Improving both domain robustness and domain adaptability in machine translation

Anonymous ACL submission

Abstract

We address two problems of domain adaptation in neural machine translation. First, we want to reach domain *robustness*, i.e., good quality of both domains from the training data, and domains unseen in the training data. Second, we want our systems to be *adaptive*, i.e., making it possible to finetune systems with just hundreds of in-domain parallel sentences. In this paper, we introduce a novel combination of two previous approaches, word adaptive modelling, which addresses domain robustness, and meta-learning, which addresses domain adaptability, and we present empirical results showing that our new combination improves both of these properties.[1]

1 Introduction

The success of Neural Machine Translation (NMT; Bahdanau et al., 2015; Vaswani et al., 2017) heavily relies on large-scale high-quality parallel data, which is difficult to obtain in some domains. We study two major problems in NMT domain adaptation. First, models should work well on both seen domains (the domains in the training data) and unseen domains (domains which do not occur in the training data). We call this property *domain robustness*. Second, with just hundreds of in-domain sentences, we want to be able to quickly adapt to a new domain. We call this property *domain adaptability*. There are a few works attempting to solve these two problems. Jiang et al. (2020) proposed using individual modules for each domain with a word-level domain mixing strategy, which they showed has domain robustness on seen domains. We show that in fact word-level domain mixing can also have domain robustness on unseen domains, a new result. Sharaf et al. (2020); Zhan et al. (2021) use meta-learning approaches for improving on unseen domains. This work has strengths in adaptability to unseen domains but sacrifices robustness on seen domains.

Our goal is to develop a method which makes the model domain adaptable while maintaining robustness. We show that we can combine meta-learning with a robust word-level domain mixing system to obtain both domain robustness and domain adaptability simultaneously in a single model. The reasons are as follows: i) word-level domain mixing is better at capturing the domain-specific knowledge on seen domains, and is more adaptive in the process of domain knowledge sharing on unseen domains (Jiang et al., 2020); ii) meta-learning fails to work in seen domains, hence we considered using domain-specific knowledge learned from word-level domain mixing to improve the performance in seen domains; iii) meta-learning show its strength in adapting to new domains, allowing us to use the domain knowledge shared from seen domains to improve the performance on new unseen domains.

To achieve this, we propose RMLNMT (robust meta-learning NMT), a more robust and adaptive meta-learning-based NMT domain adaptation framework. 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. Then, we train a domain classifier based on BERT (Devlin et al., 2019) to score training sentences; the score measures similarity between out-of-domain and general-domain sentences. Finally, we improve domain adaptability by integrating the domain-mixing model into a meta-learning framework with the domain classifier using a balanced sampling strategy.

We evaluate RMLNMT on two translation tasks: English→German and English→Chinese. We conduct experiments for NMT domain adaptation in two low-resource scenarios. In the first scenario, a word-level domain mixing model is trained, and we carry out an evaluation of domain robustness.

[1] Our source code is attached and will be made publicly available.

We also show that meta-learning on the seen domains fails to improve the domain robustness on unseen domains. In the second scenario, we combine domain robust word-level domain mixing with meta-learning using only hundreds of in-domain sentences, and show that this combination has both domain robustness and domain adaptability.

The rest of the paper is organized as follows: We first describe related work (§2) and the models in detail (§3). Then we define the experimental setup (§4) and evaluate domain robustness and domain adaptability (§5). Finally, we analyse the results through an ablation study (§6).

2 Related Work

Domain Adaptation for NMT. Domain Adaptation for NMT typically uses additional in-domain monolingual data or a small amount of in-domain parallel data to improve the performance of domain translation in new domains (Chu and Wang, 2018). Current approaches can be categorized into two groups by granularity: From a sentence-level perspective, researchers either use data selection methods (Moore and Lewis, 2010; Axelrod et al., 2011) to select the training data that is similar to out-of-domain parallel corpora or train a classifier (Rieß et al., 2021) or utilize a language model (Wang et al., 2017; Zhan et al., 2021) to better weight the sentences. From a word-level perspective, researchers try to model domain distribution at the word level, since a word in a sentence can be related to more domains than just the sentence domain (Zeng et al., 2018; Yan et al., 2019; Hu et al., 2019; Sato et al., 2020; Jiang et al., 2020). In this work, we combine sentence-level (domain classifier) and word-level (domain mixing) domain information.

