Domain Adaptation for Machine Translation in Few-Shot Setting

Wen Lai

Center for Information and Language Processing, LMU Munich, Germany

Helsinki NLP (02.03.2023)

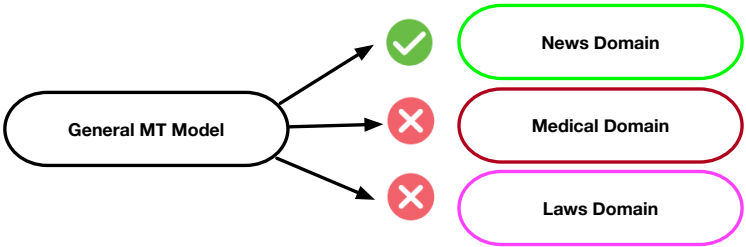
About me

- I am a second-year PhD student in the Center for information and language processing at LMU Munich under the supervision of Prof. Alexander Fraser.

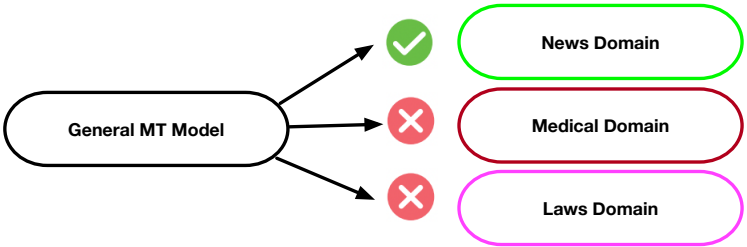
About me

- I am a second-year PhD student in the Center for information and language processing at LMU Munich under the supervision of Prof. Alexander Fraser.
- Research Interest
  - **Machine Translation**: Low-resource MT, Domain Adaptation, Multilingual MT
  - **Machine Learning**: Meta-Learning, Transfer Learning, Contrastive Learning

Domain Adaptation for Machine Translation



Domain Adaptation for Machine Translation



- Transfer learning: use of source domain D_s and source task T_s to improve the performance of target domain D_t and target task T_t.
- The information of D_s and T_s is transferred to D_t and T_t.
- Domain Adaptation for Machine Translation: a type of isomorphic transfer learning where $T_s = T_t$.

Challenges

- **Terminology**: Different domains often have their own unique terminology and jargon.

Challenges

- **Terminology**: Different domains often have their own unique terminology and jargon.
- **Style**: Different domains may have different writing styles and conventions.

Challenges

- **Terminology**: Different domains often have their own unique terminology and jargon.
- **Style**: Different domains may have different writing styles and conventions.
- **Context**: Different domains may require different levels of context to accurately translate text.

Challenges

- **Terminology**: Different domains often have their own unique terminology and jargon.
- **Style**: Different domains may have different writing styles and conventions.
- **Context**: Different domains may require different levels of context to accurately translate text.
- **Data availability**: Machine translation systems rely on large amounts of high-quality data to learn how to accurately translate text. However, data in different domains may be scarce or low-quality, which can make it difficult to develop accurate machine translation models.

Few-shot Learning

- **Goal**: How to use a small amount of data to train the model?

Few-shot Learning

- **Goal**: How to use a small amount of data to train the model?
- **Why we need few-shot learning in NLP?**

Few-shot Learning

- **Goal**: How to use a small amount of data to train the model?
- **Why we need few-shot learning in NLP?**
  - Limited labeled data

Few-shot Learning

- **Goal**: How to use a small amount of data to train the model?
- **Why we need few-shot learning in NLP?**
  - Limited labeled data
  - Rapid adaptation to new unseen domains

Few-shot Learning

- **Goal**: How to use a small amount of data to train the model?
- **Why we need few-shot learning in NLP?**
  - Limited labeled data
  - Rapid adaptation to new unseen domains
  - Human-like learning

Few-shot Learning

- **Goal**: How to use a small amount of data to train the model?
- **Why we need few-shot learning in NLP?**
  - Limited labeled data
  - Rapid adaptation to new unseen domains
  - Human-like learning
  - Generalization

Few-shot Learning

- **Goal**: How to use a small amount of data to train the model?
- **Why we need few-shot learning in NLP?**
  - Limited labeled data
  - Rapid adaptation to new unseen domains
  - Human-like learning
  - Generalization
- **Few-Shot Learning Techniques**: Meta-Learning, Data Augmentation, Transfer Learning, Zero-shot Learning, Active Learning.

Meta-Learning

- **Goal**: learn how to learn new tasks quickly and effectively with limited data by leveraging knowledge gained from previous tasks Lee u. a. (2022)[1].

[1] Meta Learning for Natural Language Processing: A Survey (Lee et al., NAACL 2022)

Meta-Learning

- **Goal**: learn how to learn new tasks quickly and effectively with limited data by leveraging knowledge gained from previous tasks Lee u. a. (2022)[1].
- **Two Stage**
  - Meta-Training: learns how to learn new tasks by observing and analyzing a set of "meta-training"tasks.
  - Meta-testing: uses the knowledge learned from meta-training stage to quickly adapt to new "task-level"tasks with limited data.

