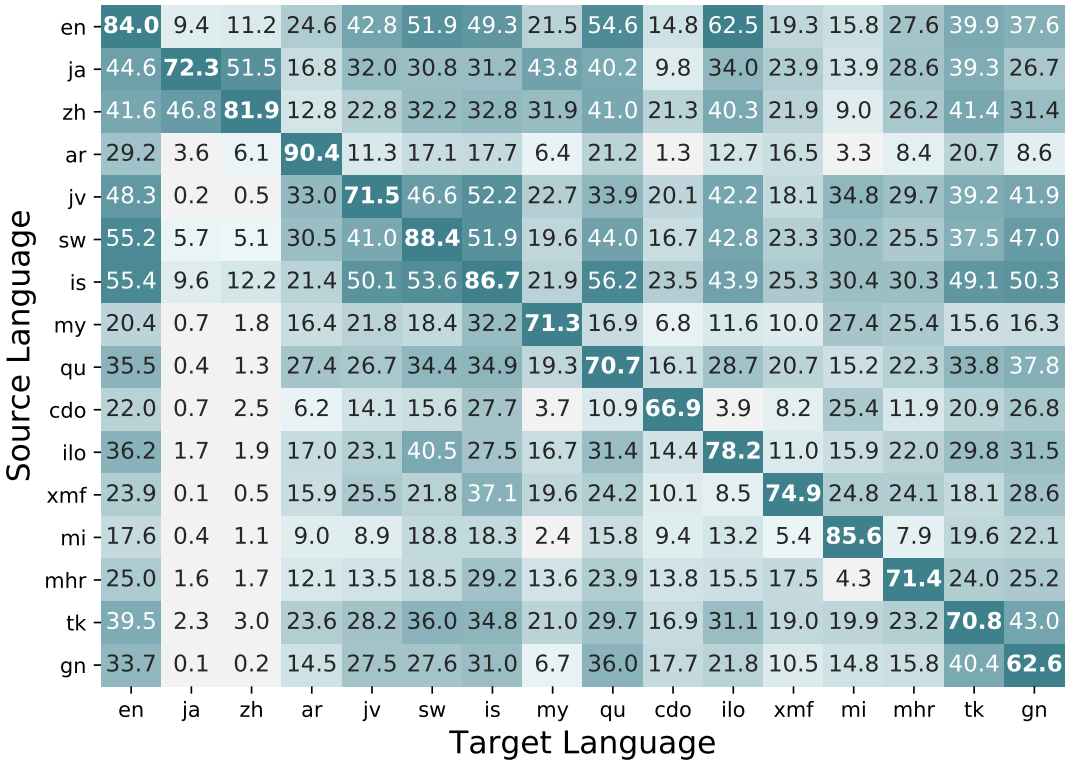


Figure 6: Mean F1 scores of XLM-R^{Base} with MLM fine-tuning on source language data (XLM-R^{Base} MLM-SRC) for cross-lingual transfer on NER.