Curriculum Learning for NMT. Curriculum learning (Bengio et al., 2009) starts with easier tasks and then progressively gain experience to process more complex tasks and have proved useful in NMT domain adaptation. Stojanovski and Fraser (2019) utilize curriculum learning to improve anaphora resolution in NMT systems. Zhang et al. (2019) use a language model to compute a similarity score between domains, from which a curriculum is devised for adapting NMT systems to specific domains from general domains. Similarly, Zhan et al. (2021) use language model divergence scores as the curriculum to improve the performance of NMT domain adaptation with meta-learning in low-resource scenarios. In this paper, we improve the performance of NMT domain adaptation using curriculum learning based on a domain classifier.

Meta-Learning for NMT. Gu et al. (2018) apply model-agnostic meta-learning (MAML; Finn et al., 2017) to NMT. They show that MAML effectively improves low-resource NMT. Li et al. (2020) and Sharaf et al. (2020) propose to formulate the problem of low-resource domain adaptation in NMT as a meta-learning problem: the model learns to quickly adapt to an unseen new domain from a general domain. Recently, Zhan et al. (2021) propose to use language model divergence score as the curriculum to improve the performance of NMT domain adaptation. In this paper, we improve the *domain robustness* through a word-level domain-mixing model and integrate it into a meta-learning framework to improve the *domain adaptability*.

We approach meta-learning similarly to Zhan et al. (2021), which used the language model divergence score as curricula for improving the performance of NMT domain adaptation. In contrast, we use the probability of being out-of-domain assigned by the domain classifier to guide the curriculum; we also use a balanced sample strategy to split the tasks (see more details in Section 3.3). Furthermore, our meta-learning work does not use a plain transformer as the pre-trained model, but relies on a word-level domain mixing model (Jiang et al., 2020), which we will show is effective and robust in multi-domain adaptation. Finally, we use a stronger baseline, as we will discuss in the evaluation section (§4).

3 Method

In our initial experiments, we observed that the traditional meta-learning approach for NMT domain adaptation sacrifices the domain robustness on seen domains in order to improve the domain adaptability on unseen domains (see more details in Table 1 and Table 2, these will be discussed in Section 5). To address these issues, we propose a novel approach, RMLNMT, which combines meta-learning with a word-level domain-mixing system to improve both domain robustness and domain adaptability simultaneously in a single model. RMLNMT consists of three parts: Word-Level Domain Mixing, Domain Classification, and Online Meta-Learning. Figure 1 illustrates RMLNMT.