[1] Meta Learning for Natural Language Processing: A Survey (Lee et al., NAACL 2022)

Meta-Learning

- **Goal**: learn how to learn new tasks quickly and effectively with limited data by leveraging knowledge gained from previous tasks Lee u. a. (2022)[1].
- **Two Stage**
  - Meta-Training: learns how to learn new tasks by observing and analyzing a set of "meta-training"tasks.
  - Meta-testing: uses the knowledge learned from meta-training stage to quickly adapt to new "task-level"tasks with limited data.
- **Algorithm**
  - MAML Finn u. a. (2017)
  - FOMAML Finn u. a. (2017)
  - Reptile Nichol u. a. (2018)

[1] Meta Learning for Natural Language Processing: A Survey (Lee et al., NAACL 2022)

Two Settings

- **Bilingual Setting**
  - Wen Lai, Jindřich Libovický, and Alexander Fraser. 2022. Improving Both Domain Robustness and Domain Adaptability in Machine Translation. In Proceedings of the 29th International Conference on Computational Linguistics, pages 5191–5204, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.
- **Multilingual Setting**
  - Wen Lai, Alexandra Chronopoulou, and Alexander Fraser. 2022. m^4*Adapter*: Multilingual Multi-Domain Adaptation for Machine Translation with a Meta-Adapter. In Findings of the Association for Computational Linguistics: EMNLP 2022, pages 4282–4296, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Improving Both Domain Robustness and Domain Adaptability in Machine Translation

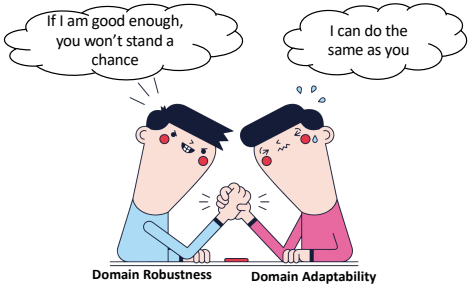
Wen Lai[1], Jindřich Libovický[2], Alexander Fraser[1]

[1]Center for Information and Language Processing, LMU Munich, Germany
[2]Faculty of Mathematics and Physics, Charles University, Prague, Czech Republic

COLING 2022

Motivation

- We consider **two problems** of NMT domain adaptation using meta-learning:
  - **Domain Robustness**: we want to reach high quality on both domains seen in the training data and unseen domains.
  - **Domain Adaptability**: we want to make it possible to finetune systems with just hundreds of in-domain parallel sentences.



Related Works

- Domain Adaptability
  - Traditional fine-tuning: an out-of-domain model is continually trained on in-domain data Dakwale und Monz (2017).
  - Meta-Learning: train models which can be later rapidly adapted to new scenarios using only a small amount of data (Sharaf u. a. (2020); Zhan u. a. (2021)).
- Domain Robustness
  - Müller u. a. (2020) defined the concept of domain robustness and propose to improve the domain robustness by subword regulaization, defensive distillation, reconstruction and neural noisy channel ranking.
  - Jiang u. a. (2020) proposed using individual modules for each domain with a word-level domain mixing strategy, which shows domain robustness on seen domains.

Challenges

- Domain Adaptability
  - Traditional fine-tuning: suffer from so-called catastrophic forgetting problem, resulting in deteriorated model performance in general domains.
  - Meta-Learning: neglect the problem of robustness towards domains unseen at training time.

Challenges

- Domain Adaptability
  - Traditional fine-tuning: suffer from so-called catastrophic forgetting problem, resulting in deteriorated model performance in general domains.
  - Meta-Learning: neglect the problem of robustness towards domains unseen at training time.
- Domain Robustness
  - The work on domain robustness, however, tends to neglect the adaptability of the models for new domains.

Research Question

- Which of these two properties is more important in NMT domain adaptation?

Research Question

- Which of these two properties is more important in NMT domain adaptation?
- Can we improve both of the two properties at the same time?

Goal

- Design a novel approach **RMLNMT** to improve both of the properties simultaneously.

Goal

- Design a novel approach **RMLNMT** to improve both of the properties simultaneously.
  - Domain Robustness

Goal

- Design a novel approach **RMLNMT** to improve both of the properties simultaneously.
  - Domain Robustness
    - We first train a word-level domain mixing model to improve the robustness on seen domains, and show that, surprisingly, this improves robustness on unseen domains as well.

Goal

- Design a novel approach **RMLNMT** to improve both of the properties simultaneously.
  - Domain Robustness
    - We first train a word-level domain mixing model to improve the robustness on seen domains, and show that, surprisingly, this improves robustness on unseen domains as well.
  - Domain Adaptability

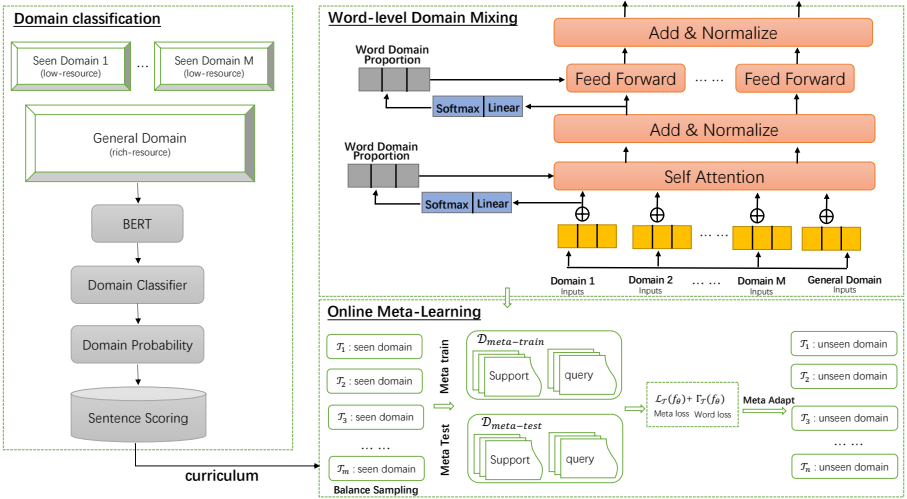
Goal

- Design a novel approach **RMLNMT** to improve both of the properties simultaneously.
  - Domain Robustness
    - We first train a word-level domain mixing model to improve the robustness on seen domains, and show that, surprisingly, this improves robustness on unseen domains as well.
  - Domain Adaptability
    - we train a domain classifier based on BERT to score training sentences; the score measures similarity between out-of-domain and general-domain sentences.