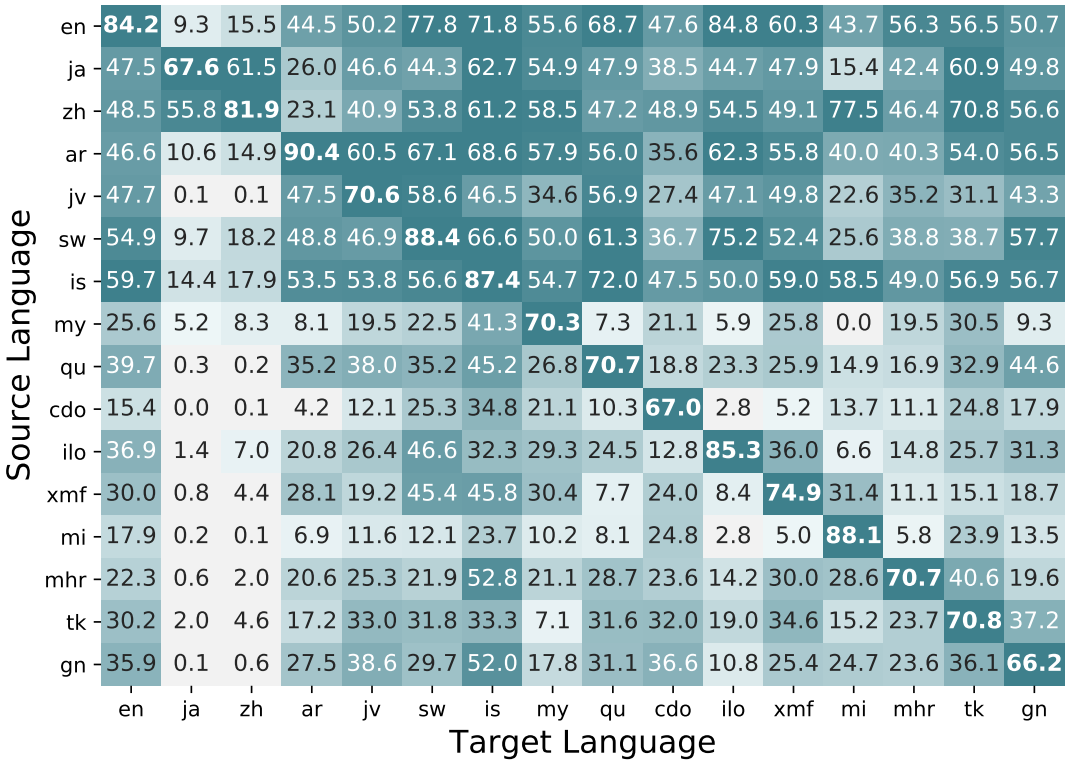


Figure 7: Mean F1 scores of XLM-R^{Base} with MLM fine-tuning on target language data (XLM-R^{Base} MLM-TRG) for cross-lingual transfer on NER.

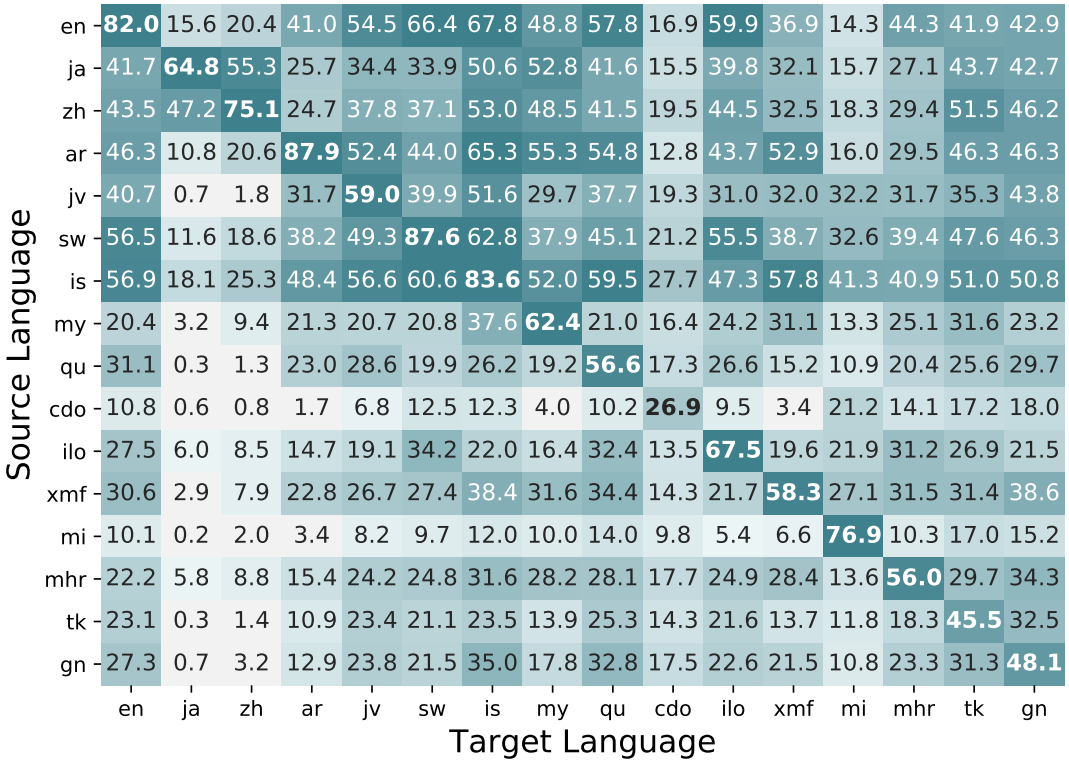


Figure 8: Mean F1 scores of our framework without language adapters and invertible adapters (MAD-X^{Base} – LAD – INV) for cross-lingual transfer on NER.