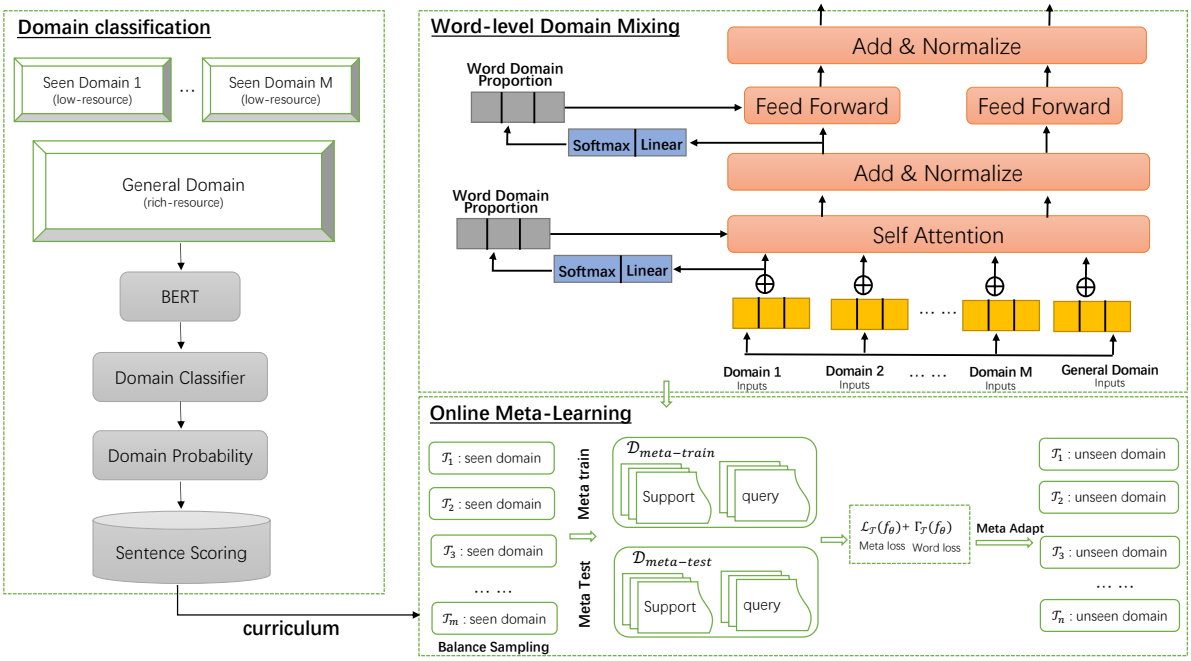

Figure 1: Method overview. The whole procedure mainly consists of three parts: domain classification, word-level domain mixing and online meta-learning.

3.1 Word-level Domain Mixing

In order to improve the robustness of NMT domain adaptation, we follow the approach of Jiang et al. (2020) and train the word-level layer-wise domain mixing NMT model. We provide a brief review of this approach here; please refer to Jiang et al. (2020) for more details.

Domain Proportion. From a sentence-level perspective (i.e., the classifier-based curriculum step), each sentence has a domain label. However, the domain of a word in the sentence is not necessarily consistent with the sentence domain. E.g., the word *doctor* shares the same embedding can have a different meaning in the medical domain and the academic domain. More specifically, for k domains, the embedding $\mathbf{w} \in \mathbb{R}^d$ of a word, and a matrix $R \in \mathbb{R}^{k \times d}$, the domain proportion of the word is represented by a smoothed softmax function as:

$$\Phi(\mathbf{w}) = (1-\epsilon) \cdot \mathrm{softmax}(R\mathbf{w}) + \epsilon/k,$$

where $\epsilon \in (0,1)$ is a smoothing parameter to prevent the output of $\Phi(\mathbf{w})$ from collapsing towards 0 or 1.

Domain Mixing. The standard Transformer (Vaswani et al., 2017) models the multi-head attention mechanism to focus on information in different representation subspaces from different positions:

$$\mathrm{MultiHead}(Q,K,V) = \mathrm{Concat}\left(h_1,\ldots,h_{\mathrm{h}}\right) W^O$$

$$h_i = \mathrm{Attention}\left(QW_i^Q, KW_i^K, VW_i^V\right),$$

where $W_i^Q, W_i^K, W_i^V \in \mathbb{R}^{d \times d/m}$ and $W^O \in \mathbb{R}^{d \times d}$. For the i-th head h_i, m is the number of heads, and d is the dimension of the model output.

Following Jiang et al. (2020), 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. Specifically, we take the weighted average of the linear transformation based on the domain proportion Φ. For example, we consider the point-wise linear transformation $\{W_{i,V,j}\}_{j=1}^k$ on the t-th word of the input, V_t, of all domains. The mixed linear transformation can be written as

$$\bar{V}_{i,t} = \sum_{j=1}^{k} V_t^\top W_{i,V,j} \Phi_{V,j}\left(V_t\right),$$

where $\Phi_{V,j}\left(V_t\right)$ denotes the j-th entry of $\Phi_V\left(V_t\right)$, and Φ_V is the domain proportion layer related to V. For other linear transformations, we apply the domain mixing scheme in the same way for all attention layers and the fully-connected layers.