Goal

- Design a novel approach **RMLNMT** to improve both of the properties simultaneously.
  - Domain Robustness
    - We first train a word-level domain mixing model to improve the robustness on seen domains, and show that, surprisingly, this improves robustness on unseen domains as well.
  - Domain Adaptability
    - we train a domain classifier based on BERT to score training sentences; the score measures similarity between out-of-domain and general-domain sentences.
    - we integrate the domain-mixing model into a meta-learning framework with the domain classifier using a balanced sampling strategy.

Framework


Domain classification
Seen Domain 1
(low-resource)
Seen Domain M
(low-resource)
General Domain
(rich-resource)
BERT
Domain Classifier
Domain Probability
Sentence Scoring
curriculum
Word-level Domain Mixing
Word Domain Proportion
Softmax
Linear
Add & Normalize
Feed Forward
Feed Forward
Add & Normalize
Word Domain Proportion
Softmax
Linear
Self Attention
Domain 1 Inputs
Domain 2 Inputs
Domain M Inputs
General Domain Inputs
Online Meta-Learning
$\mathcal{T}_1$: seen domain
$\mathcal{T}_2$: seen domain
$\mathcal{T}_3$: seen domain
$\mathcal{T}_m$: seen domain
Balance Sampling
Meta train
Meta Test
$\mathcal{D}_{meta-train}$
Support
query
$\mathcal{D}_{meta-test}$
Support
query
$\mathcal{L}_\mathcal{T}(f_\theta) + \Gamma_\mathcal{T}(f_\theta)$
Meta loss
Word loss
Meta Adapt
$\mathcal{T}_1$: unseen domain
$\mathcal{T}_2$: unseen domain
$\mathcal{T}_3$: unseen domain
$\mathcal{T}_n$: unseen domain

Word-level Domain Mixing

- Observation
  - The domain of a word in the sentence is not necessarily consistent with the sentence domain. Therefore, we assume that every word in the vocabulary has a domain proportion, which indicates its domain preference.
- Domain Mixing in Multi-Head Attention Module
  - Each domain has its own multi-head attention modules. Therefore, we can integrate the domain proportion of each word into its multi-head attention module.
  - We process the domain mixing scheme in the same way for all attention layers and the fully-connected layers.
- The model can be efficiently trained by minimizing the composite loss:

$$L^* = L_{\text{gen}}(\theta) + L_{\text{mix}}(\theta)$$

Domain Classification

- Rieß u. a. (2021) show that using scores from simple domain classifier are more effective than scores from language models for NMT domain adaptation.
- We compute domain similarity using a sentence-level classifier, but in contrast with previous work, we based our classifier on a pre-trained language model (BERT).

Online Meta-Learning

- We use domain classification scores as the curriculum to split the corpus into small tasks, so that the sentences more similar to the general domain sentences are selected in early tasks.
- Previous meta-learning approaches (Sharaf u. a. (2020); Zhan u. a. (2021)) are based on token-size based sampling, which proved to be not balanced since some tasks did not contain all seen domains, especially in the early tasks. To address these issues, we sample the data uniformly from the domains to compensate for imbalanced domain distributions based on domain classifier scores.
- Following the balanced sampling, the process of meta-training is to update the current model parameter from θ to θ' using a MAML (Finn u. a. (2017)) objective with the traditional sentence-level meta-learning loss $\mathcal{L}_{\mathcal{T}}(f_\theta)$ and the word-level loss $\Gamma_{\mathcal{T}}(f_\theta)$ (L^* of $\mathcal{T}$).

$$L_{\mathcal{T}}(f_\theta) = \mathcal{L}_{\mathcal{T}}(f_\theta) + \Gamma_{\mathcal{T}}(f_\theta)$$

Datasets

- For English→German translation task, we evaluate ten domains, which publicly avaliable on OPUS [2];
  - *Bible, Books, ECB, EMEA, GlobalVoices, JRC, KDE, TED, WMT-News, COVID-19*
- For English→Chinese translation task, we use UM-Corpus [3] containing eight domains.
  - *Education, Microblog, Science, Subtitles, Laws, News, Spoken, Thesis*

[2] `https://opus.nlpl.eu`

[3] `http://nlp2ct.cis.umac.mo/um-corpus`

Baselines

- **Vanilla**. A standard Transformer-based NMT system. Note that we also use the $\mathcal{D}_{\text{meta-train}}$ corpus, which is more fair and stronger baseline;
- **Plain finetuning**. Fine-tune the vanilla system for each domain;
- **Plain finetuning + tag**. Using domain tag to fine-tune the system (Kobus et al., 2017);
- **Meta-MT**. Standard meta-learning approach (Sharaf u. a. (2020));
- **Meta-Curriculum (LM)**. Meta-Learning approach using LM scores as the curriculum to sample the task (Zhan u. a. (2021));
- **Meta-based w/o FT**. This series of experiments uses the meta-learning system prior to adaptation to the specific domain, which can be used to evaluate the domain robustness of meta-based models.