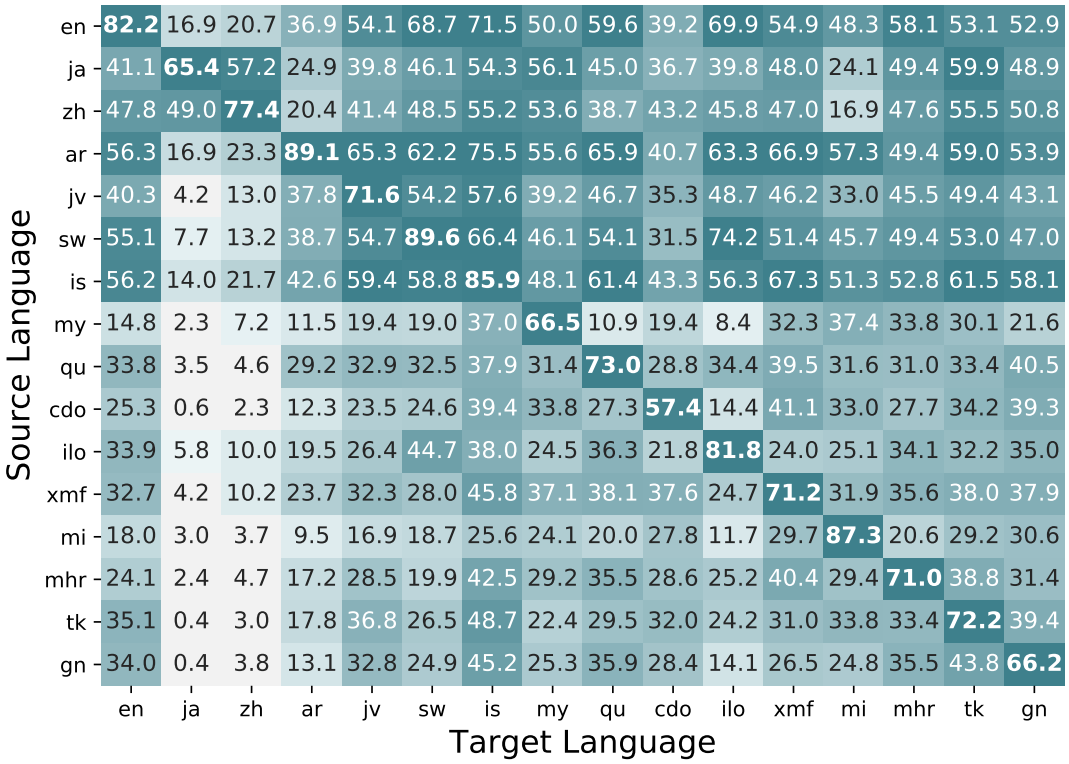


Figure 9: Mean F1 scores of our framework without invertible adapters (MAD-X^{Base} – INV) for cross-lingual transfer on NER.

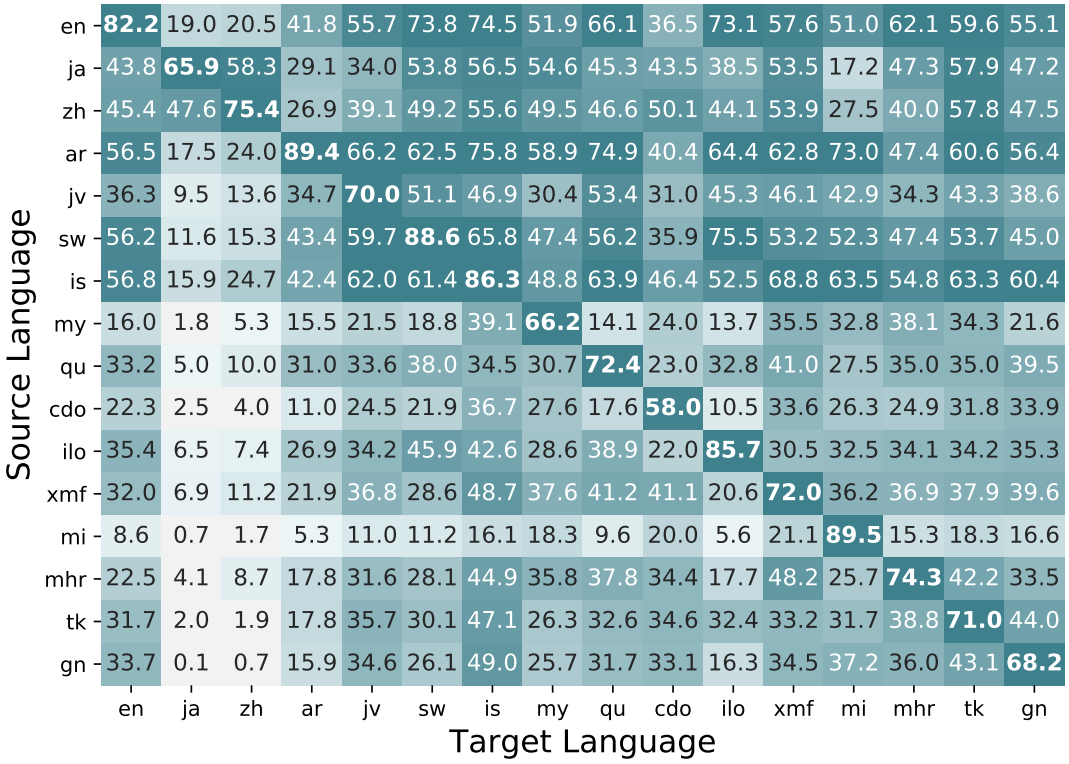


Figure 10: Mean F1 scores of our complete adapter-based framework (MAD-X^{Base}) for cross-lingual transfer on NER.