Training. The model can be efficiently trained by minimizing a composite loss:

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

where θ contains the parameter in encoder, decoder and domain proportion. $L_{\mathrm{gen}}\left(\theta\right)$ denotes the

cross-entropy loss over training data $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^{n}$ and $L_{\text{mix}}(\theta)$ denotes the cross-entropy loss over the words/domain labels. For $L_{\text{mix}}(\theta)$, we compute the cross-entropy loss of its domain proportion $\Phi(\mathbf{w})$ as $-\log(\Phi_J(\mathbf{w}))$, which take J as the domain label. Hence, $L_{\text{mix}}(\theta)$ is computed as the sum of the cross-entropy loss over all such pairs of word labels of the training data.

3.2 Domain Classification

Domain similarity has been successfully applied in NMT domain adaptation. Moore and Lewis (2010) calculate cross-entropy scores with a language model to represent the domain similarity. Rieß et al. (2021) leverage simple classifiers to compute similarity scores; these scores are more effective than scores from language models for domain adaptation of NMT. Following Rieß et al. (2021), we compute domain similarity using a sentence-level classifier, but in contrast with their work, we based our classifier on a pre-trained language model.

Given k domain corpora (one general domain corpus and n out-of-domain corpora), we trained a sentence classification model M based on BERT (Devlin et al., 2019). For a sentence x with a domain label L_x, a simple softmax is added to the top of the model M to predict the domain probability of sentence x:

$$P(x \mid h) = \text{softmax}(Wh),$$

where W is the parameter matrix of M and h is the hidden state of M. $P(x \mid h)$ is a probability set, which contains k probability scores indicating the similarity of sentence x to each domain. A higher probability P of general domain means the domain of sentence x is more similar to the general domain, and vice versa. We finally select the probability of the general domain as the score of the sentence x and use this score as the curriculum to split the task in meta-learning (see more details in Section 3.3). A higher score indicates that the sentence is more similar to the general domain, so we will select it earlier.

3.3 Online Meta-Learning

The idea of meta-learning is to use a small set of source tasks $\{\mathcal{T}_1, \ldots, \mathcal{T}_n\}$ to find the initialization of model parameters θ from which learning a target task $\mathcal{T}_0$ would require only a small number of training examples. Meta-learning algorithms

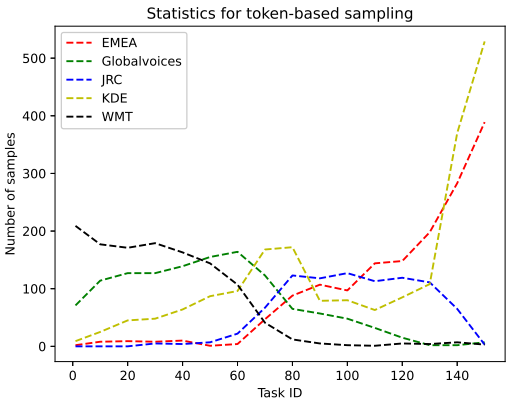

Figure 2: The statistic of samples in the task for tokenize-based splitting strategy.

consist of three main steps: (i) split the seen domain corpus into small tasks $\mathcal{T}$ containing a small amount of data as $\mathcal{D}_{\text{meta-train}}$ and $\mathcal{D}_{\text{meta-test}}$ to simulate the low-resource scenarios. Data for each task $\mathcal{T}_i$ is decomposed into two sub-sets: a support set $\mathcal{T}_{support}$ used for training the model and a query set $\mathcal{T}_{query}$ used for evaluating the model; (ii) leverage a meta-learning policy to adapt model parameters to different small tasks using $\mathcal{D}_{\text{meta-train}}$ datasets. We use MAML, proposed by Finn et al. (2017), and instantiated for the meta-learning to adapt the NMT systems in different domains; (iii) finetune the model using the support set of $\mathcal{D}_{\text{meta-test}}$. Algorithm 1 shows the complete algorithm.

