m4Adapter: Multilingual Multi-Domain Adaptation for Machine Translation with a Meta-Adapter

Wen Lai[1], **Alexandra Chronopoulou**[1,2] and **Alexander Fraser**[1,2]

[1] Center for Information and Language Processing, LMU Munich, Germany

[2] Munich Center for Machine Learning, Germany

{lavine, achron, fraser}@cis.lmu.de

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Introduction

- We consider two problems of MNMT adaptation: **Domain Adaptation** and **Language Adaptation**.
- Common practice tends to *fine-tune* or use lightweight *adapters* to adapt the MNMT model to a new domain/language pair, which considers the two problems separately.
- We consider a very challenging scenario: adapting the MNMT model both to a new domain and to a new language pair at the same time.
- To this end, we propose $m^4Adapter$ (Multilingual Multi-Domain Adaptation for Machine Translation with a Meta-Adapter).
- An ablation study shows that our approach more effectively transfers domain knowledge across different languages and language information across different domains.

Method

- We propose a 2-step approach:
  - **Meta-Training**: we perform meta-learning with adapters to efficiently learn parameters in a shared representation space across multiple tasks using a small amount of training data (5000 samples);
  - **Meta-Adaptation**: we fine-tune the trained model to a new domain and language pair simultaneously using an even smaller dataset (500 samples).

Task Definition

- We address multilingual multi-domain translation as a multi-task learning problem. Specifically, a translation task in a specific textual domain corresponds to a Domain-Language-Pair (**DLP**). For example, an English-Serbian translation task in the 'Ubuntu' domain is denoted as a DLP 'Ubuntu-en-sr'.

Task Sampling

- Given d domains and l languages, we sample some DLPs per batch among all $d \cdot l \cdot (l-1)$ tasks.
- We follow a temperature-based heuristic sampling strategy (aharoni et al., 2019), which defines the probability of any dataset as a function of its size.

Meta-Learning Algorithm

- We follow *Reptile* (nichol et al., 2018), an alternative first-order meta-learning algorithm that uses a simple update rule:

$$\psi \leftarrow \psi + \beta \frac{1}{|\{\mathcal{T}_i\}|} \sum_{\mathcal{T}_i \sim \mathcal{M}} (\psi_i^{(k)} - \psi)$$

- Where $\psi_i^{(k)}$ is $U_i^k(\theta, \psi)$ and β is a hyper-parameter.

Meta-Adapter

- We propose *Meta-Adapter*, which inserts adapter layers into the meta-learning training process. Different from the traditional adapter training process, we only need to train a single meta-adapter to adapt to all new language pairs and domains.

Meta-Adaptation

- After the meta-training phase, the parameters of the adapter are fine-tuned to adapt to new tasks using a small amount of data to simulate a low-resource scenario.

Algorithm 1 $m^4Adapter$

Input: $\mathcal{D}_{train}$ set of DLPs for meta training; Pre-trained MNMT model θ

```
Initialize P_D(i) based on temperature sampling
while not converged do
    ▷ Perform Reptile Updates
    Sample m DLPs T_1, T_2, ..., T_m from M
    for i = 1,2,...,m do
        ψ_i^(k) ← U_i^k(θ, ψ), denoting k gradient
        updates from ψ on batches of DLP T_i
        while keeping θ frozen
    end for
    ψ ← ψ + (β/m) Σ_{i=1}^m (ψ_i^(k) − ψ)
end while
return Meta-Adapter parameter ψ
```

Main Results (Meta-Training)

- Motivated by (Lai et al., 2022), we compare our approach to multiple baselines in terms of domain robustness.

	BLEU	specific domain		
		TED	Ubuntu	KDE
m2m	18.18	16.20	20.61	22.04
m2m + FT	20.84	17.53	28.81	29.19
m2m + tag	22.70	18.70	**31.86**	31.53
agnostic-adapter	23.70	**19.82**	31.07	32.74
stack-adapter	21.06	18.34	29.17	30.26
meta-learning	20.01	17.57	28.11	28.59
$m^4Adapter$	**23.89**	19.77	31.46	**32.91**

Table 1: Performance on *meta-training* stage

Efficiency

- We compare the efficiency of baselines to traditional fine-tuning and list their number of trainable parameters and training/adapting time in the Table 2.