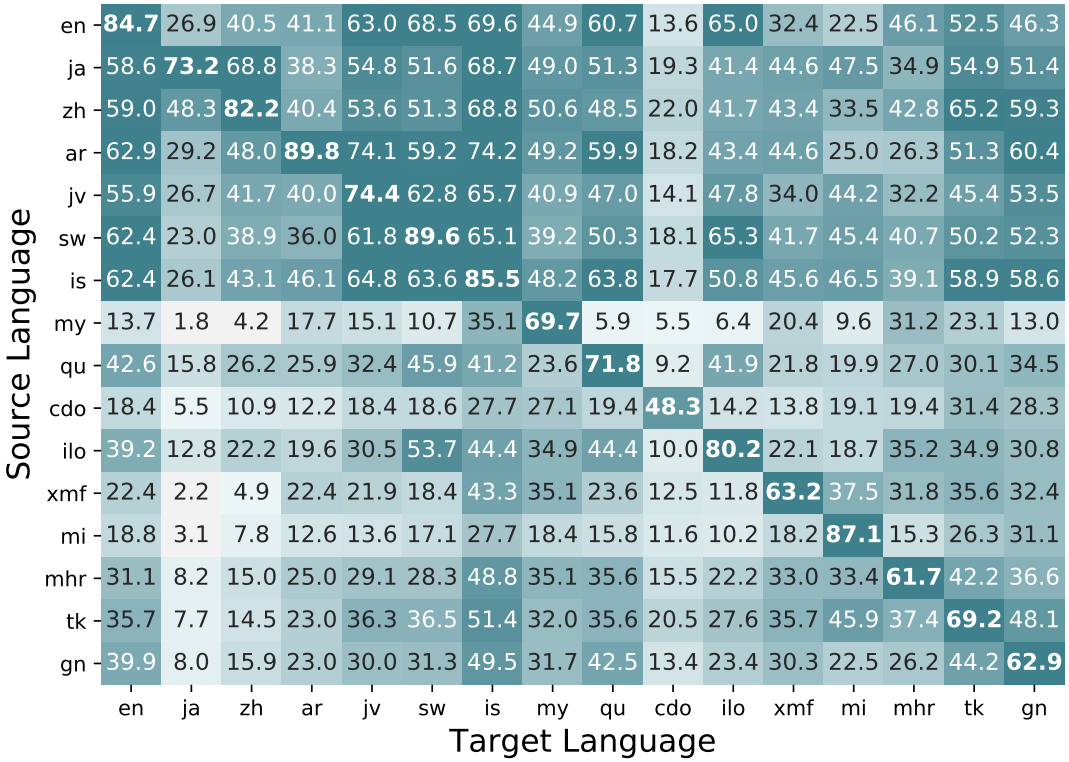


Figure 11: Mean F1 scores of mBERT for cross-lingual transfer on NER.

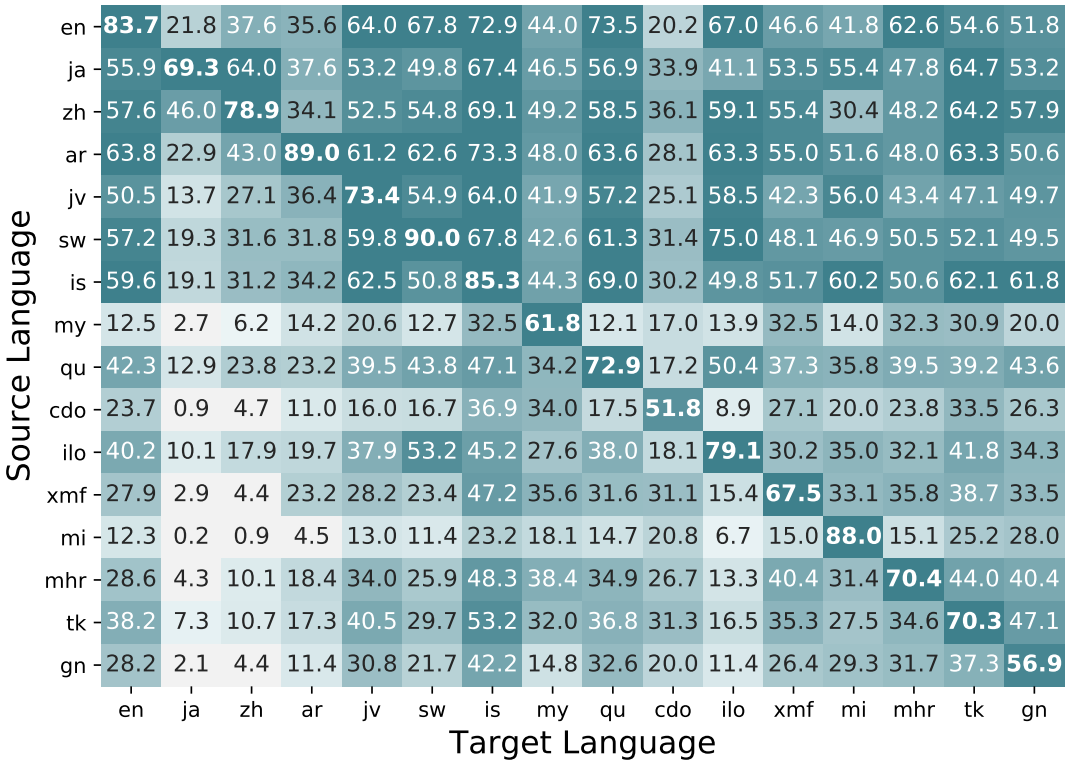


Figure 12: Mean F1 scores of our complete adapter-based framework (MAD-X^{mBERT}) for cross-lingual transfer on NER.