Split Tasks. Zhan et al. (2021) propose a curriculum-based task splitting strategy, which uses divergence scores computed by a language model as the curriculum to split the corpus into small tasks. We follow a similar idea, but propose to use predictions from a domain classifier as the criterion for splitting the data. Concretely, we first train a domain classifier with BERT; the classifier scores sentences, indicating domain similarity between an in-domain sentence and a general domain sentence (see Section 3.2). The tasks are then split according to the scores; sentences more similar to the general domain sentences are selected in early tasks.

Balanced Sampling. Traditional meta-learning approaches (Sharaf et al., 2020; Zhan et al., 2021) are based on token-size based sampling, which uses $8k$ or $16k$ token sizes split into many small tasks. However, the splitting process for the domain is not balanced, since some tasks did not contain all seen domains, especially in the early tasks. As we can see in Figure 2, the token-based splitting methods usually allocate more samples on domain-similar

domains (*WMT*, *Globalvoices*) and allocate small samples on domain-distant domains (*EMEA*, *JRC*) in the sampling of early tasks. This can cause problems in our method since the model architecture is dynamically changing according to the numbers of domains (see more details in Section 3.1).

To address these issues, we sample the data uniformly from the domains to compensate for imbalanced domain distributions based on domain classifier scores.

Meta-Training. Following the balanced sampling, the process of meta-training is to update the current model parameter on $\mathcal{T}_{support}$ from θ to θ', and then evaluate on $\mathcal{T}_{query}$. The model parameter θ' is updated to minimize the meta-learning loss through MAML.

Given a pre-trained model f_θ (initialized with parameter θ trained on word-level domain mixing) and the meta-train data $\mathcal{D}_{\text{meta-train}}$, for each task $\mathcal{T}$, we learn to use one gradient update the model parameter from θ to θ' as follows:

$$\theta' = \theta - \alpha \nabla_\theta L_\mathcal{T}(f_\theta)$$

where α is the learning rate and L is the loss function. In our methods, we consider both 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 $\mathcal{T}$) calculated from the word-level domain mixing pre-trained model. More formally, the loss is updated as follows:

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

Note that the meta-training phrase is not adapted to a specific domain, so it can be used as a metric to evaluate the domain robustness of the model.

Meta-Adaptation. After the meta-training phase, the parameters are updated to adapt to each domain using the small *support set* of $\mathcal{D}_{\text{meta-test}}$ corpus to simulate the low-resource scenarios. Then performance is evaluated on the *query set* of $\mathcal{D}_{\text{meta-test}}$.

4 Experiments

Datasets. We experiment with English→German (**en2de**) and English→Chinese (**en2zh**) translation tasks. For the *en2de* task, we use the same corpora as Zhan et al. (2021). The data consists of corpora in nine domains (Bible, Books, ECB, EMEA, GlobalVoices, JRC, KDE, TED, WMT-News) publicly available on OPUS[2] (Tiedemann, 2012) and COVID-19 corpus[3]. For *en2zh*, we use UM-Corpus (Tian et al., 2014) containing eight domains: Education, Microblog, Science, Subsitles, Laws, News, Spoken, Thesis. We use WMT14 (*en2de*) and WMT18 (*en2zh*) corpus published on the WMT website[4] as our general domain corpora. We use WMT19 English monolingual corpora to train the LM model so that we can reproduce results from previous work.