Baselines

- **Vanilla**. A standard Transformer-based NMT system. Note that we also use the $\mathcal{D}_{\text{meta-train}}$ corpus, which is more fair and stronger baseline;
- **Plain finetuning**. Fine-tune the vanilla system for each domain;
- **Plain finetuning + tag**. Using domain tag to fine-tune the system (Kobus et al., 2017);
- **Meta-MT**. Standard meta-learning approach (Sharaf u. a. (2020));
- **Meta-Curriculum (LM)**. Meta-Learning approach using LM scores as the curriculum to sample the task (Zhan u. a. (2021));
- **Meta-based w/o FT**. This series of experiments uses the meta-learning system prior to adaptation to the specific domain, which can be used to evaluate the domain robustness of meta-based models.
- *we enlarged the baseline models to have $\sqrt{k}$ times larger embedding dimension, so the baseline has the same number of parameters.

Main Results(Domain Robustness)

- Domain robustness shows the effectiveness of the model both in seen and unseen domains. Hence, we use the model without fine-tuning to evaluate the domain robustness.
- English→German

	Models	Unseen					Seen				
		Covid	Bible	Books	ECB	TED	EMEA	Globalvoices	JRC	KDE	WMT
1	Vanilla	24.34	**12.08**	12.61	29.96	27.89	37.27	24.19	39.84	27.75	27.38
2	Vanilla + tag	24.86	12.04	12.46	30.03	27.93	38.37	24.56	40.75	28.23	27.26
3	Meta-MT w/o FT	23.69	11.07	12.10	29.04	26.86	30.94	23.73	38.82	23.04	26.13
4	Meta-Curriculum (LM) w/o FT	23.70	11.16	12.24	28.22	27.21	33.49	24.27	39.21	27.60	25.83
5	RMLNMT w/o FT	**25.48**	11.48	**13.11**	**31.42**	**28.05**	**47.00**	**26.35**	**51.13**	**32.80**	**28.37**

- RMLNMT shows the best domain robustness compared with other models both in seen and unseen domains.
- The traditional meta-learning approach (Meta-MT, Meta-Curriculum) without fine-tuning is even worse than the standard transformer model in seen domains.

Main Results (Domain Adaptability)

- We evaluate the domain adaptability by testing that the model quickly adapts to new domains using just hundreds of in-domain parallel sentences.
- English→Chinese

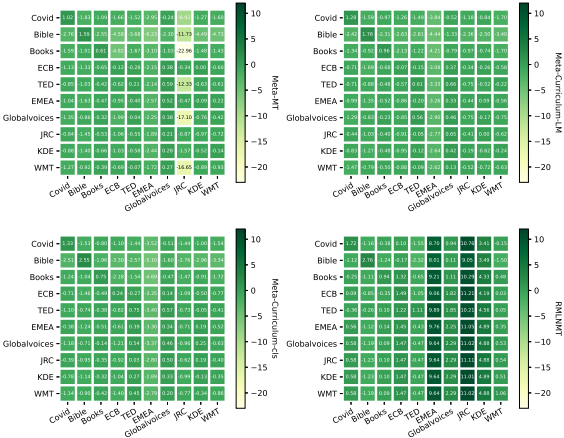
	Models	Unseen				Seen			
		Education	Microblog	Science	Subtitles	Laws	News	Spoken	Thesis
1	Plain FT	27.05	26.31	32.09	17.77	47.64	28.28	25.73	28.47
2	Plain FT + tag	27.13	26.48	32.12	17.94	47.91	28.84	26.35	29.58
3	Meta-MT + FT	29.33	27.48	33.12	18.77	45.21	28.43	26.82	29.20
4	Meta-Curriculum (LM) + FT	28.91	27.20	33.19	18.93	45.46	28.17	27.84	29.47
5	RMLNMT + FT	**30.91**	**28.52**	**34.51**	**20.13**	**57.58**	**30.42**	**28.03**	**32.25**

- we observe that the traditional meta-learning approach shows high adaptability to unseen domains but fails on seen domains due to limited domain robustness.
- In contrast, RMLNMT shows its domain adaptability both in seen and unseen domains, and maintains the domain robustness simultaneously.

Main Results (Cross-Domain Robustness)

- we use the fine-tuned model of one specific domain to generate the translation for other domains.
- Given k domains, we use the fine-tuned model M_J with the domain label of J to generate the translation of k domains.

Methods	Avg
Meta-MT	-1.97
Meta-Curriculum (LM)	-0.96
Meta-Curriculum (cls)	-0.98
RMLNMT	**2.64**



Further Analysis (Different Classifiers)

Classifier	Unseen					Seen				
	Covid	Bible	Books	ECB	TED	EMEA	Globalvoices	JRC	KDE	WMT
CNN	24.12	13.57	12.74	30.31	28.14	46.12	25.17	50.52	31.15	26.34
BERT-many-labels	25.89	14.77	13.71	**32.10**	29.28	**47.41**	**26.70**	51.34	32.76	28.17
BERT-2-labels	26.10	14.85	13.58	31.99	29.17	46.80	26.46	**51.56**	32.83	28.37
mBERT-many-labels	26.10	14.73	13.69	31.93	29.11	47.02	26.33	51.13	32.69	27.91
mBERT-2-labels	**26.53**	**15.37**	**13.71**	31.97	**29.47**	47.02	26.55	51.13	**32.88**	**28.37**

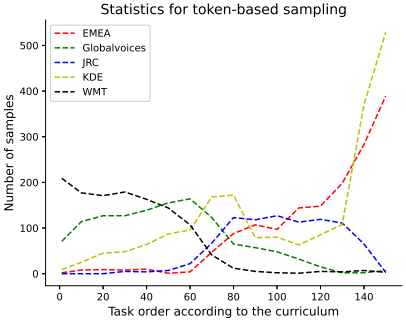
- **2-labels**: we distinguish only two classes: out-of-domain and in-domain
- **many-labels**: sentences are labeled with the respective domain labels.