Method	#Param.	$Time_T$	$Time_A$
m2m	418M (100%)	-	-
m2m + FT	418M (100%)	100%	100%
m2m + tag	418M (100%)	100%	100%
agnostic-adapter	3.17M (0.75%)	42%	150%
stack-adapter	$k\cdot$ 3.17M ($k\cdot$ 0.75%)	$k\cdot$ 42%	200%
meta-learning	418M (100%)	75%	500%
$m^4Adapter$	3.17M (0.75%)	34%	300%

Table 2: Efficiency of $m^4Adapter$

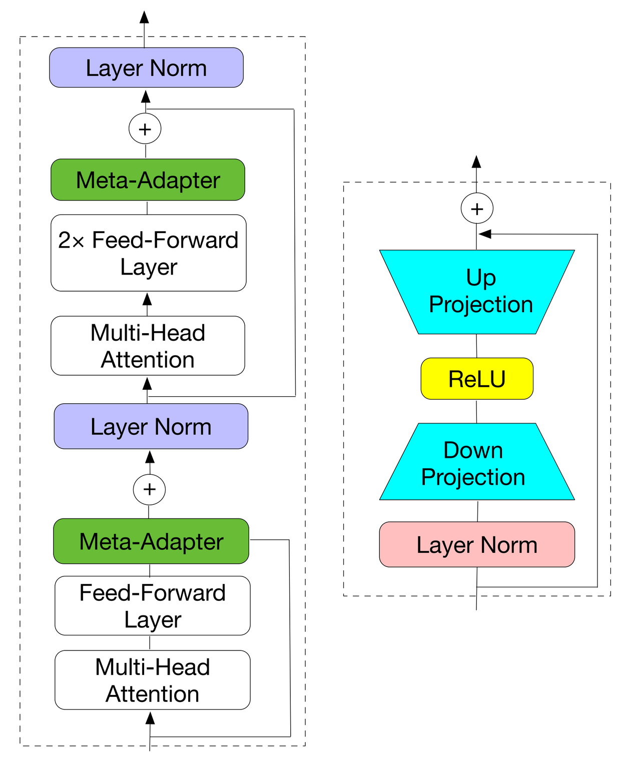

Figure 1: Architecture of $m^4Adapter$

Main Results (Meta-Adaptation)

	DLP (meta-adaptation domain)			specific DLP					
	UN	Tanzil	Infopankki	UN-ar-en	Tanzil-ar-en	Infopankki-ar-en	UN-ar-ru	Tanzil-ar-ru	Infopankki-ar-ru
m2m	32.28	8.72	17.40	38.94	6.44	22.57	22.96	3.64	15.05
m2m + FT	29.93	8.26	15.88	35.11	6.85	21.33	19.10	3.05	14.19
m2m + tag	29.88	8.06	15.93	34.39	6.63	20.12	19.37	2.65	13.68
agnostic-adapter	30.56	8.42	17.36	36.13	6.12	23.08	20.64	3.63	14.96
stack-adapter	29.64	8.14	17.19	35.31	5.83	22.14	19.17	2.34	13.85
meta-learning	32.21	7.02	16.73	37.13	5.50	18.91	22.68	1.70	15.23
$m^4Adapter$	**33.53**	**9.87**	**18.43**	**39.05**	**8.56**	**23.21**	**25.22**	**4.33**	**17.48**
Δ	+1.25	+1.15	+1.03	+0.11	+2.12	+0.64	+2.26	+0.69	+2.43