Data Preprocessing. For English and German, we preprocessed all data with the Moses tokenizer[5] and use sentencepiece[6] (Kudo and Richardson, 2018) to encode the corpus with a joint vocabulary, with size 40,000. After that, we filter the sentence longer than 175 tokens and deduplicate the corpus. For Chinese, we perform segmentation with `pkuseg` [7] (Luo et al., 2019). To have a fair comparison with previous methods (Sharaf et al., 2020; Zhan et al., 2021), we use the same setting, which randomly sub-sampled $\mathcal{D}_{\text{meta-train}}$ and $\mathcal{D}_{\text{meta-test}}$ for each domain with fixed token sizes in order to simulate domain adaptation tasks in low-resource scenarios. More details for data used in this paper can be found in Appendix A.1.

Algorithm 1 RMLNMT (Robust Meta-Learning NMT Domain Adaptation)

Require: Domain classifier model cls; Pretrained domain-mixing model θ;

Score the sentence in $\mathcal{D}_{\text{meta-train}}$ using cls
for N epochs **do**
  Split corpus into n tasks based on step 1
  Balance sample through all tasks
  for task $\mathcal{T}_i$, $i = 1 \dots n$ **do**
    Evaluate loss $L_\mathcal{T}(f_\theta) = \mathcal{L}_{\mathcal{T}_i}(f_\theta) + \Gamma_{\mathcal{T}_i}(f_\theta)$ on support set
    Update the gradient with parameters $\theta' = \theta - \alpha \nabla_\theta L_\mathcal{T}(f_\theta)$
  end for
  Update the gradient with parameters $\theta' = \theta - \beta \nabla_\theta L_\mathcal{T}(f_\theta)$ on query set
end for
return RMLNMT model parameter θ'

[1] We confirmed with Zhan et al. (2021) via email that they did not deduplicate the corpus, which is the main reason for the difference between our results and their results.

[2] opus.nlpl.eu
[3] github.com/NLP2CT/Meta-Curriculum
[4] http://www.statmt.org
[5] github.com/moses-smt/mosesdecoder
[6] github.com/google/sentencepiece
[7] github.com/lancopku/pkuseg-python

	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	**Meta-MT w/o FT**	23.69	11.07	12.10	23.04	26.86	30.94	23.73	38.82	23.04	26.13
3[1]	**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
4	**Meta-Curriculum (cls) w/o FT**	24.03	11.30	12.29	27.49	27.61	32.16	24.55	39.07	26.92	25.83
5	**Word-level Adaptive**	**25.43**	**12.53**	**13.11**	**31.11**	**28.50**	**47.28**	**24.70**	**50.99**	**30.93**	26.64

Table 1: Domain Robustness: BLEU scores on the English → German translation task. *w/o* denotes the meta-learning systems without fine-tuning, FT denotes fine-tuning. Best results are highlighted in bold.

	Models	Unseen					Seen				
		Covid	Bible	Books	ECB	TED	EMEA	Globalvoices	JRC	KDE	WMT
1	**Plain FT**	24.81	12.61	12.78	30.48	28.36	37.26	24.26	40.02	27.99	27.31
2	**Meta-MT + FT**	25.83	14.20	13.39	30.36	28.57	34.69	24.64	39.15	27.47	26.38
3	**Meta-Curriculum (LM) + FT**	**26.66**	14.37	13.70	30.41	28.97	34.00	24.72	39.61	27.37	26.68
4	**Meta-Curriculum (cls) + FT**	26.14	15.16	13.53	30.72	29.11	33.96	24.72	39.40	27.86	26.45
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 + FT	26.53	**15.37**	**13.72**	**31.97**	**29.47**	**47.02**	**26.55**	**51.13**	**32.88**	**28.37**

Table 2: Domain Adaptability: BLEU scores on the English → German translation task.