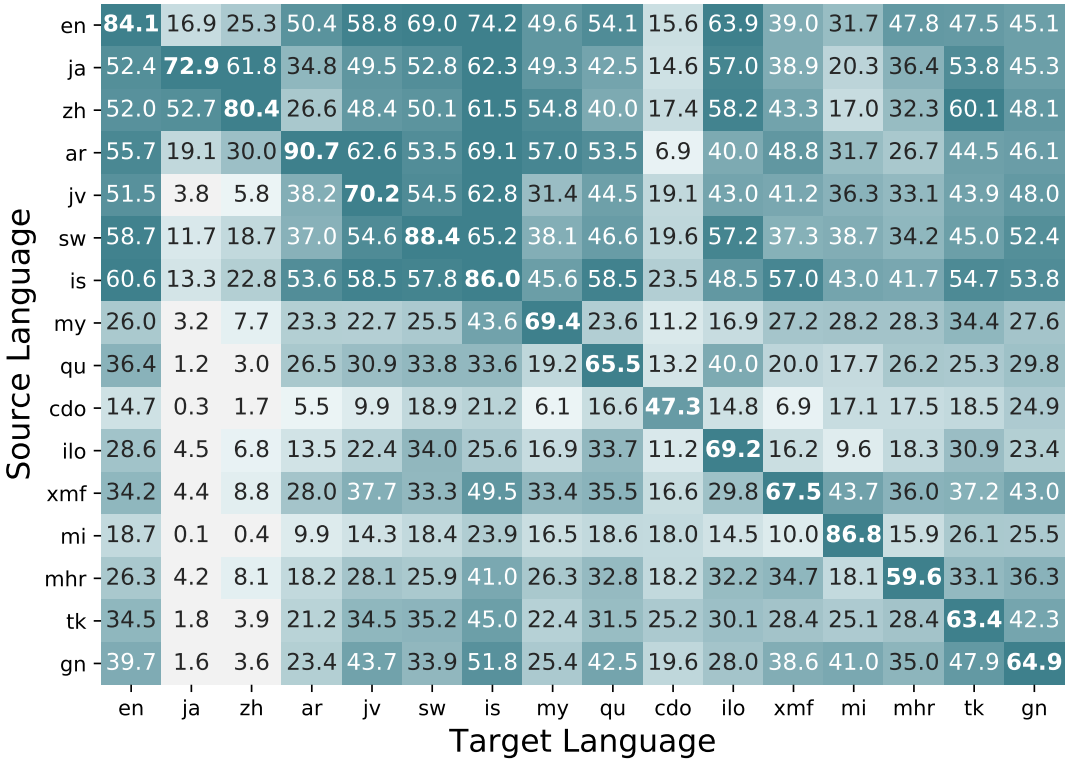


Figure 13: Mean F1 scores of XLM-R^{Large} for cross-lingual transfer on NER.