- the performance of RMLNMT is not directly proportional to the accuracy of the classifier. This is because the accuracy of the classifier is close between BERT-based models and the primary role of the classifier is to construct the curriculum for splitting the tasks.

Further Analysis (Different Sampling Strategy)

Sampling Strategy	Unseen					Seen				
	Covid	Bible	Books	ECB	TED	EMEA	Globalvoices	JRC	KDE	WMT
Token-based sampling	25.30	11.38	12.70	31.61	28.01	47.51	26.50	**51.31**	32.88	28.03
Balance sampling	**25.47**	**11.51**	**12.79**	**32.08**	**28.98**	**47.64**	**26.58**	51.25	**32.91**	**28.07**

- Our methods can result in small improvements in performance.



Further Analysis (Different Fine-tuning Strategy)

Finetune Strategy	Unseen					Seen				
	Covid	Bible	Books	ECB	TED	EMEA	Globalvoices	JRC	KDE	WMT
FT-unseen	25.23	13.18	12.73	**32.45**	28.41	46.35	25.83	50.85	32.30	26.88
FT-seen	24.58	11.73	12.57	30.79	27.29	46.58	25.73	50.91	31.78	26.51
FT-all	15.00	7.77	9.06	21.33	16.98	24.69	14.63	27.59	12.77	15.75
FT-specific	**26.53**	**15.37**	**13.71**	31.97	**29.47**	**47.02**	**26.33**	**51.13**	**32.83**	**28.37**

- **FT-unseen**: fine-tuning using all unseen domain corpora
- **FT-seen**: fine-tuning using all seen domain corpora
- **FT-all**: fine-tuning using all out-of-domain corpora (seen and unseen domains)
- **FT-specific**: using the specific domain corpus to fine-tune the specific models

- Fine-tuning in one specific domain obtains robust results among all the strategies.

Conclusion

- We presented RMLNMT, a robust meta-learning framework for low-resource NMT domain adaptation reaching both high domain adaptability and domain robustness (both in the seen domains and unseen domains).
- We found that domain robustness dominates the results compared to domain adaptability in meta-learning based approaches.
- The results show that RMLNMT works best in setups that require high robustness in low-resource scenarios.

Email: lavine@cis.lmu.de
Homepage: `https://lavine-lmu.github.io`
Address: Oettingenstraße 67, 80538 Munich, Germany


Paper


Code


Blog

m^4*Adapter*: Multilingual Multi-Domain Adaptation for Machine Translation with a Meta-Adapter

Wen Lai, Alexandra Chronopoulou, Alexander Fraser

Center for Information and Language Processing, LMU Munich, Germany

Findings of EMNLP 2022

Background

- We consider **two problems** of Multilingual Multi-Domain Neural Machine Translation (MNMT) adaptation:
  - **Domain Adaptation**: adapt the MNMT model to a new domain.
  - **Language Adaptation**: adapt the MNMT model to a new language pair.

Background

- We consider **two problems** of Multilingual Multi-Domain Neural Machine Translation (MNMT) adaptation:
  - **Domain Adaptation**: adapt the MNMT model to a new domain.
  - **Language Adaptation**: adapt the MNMT model to a new language pair.
- Common Practice
  - *fine-tuning* the model on new domain / language pair data for NMT (Freitag and Al-Onaizan, 2016; Dakwale and Monz, 2017).
  - use lightweight, learnable units inserted between transformer layers, which are called *adapters* (Bapna and Firat, 2019).

Challenges

- Fine-tuning methods require updating the parameters of the whole model for each new domain, which is costly;

Challenges

- Fine-tuning methods require updating the parameters of the whole model for each new domain, which is costly;
- When fine-tuning on a new domain, catastrophic forgetting reduces the performance on all other domains, and proves to be a significant issue when data resources are limited;

Challenges

- Fine-tuning methods require updating the parameters of the whole model for each new domain, which is costly;
- When fine-tuning on a new domain, catastrophic forgetting reduces the performance on all other domains, and proves to be a significant issue when data resources are limited;
- Adapter-based approaches require training domain adapters for each domain and language adapters for all languages, which also becomes parameter-inefficient when adapting to a new domain and a new language because the parameters scale linearly with the number of domains and languages;

Challenges

- Fine-tuning methods require updating the parameters of the whole model for each new domain, which is costly;
- When fine-tuning on a new domain, catastrophic forgetting reduces the performance on all other domains, and proves to be a significant issue when data resources are limited;
- Adapter-based approaches require training domain adapters for each domain and language adapters for all languages, which also becomes parameter-inefficient when adapting to a new domain and a new language because the parameters scale linearly with the number of domains and languages;
- Current methods consider the two problems separately.

Research Question

- Can we improve both two adaptation problems at the same time?

Research Question

- Can we improve both two adaptation problems at the same time?
- Can we transfer the language knowledge across domains and domain knowledge across languages?

Method (Overview)

- We proposed a new approach m^{4}Adapter that consider a very challenging scenario: adapting the MNMT model both to a new domain and to a new language pair at the same time.

Method (Overview)

- We proposed a new approach m^{4}Adapter that consider a very challenging scenario: adapting the MNMT model both to a new domain and to a new language pair at the same time.
- 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).

Method (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'.