Table 3: Main results on meta-adaptation stage

Ablation Study

	meta-adaptation domain							specific DLP (hr-sr)				
	EUbookshop	KDE	OpenSubtitles	QED	TED	Ubuntu	Bible	EUbookshop	KDE	OpenSubtitles	QED	TED
m2m	17.77	22.05	14.13	18.34	16.20	20.62	9.80	11.43	25.37	19.01	12.25	8.14
m2m + FT	12.73	24.56	16.22	20.46	18.74	31.32	11.30	9.79	21.05	53.34	23.87	20.81
m2m + tag	13.03	25.34	16.12	17.75	17.04	26.29	11.49	10.13	29.64	49.54	19.78	20.43
agnostic-adapter	16.24	**25.85**	17.90	21.71	**20.08**	**31.53**	11.75	9.05	**30.64**	54.04	22.79	21.19
stack-adapter	13.25	24.19	17.21	19.56	18.37	28.27	10.38	10.55	24.50	42.94	22.02	20.95
meta-learning	13.61	24.91	16.22	17.70	16.40	24.93	11.84	7.90	27.85	52.50	20.41	19.00
$m^4Adapter$	**18.99**	25.22	**17.94**	**21.71**	19.86	31.37	**12.12**	**12.05**	30.49	**54.30**	**23.92**	**21.32**
Δ	+2.75	-0.63	+0.04	+0.00	-0.22	-0.16	+0.37	+3.00	-0.15	+0.26	+1.13	+0.13

Table 4: Domain transfer via languages

	meta-adaptation language pair				specific DLP (de-en)					
	de-en	en-fr	fi-uk	is-lt	EUbookshop	KDE	OpenSubtitles	QED	TED	Ubuntu
m2m	24.52	29.20	12.34	12.55	19.59	26.48	15.89	26.34	28.14	30.65
m2m + FT	23.29	24.44	11.29	9.59	16.04	23.17	13.34	21.39	26.20	39.59
m2m + tag	22.52	24.97	11.71	11.22	15.86	23.67	11.72	20.64	25.97	37.25
agnostic-adapter	28.33	**30.93**	**15.42**	**14.38**	20.16	**28.72**	17.97	**27.66**	**33.63**	41.89
stack-adapter	23.37	24.96	11.51	11.09	16.14	22.51	13.84	22.29	27.67	36.73
meta-learning	25.08	28.26	13.40	12.83	17.88	21.20	16.32	24.96	30.32	39.81
$m^4Adapter$	**28.37**	30.80	15.24	14.05	**20.20**	28.19	**18.06**	27.18	33.32	**43.24**
Δ	+0.04	-0.13	-0.18	-0.33	+0.04	-0.53	+0.09	-0.48	-0.31	+1.35

Table 5: Language transfer via domains

Analysis

- **Main Results**
  - **Meta-Training**. Table 1 shows that $m^4Adapter$ obtains a performance that is on par or better than *agnostic-adapter*.
  - **Meta-Adaptation**. From Table 3, we observe that $m^4Adapter$ performs well when adapting to the new domains and new language pairs at the same time. In addition, no baseline system outperforms the original m2m model, which implies that these models are unable to transfer language or domain knowledge from the MNMT model.
  - **Efficiency**. As shown in Table 2, we find that $m^4Adapter$ only updates the adapter parameters while freezing the MNMT model's parameters. Therefore, it has fewer trainable parameters compared to fine-tuning.
- **Ablation Study**
  - **Domain Transfer via Languages**. In Table 4, we observe that $m^4Adapter$ outperform the original m2m model, indicating that the model encodes language knowledge and can transfer this knowledge to new domains.
  - **Language Transfer via Domains**. Table 5 shows that $m^4Adapter$ obtains a performance that is on par or better than the *agnostic-adapter*, and show the ability of domain transfer across different languages.

Conclusion

- We present $m^4Adapter$, a novel multilingual multi-domain NMT adaptation framework which combines meta-learning and parameter-efficient fine-tuning with adapters.
- $m^4Adapter$ is effective on adapting to new languages and domains simultaneously in low-resource settings.
- We find that $m^4Adapter$ also transfers language knowledge across domains and transfers domain information across languages.
- In addition, $m^4Adapter$ is efficient in training and adaptation, which is practical for online adaptation (etchegoyhen et al., 2021) to complex scenarios (new languages and new domains) in the real world.

Acknowledgement

This work was supported by funding to Wen Lai's PhD research from LMU-CSC (China Scholarship Council) Scholarship Program. This work has received funding from the European Research Council under the European Union's Horizon 2020 research and innovation program (grant agreement #640550). This work was also supported by the DFG (grant FR 2829/4-1).

Paper Code Blog