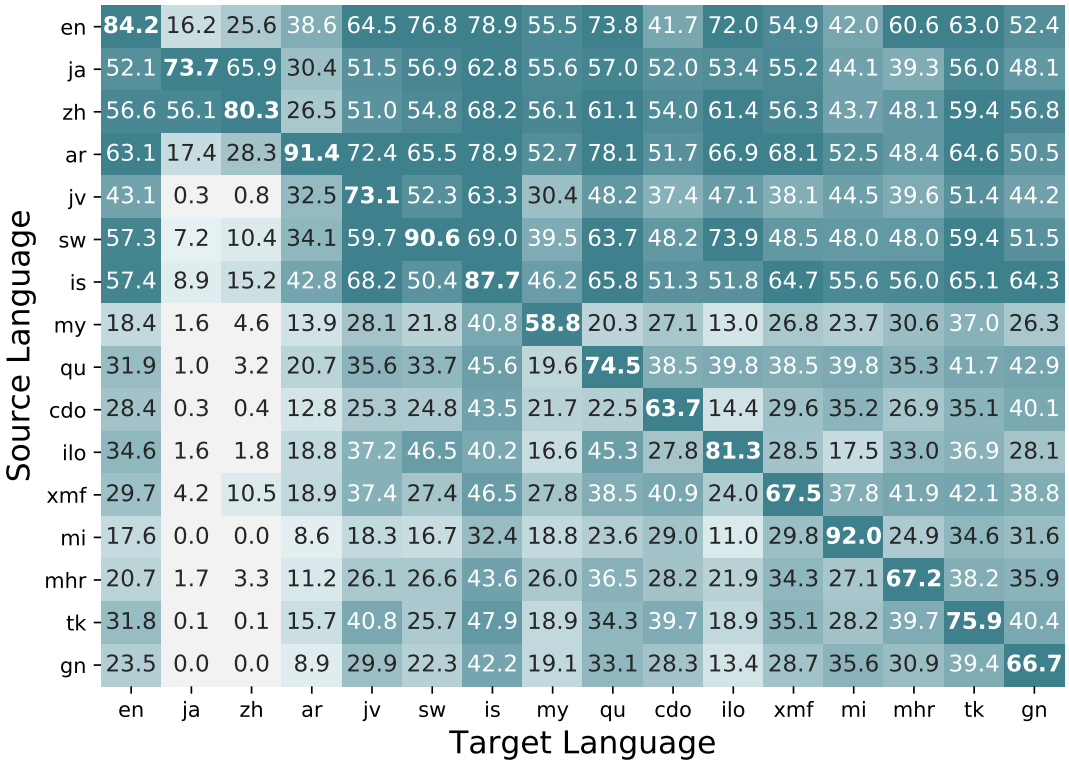


Figure 14: Mean F1 scores of our complete adapter-based framework (MAD-X^{Large}) for cross-lingual transfer on NER.

Model	en	et	ht	id	it	qu	sw	ta	th	tr	vi	zh	avg
XLM-R$^{Base}_{\rightarrow COPA}$	57.6	59.8	49.4	58.0	56.0	50.7	57.2	56.6	52.8	56.2	58.5	56.6	55.8
XLM-R^{Base} MLM-TRG$_{\rightarrow COPA}$	57.6	57.8	48.6	60.8	54.4	49.5	55.4	55.8	54.2	54.8	57.6	57.2	55.3
XLM-R$^{Base}_{\rightarrow SIQA}$	68.0	59.4	49.2	67.2	63.6	51.0	57.6	58.8	61.6	60.4	65.8	66.0	60.7
XLM-R$^{Base}_{\rightarrow SIQA \rightarrow COPA}$	66.8	58.0	51.4	65.0	60.2	51.2	52.0	58.4	62.0	56.6	65.6	**68.8**	59.7
XLM-R^{Base} MLM-TRG$_{\rightarrow SIQA \rightarrow COPA}$	66.8	59.4	50.0	**71.0**	61.6	46.0	**58.8**	60.0	**63.2**	**62.2**	**67.6**	67.4	61.2
MAD-X$^{Base}_{\rightarrow COPA}$	48.1	49.0	51.5	50.7	50.7	49.1	52.7	52.5	48.7	53.3	52.1	50.4	50.7
MAD-X$^{Base}_{\rightarrow SIQA}$	67.6	59.7	51.7	66.2	**64.4**	**54.0**	53.9	61.3	61.1	60.1	65.4	66.7	61.0
MAD-X$^{Base}_{\rightarrow SIQA \rightarrow COPA}$	**68.3**	**61.3**	**53.7**	65.8	63.0	52.5	56.3	**61.9**	61.8	60.3	66.1	67.6	**61.5**

Table 7: Accuracy scores of all models on the XCOPA test sets when transferring from English. Models are either only fine-tuned on the COPA training set ($\rightarrow COPA$), only fine-tuned on the SIQA training set ($\rightarrow SIQA$) or fine-tuned first on SIQA and then on COPA ($\rightarrow SIQA \rightarrow COPA$).