Method (Task Sampling)

- Given d domains and l languages, we sample some DLPs per batch among all $d \cdot l \cdot (l-1)$ tasks.
- We consider a standard *m-way-n-shot* meta-learning scenario: assuming access to $d \cdot l \cdot (l-1)$ DLPs, a *m-way-n-shot* task is created by first sampling m DLPs ($m \ll l \cdot (l-1)$); then, for each of the m sampled DLPs, $(n+q)$ examples of each DLP are selected; the n examples for each DLP serve as the support set to update the parameter of pre-trained model, while q examples constitute the query set to evaluate the model.
- We follow a temperature-based heuristic sampling strategy Aharoni et al., 2019[4], which defines the probability of any dataset as a function of its size.

[4] Massively Multilingual Neural Machine Translation (Aharoni et al., NAACL 2019)

Method (Meta-Learning Algorithm)

- We follow *Reptile* (Nichol et al., 2018[5]), an alternative first-order meta-learning algorithm that uses a simple update rule:

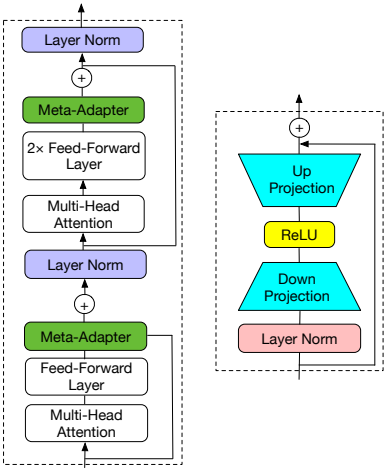
$$\psi \leftarrow \psi + \beta \frac{1}{|\{\mathcal{T}_i\}|} \sum_{\mathcal{T}_i \sim \mathcal{M}} (\psi_i^{(k)} - \psi)$$

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

[5] On first-order meta-learning algorithms (Nichol et al., arXiv preprint arXiv:1803.02999 (2018))

Method (Meta-Adapter)

- We propose training a *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.



Method (Meta-Adaptation)

- After the meta-training phase, the parameters of the adapter are fine-tuned to adapt to new tasks (as both the domain and language pair of interest are not seen during the meta-training stage) using a small amount of data to simulate a low-resource scenario.

Method (Meta-Adaptation)

- After the meta-training phase, the parameters of the adapter are fine-tuned to adapt to new tasks (as both the domain and language pair of interest are not seen during the meta-training stage) using a small amount of data to simulate a low-resource scenario.
- We find that this step is essential to our approach, as it permits adapting the parameters of the meta-learned model to the domain and language pair of interest. This step uses a very small amount of data (500 samples), which we believe could realistically be available for each DLP.

Datasets

- We split the datasets in two groups: *meta-training* and *meta-adapting*.
- We list the datasets used, each treated as a different domain: *EUbookshop*, *KDE*, *OpenSubtitles*, *QED*, *TED*, *Ubuntu*, *Bible*, *UN*, *Tanzil*, *Infopankki*. The datasets cover the following languages (ISO 639-1 language code[6]): *en*, *de*, *fr*, *mk*, *sr*, *et*, *hr*, *hu*, *fi*, *uk*, *is*, *lt*, *ar*, *es*, *ru*, *zh* and are publicly available on OPUS[7] (Tiedemann et al., 2012).

[6] https://en.wikipedia.org/wiki/List_of_ISO_639-1_codes
[7] https://opus.nlpl.eu

Baselines

- **m2m**: Using the original m2m model (Fan u. a. (2021)) to generate the translations;
- **m2m + FT**: Fine-tuning m2m on all DLPs;
- **m2m + tag**: Fine-tuning m2m with domain tags (Kobus et al., 2017) on all DLPs;
- **agnostic-adapter**: Mixing the data from all DLPs to train the adapters (Cooper Stickland u. a. (2021b)), to obtain language and domain-agnostic adapters;
- **stack-adapter**: Training two adapters for each language pair and domain, then stacking both adapters (Cooper Stickland u. a. (2021a));
- **meta-learning**: Traditional meta-learning methods using the MAML algorithm (Sharaf u. a. (2020)) on all DLPs.

Baselines

- **m2m**: Using the original m2m model (Fan u. a. (2021)) to generate the translations;
- **m2m + FT**: Fine-tuning m2m on all DLPs;
- **m2m + tag**: Fine-tuning m2m with domain tags (Kobus et al., 2017) on all DLPs;
- **agnostic-adapter**: Mixing the data from all DLPs to train the adapters (Cooper Stickland u. a. (2021b)), to obtain language and domain-agnostic adapters;
- **stack-adapter**: Training two adapters for each language pair and domain, then stacking both adapters (Cooper Stickland u. a. (2021a));
- **meta-learning**: Traditional meta-learning methods using the MAML algorithm (Sharaf u. a. (2020)) on all DLPs.
- * For stack-adapter, the number of language pair adapters and domain adapters to be trained is proportional to the number of language pairs and the number of domains.

Main Results (Domain Robustness)

- Motivated by Lai et al., 2022[a], we compare our approach to multiple baselines in terms of domain robustness.
- *m*^{4}*Adapter* obtains a performance that is on par or better than *agnostic-adapter*, which is a robust model.

[a] Improving Both Domain Robustness and Domain Adaptability in Machine Translation (Lai et al., COLING 2022)

	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*^{4}*Adapter	**23.89**	19.77	31.46	**32.91**

Main Results (Domain Adaptability)

	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*^{4}*Adapter	**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

- *m*^{4}*Adapter* performs well when adapting to the *meta-adaptation* domains and language pairs at the same time.
- We observe that no baseline system outperforms the original m2m model. This implies that these models are unable to transfer language or domain knowledge from the MNMT model.

Further Analysis (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.