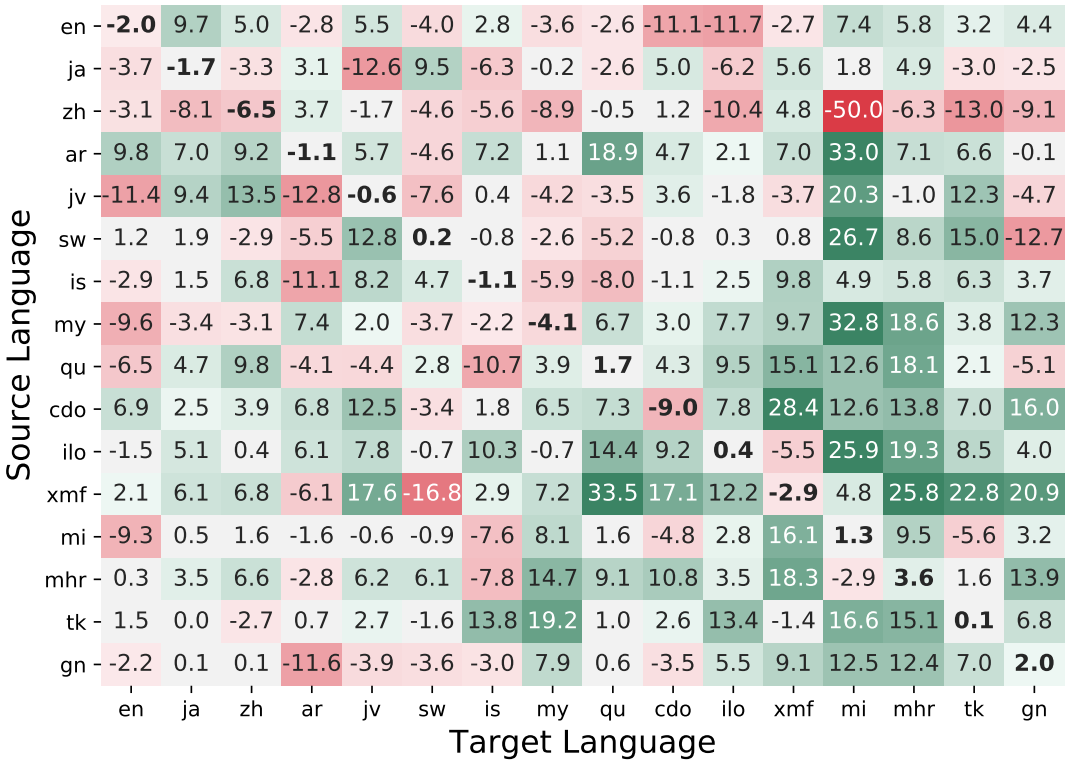


Figure 15: Relative F_1 improvement of MAD-X^{Base} over XLM-R^{Base} MLM-TRG in cross-lingual NER transfer.

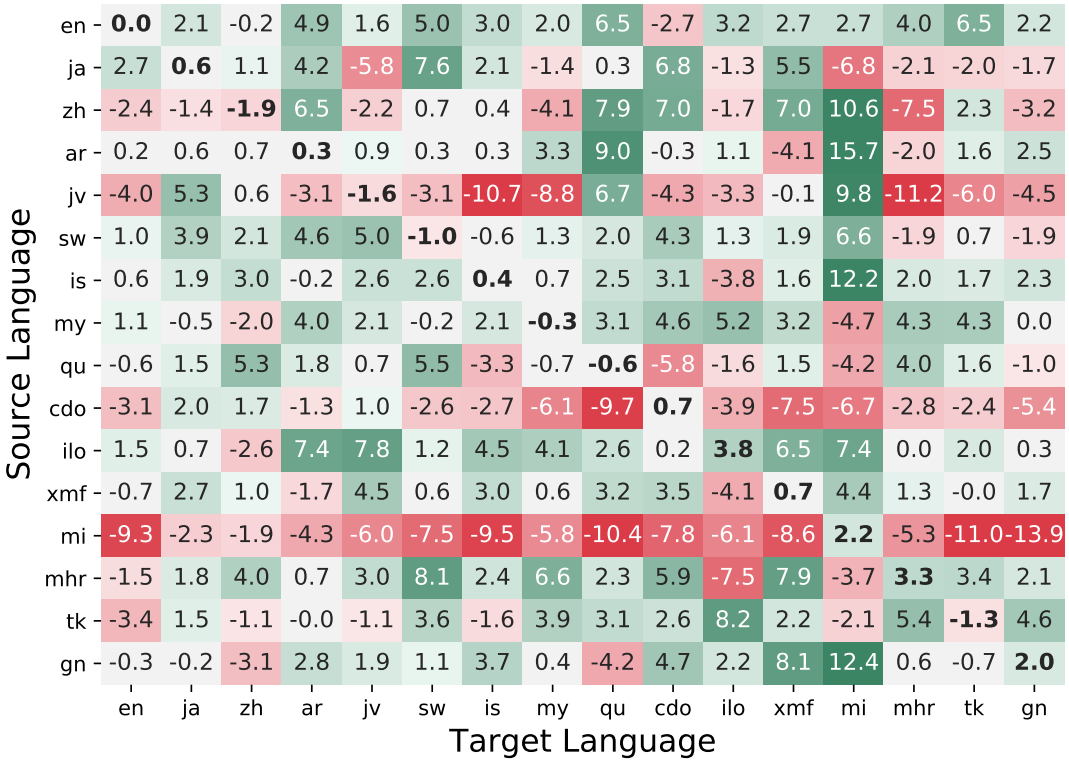


Figure 16: Relative F_1 improvement of MAD-X^{Base} over MAD-X^{Base} –INV in cross-lingual NER transfer.

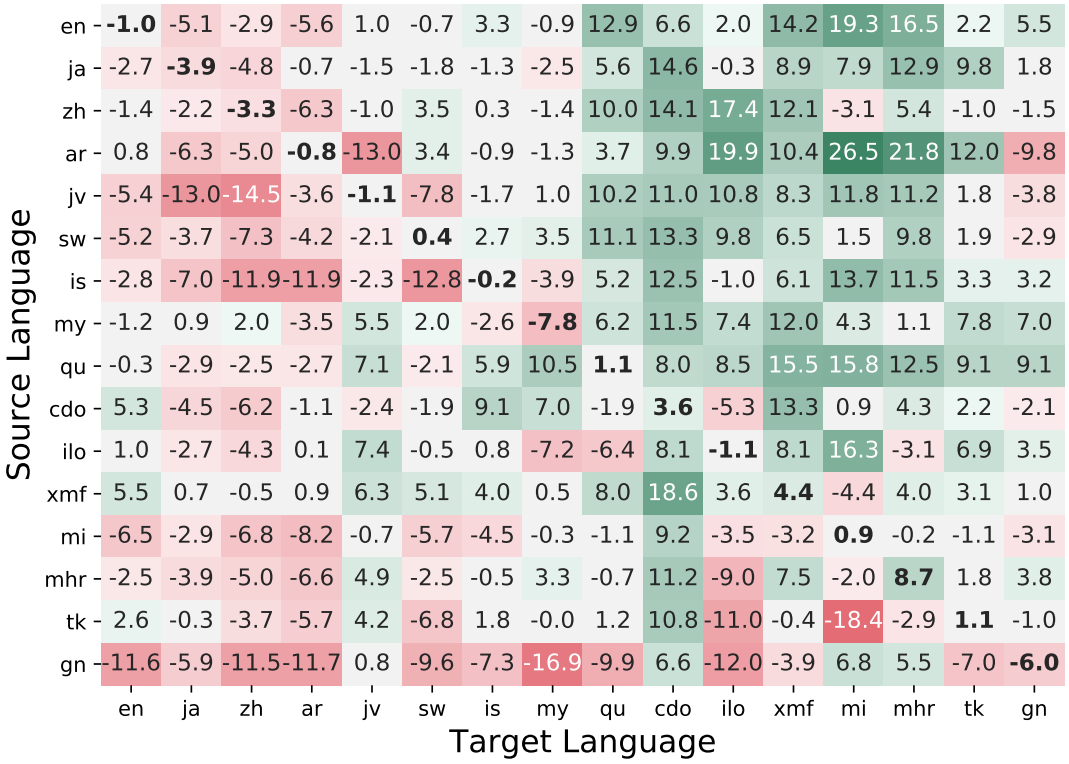


Figure 17: Relative F_1 improvement of MAD-X^{mBERT} over mBERT in cross-lingual NER transfer.

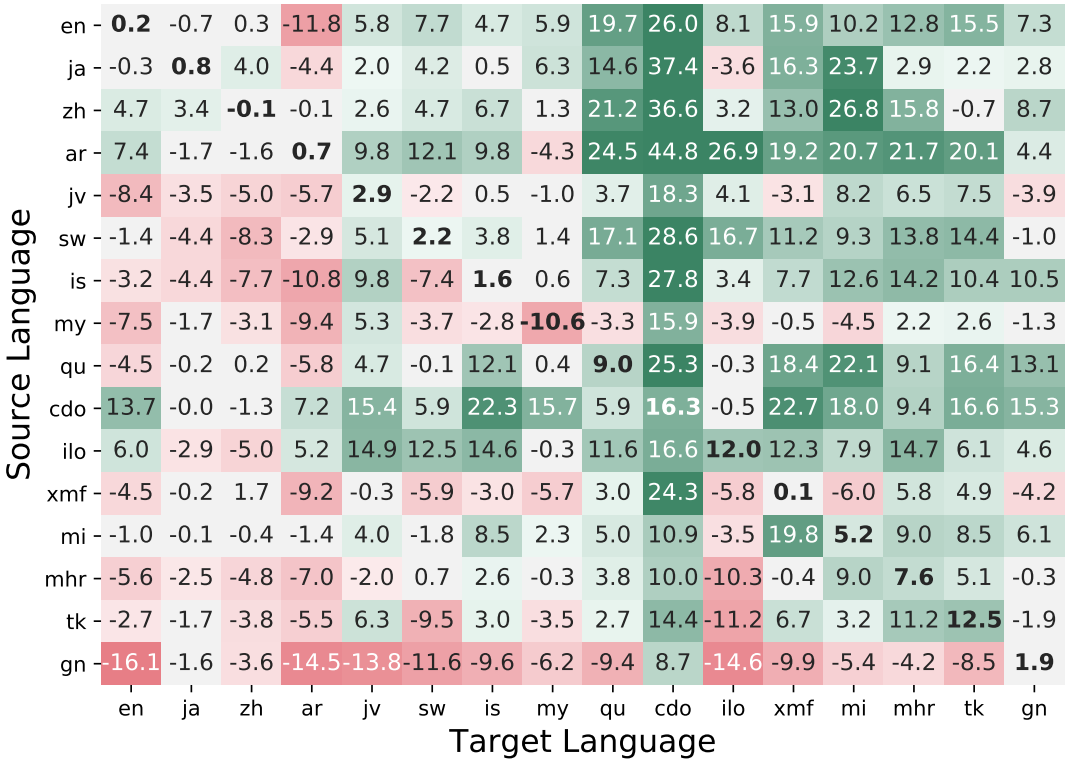


Figure 18: Relative F_1 improvement of MAD-X^{Large} over XLM-R^{Large} in cross-lingual NER transfer.