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*^{4}*Adapter	3.17M (0.75%)	34%	300%

Further Analysis (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.
- *m*^{4}*Adapter* only updates the adapter parameters while freezing the MNMT model's parameters. therefore, it has fewer trainable parameters compared to fine-tuning.

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*^{4}*Adapter	3.17M (0.75%)	34%	300%

Further Analysis (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.
- *m^4Adapter* only updates the adapter parameters while freezing the MNMT model's parameters. therefore, it has fewer trainable parameters compared to fine-tuning.
- Our approach requires more time than traditional adapter methods but is faster compared with updating the entire model using traditonal meta-learning.

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%

Further Analysis (Domain Transfer via Languages)

- We define domain transfer via languages, i.e., the ability to transfer domains while keeping the languages unchanged.

	meta-adaptation domain							specific DLP (hr-sr)						
	EUbookshop	KDE	OpenSubtitles	QED	TED	Ubuntu	Bible	EUbookshop	KDE	OpenSubtitles	QED	TED	Ubuntu	Bible
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	22.33	2.01
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	**34.08**	12.57
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	34.15	13.25
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	28.83	10.59
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	25.41	10.14
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	31.24	10.42
m*^{4}*Adapter	**18.99**	25.22	**17.94**	**21.71**	19.86	31.37	**12.12**	**12.05**	30.49	**54.30**	**23.92**	**21.32**	33.71	**13.69**
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	+4.88	+3.1

Further Analysis (Domain Transfer via Languages)

- We define domain transfer via languages, i.e., the ability to transfer domains while keeping the languages unchanged.

	meta-adaptation domain							specific DLP (hr-sr)						
	EUbookshop	KDE	OpenSubtitles	QED	TED	Ubuntu	Bible	EUbookshop	KDE	OpenSubtitles	QED	TED	Ubuntu	Bible
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	22.33	2.01
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	**34.08**	12.57
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	34.15	13.25
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	28.83	10.59
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	25.41	10.14
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	31.24	10.42
m*^{4}*Adapter	**18.99**	25.22	**17.94**	**21.71**	19.86	31.37	**12.12**	**12.05**	30.49	**54.30**	**23.92**	**21.32**	33.71	**13.69**
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	+4.88	+3.1

- We observe that almost all baseline systems and *m*^{4}*Adapter* outperform the original m2m model, indicating that the model encodes language knowledge and can transfer this knowledge to new *meta-adaptation* domains.

Further Analysis (Domain Transfer via Languages)

- We define domain transfer via languages, i.e., the ability to transfer domains while keeping the languages unchanged.

	meta-adaptation domain							specific DLP (hr-sr)						
	EUbookshop	KDE	OpenSubtitles	QED	TED	Ubuntu	Bible	EUbookshop	KDE	OpenSubtitles	QED	TED	Ubuntu	Bible
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	22.33	2.01
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	**34.08**	12.57
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	34.15	13.25
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	28.83	10.59
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	25.41	10.14
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	31.24	10.42
m*^{4}*Adapter	**18.99**	25.22	**17.94**	**21.71**	19.86	31.37	**12.12**	**12.05**	30.49	**54.30**	**23.92**	**21.32**	33.71	**13.69**
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	+4.88	+3.1

- We observe that almost all baseline systems and *m*^{4}*Adapter* outperform the original m2m model, indicating that the model encodes language knowledge and can transfer this knowledge to new *meta-adaptation* domains.
- Our approach is comparable to the performance of *agnostic-adapter*, which performs the best among all baseline systems.

Further Analysis (Domain Transfer via Languages)

- We define domain transfer via languages, i.e., the ability to transfer domains while keeping the languages unchanged.

	meta-adaptation domain							specific DLP (hr-sr)						
	EUbookshop	KDE	OpenSubtitles	QED	TED	Ubuntu	Bible	EUbookshop	KDE	OpenSubtitles	QED	TED	Ubuntu	Bible
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	22.33	2.01
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	**34.08**	12.57
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	34.15	13.25
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	28.83	10.59
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	25.41	10.14
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	31.24	10.42
m*^{4}*Adapter	**18.99**	25.22	**17.94**	**21.71**	19.86	31.37	**12.12**	**12.05**	30.49	**54.30**	**23.92**	**21.32**	33.71	**13.69**
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	+4.88	+3.1

- We observe that almost all baseline systems and *m*^{4}*Adapter* outperform the original m2m model, indicating that the model encodes language knowledge and can transfer this knowledge to new *meta-adaptation* domains.
- Our approach is comparable to the performance of *agnostic-adapter*, which performs the best among all baseline systems.
- We also discover that domain transfer through languages is desirable in some distant domains.

Further Analysis (Language Transfer via Domains)

- We define language transfer via domains, i.e., the ability to transfer languages while keeping the domains unchanged.

	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*^{4}*Adapter	**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

Further Analysis (Language Transfer via Domains)

- We define language transfer via domains, i.e., the ability to transfer languages while keeping the domains unchanged.

	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*^{4}*Adapter	**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

- The performance of traditional fine-tuning approaches are poorer than the original m2m model, which means that these methods do not transfer the learned domain knowledge to the new *meta-adaptation* language pair.

Further Analysis (Language Transfer via Domains)

- We define language transfer via domains, i.e., the ability to transfer languages while keeping the domains unchanged.

	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*^{4}*Adapter	**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

- The performance of traditional fine-tuning approaches are poorer than the original m2m model, which means that these methods do not transfer the learned domain knowledge to the new *meta-adaptation* language pair.
- In contrast, *m*^{4}*Adapter* shows a performance that is on par or better than the *agnostic-adapter*, the most competitive model in all baseline systems.

Conclusion

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

Email: lavine@cis.lmu.de
Homepage: `https://lavine-lmu.github.io`
Address: Oettingenstraße 67, 80538 Munich, Germany


Paper


Code


Blog

Future Work

- **Online Domain Adaptation**: How to fast adapt a pretrained MT model to a new unseen domain?
- **Unsupervised Domain Adaptation**: How to use unsupervised machine learning and domain-invariant feture learning to adapt a model to new domains?
- **Low-Resource Domain Adaptation**: How to leverage additional resources, such as monolingual data, parallel data in related languages, or cross-lingual transfer learning to improve the performance of adaptation?
- **Domain-Specific Evaluation**: How to develop domain-specific evaluation metrics that capture the nuances and requirements of different domains?

Thank You!

lavine-lmu Lavine_Lai wen-lai

Email: lavine@cis.lmu.de
Homepage: `https://lavine-lmu.github.io`
Address: Oettingenstraße 67, 80538 Munich, Germany

References I

[Cooper Stickland u. a. 2021a] Cooper Stickland, Asa ; Berard, Alexandre ; Nikoulina, Vassilina: Multilingual Domain Adaptation for NMT: Decoupling Language and Domain Information with Adapters. In: *Proceedings of the Sixth Conference on Machine Translation*. Online : Association for Computational Linguistics, November 2021, S. 578–598. – URL `https://aclanthology.org/2021.wmt-1.64`

[Cooper Stickland u. a. 2021b] Cooper Stickland, Asa ; Li, Xian ; Ghazvininejad, Marjan: Recipes for Adapting Pre-trained Monolingual and Multilingual Models to Machine Translation. In: *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Online : Association for Computational Linguistics, April 2021, S. 3440–3453. – URL `https://aclanthology.org/2021.eacl-main.301`

References II

[Dakwale und Monz 2017] Dakwale, Praveen ; Monz, Christof: Fine-Tuning for Neural Machine Translation with Limited Degradation across In- and Out-of-Domain Data. In: *Proceedings of Machine Translation Summit XVI: Research Track*. Nagoya Japan, September 18 – September 22 2017, S. 156–169. – URL `https://aclanthology.org/2017.mtsummit-papers.13`

[Etchegoyhen u. a. 2021] Etchegoyhen, Thierry ; Ponce, David ; Gete, Harritxu ; Ruiz, Victor: Online Learning over Time in Adaptive Neural Machine Translation. In: *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*. Held Online : INCOMA Ltd., September 2021, S. 411–420. – URL `https://aclanthology.org/2021.ranlp-1.47`

References III

[Fan u. a. 2021] Fan, Angela ; Bhosale, Shruti ; Schwenk, Holger ; Ma, Zhiyi ; El-Kishky, Ahmed ; Goyal, Siddharth ; Baines, Mandeep ; Celebi, Onur ; Wenzek, Guillaume ; Chaudhary, Vishrav u. a.: Beyond english-centric multilingual machine translation. In: *The Journal of Machine Learning Research* 22 (2021), Nr. 1, S. 4839–4886

[Finn u. a. 2017] Finn, Chelsea ; Abbeel, Pieter ; Levine, Sergey: Model-agnostic meta-learning for fast adaptation of deep networks. In: *International conference on machine learning* PMLR (Veranst.), 2017, S. 1126–1135

[Jiang u. a. 2020] Jiang, Haoming ; Liang, Chen ; Wang, Chong ; Zhao, Tuo: Multi-Domain Neural Machine Translation with Word-Level Adaptive Layer-wise Domain Mixing. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online : Association for Computational Linguistics, Juli 2020, S. 1823–1834. – URL `https://aclanthology.org/2020.acl-main.165`

References IV

[Lee u. a. 2022] Lee, Hung-yi ; Li, Shang-Wen ; Vu, Thang: Meta Learning for Natural Language Processing: A Survey. In: *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Seattle, United States : Association for Computational Linguistics, Juli 2022, S. 666–684. – URL `https://aclanthology.org/2022.naacl-main.49`

[Müller u. a. 2020] Müller, Mathias ; Rios, Annette ; Sennrich, Rico: Domain Robustness in Neural Machine Translation. In: *Proceedings of the 14th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*. Virtual : Association for Machine Translation in the Americas, Oktober 2020, S. 151–164. – URL `https://aclanthology.org/2020.amta-research.14`

[Nichol u. a. 2018] Nichol, Alex ; Achiam, Joshua ; Schulman, John: On first-order meta-learning algorithms. In: *arXiv preprint arXiv:1803.02999* (2018)

References V

[Rieß u. a. 2021] Rieß, Simon ; Huck, Matthias ; Fraser, Alex: A Comparison of Sentence-Weighting Techniques for NMT. In: *Proceedings of Machine Translation Summit XVIII: Research Track*. Virtual : Association for Machine Translation in the Americas, August 2021, S. 176–187. – URL https://aclanthology.org/2021.mtsummit-research.15

[Sharaf u. a. 2020] Sharaf, Amr ; Hassan, Hany ; Daumé III, Hal: Meta-Learning for Few-Shot NMT Adaptation. In: *Proceedings of the Fourth Workshop on Neural Generation and Translation*. Online : Association for Computational Linguistics, Juli 2020, S. 43–53. – URL https://aclanthology.org/2020.ngt-1.5

[Zhan u. a. 2021] Zhan, Runzhe ; Liu, Xuebo ; Wong, Derek F. ; Chao, Lidia S.: Meta-curriculum learning for domain adaptation in neural machine translation. In: *Proceedings of the AAAI Conference on Artificial Intelligence* Bd. 35, 2021, S. 14310–14318