Baselines. We compare RMLNMT with the following baselines: i) **Vanilla**. A standard Transformer-based NMT system trained on the general domains (WMT14 for *en2de*, WMT18 for *en2zh*) and $\mathcal{D}_{\text{meta-train}}$ corpus in seen-domains. We use the $\mathcal{D}_{\text{meta-train}}$ corpus because meta-learning-based methods also use the $\mathcal{D}_{\text{meta-train}}$ corpus, this is a more fair and stronger baseline. ii) **Plain fine-tuning**. Fine-tune the vanilla system on support set of $\mathcal{D}_{\text{meta-test}}$. iii) **Meta-MT**. Standard meta-learning approach on domain adaptation task, which learns to adapt to new unseen domains based on a meta-learned model (Sharaf et al., 2020). iv) **Meta-Curriculum (LM)**. Meta-learning approach for domain adaptation using LM score as the curriculum to sample the task (Zhan et al., 2021). v) **Meta-Curriculum (cls)**. Similar to Meta-Curriculum (LM), domain classifier score is used instead of LM. vi) **Meta-based w/o FT**. This series of experiments uses the meta-learning system prior to adaptation to the specific domain. This can be used to evaluate the domain robustness of meta-based models (see more details in the meta-training part of Section 3.3). vii) **Word-Level Adaptive**. Multi-domain NMT with word-level layer-wise domain mixing (Jiang et al., 2020).

Implementation. We use the Transformer model (Vaswani et al., 2017) implemented in `fairseq` (Ott et al., 2019). For our word-level domain-mixing modules, we dynamically adjust the network structure according to the number of domains since every domain has its multi-head layers. Hence, the number of model parameters in the attentive sub-layers of RMLNMT is k times the number in the standard transformer (k is the number of domains in the training data). Following Jiang et al. (2020), we enlarged the baseline models to have $\sqrt{k}$ times larger embedding dimension, so the baseline has the same number of parameters. This should rule out that the improvements are due to increased parameter count rather than modeling improvements. For our meta-learning framework, we consider the general meta loss and word-adaptive loss together (as seen in Section 3.3). More details on hyper-parameters are listed in Appendix A.2.

Evaluation. For a fair comparison with previous work, we use the same data from the support set of $\mathcal{D}_{\text{meta-test}}$ to finetune the model and the same data from the query set of $\mathcal{D}_{\text{meta-test}}$ to evaluate the models. We measure case-sensitive detokenized BLEU with SacreBLEU (Post, 2018); beam search with a beam of size five is used. Because of the recent criticism of BLEU score (Mathur et al., 2020), we also evaluate our models using chrF (Popović, 2015) and COMET (Rei et al., 2020); the results are listed in Appendix A.5. We evaluated the performance of each model in terms of domain robustness and domain adaptability separately.

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.

Domain Adaptability. We evaluate the domain adaptability by testing that the model quickly adapts to new domains using just hundreds of in-

	Models	Unseen				Seen			
		Education	Microblog	Science	Subtitles	Laws	News	Spoken	Thesis
1	**Vanilla**	**6.46**	5.23	7.74	3.07	37.10	6.67	4.14	14.38
2	**Meta-MT w/o FT**	4.80	4.20	5.25	1.94	10.57	6.52	4.34	6.04
3	**Meta-Curriculum (LM) w/o FT**	5.65	5.01	5.35	1.87	24.83	6.66	**4.38**	7.25
4	**Meta-Curriculum (cls) w/o FT**	4.83	3.84	5.61	2.72	20.37	6.97	4.41	4.87
5	**Word-level Adaptive**	6.36	**5.37**	**8.09**	**3.21**	**38.48**	**7.82**	4.21	**14.94**

Table 3: Domain Robustness: BLEU scores on English → Chinese translation tasks.

	Models	Unseen				Seen			
		Education	Microblog	Science	Subtitles	Laws	News	Spoken	Thesis
1	**Plain FT**	6.02	5.95	7.73	3.10	37.06	6.43	5.05	14.68
2	**Meta-MT + FT**	6.03	5.89	7.34	2.17	30.18	5.93	5.08	11.32
3	**Meta-Curriculum (LM) + FT**	5.84	5.72	7.25	2.36	31.70	6.85	5.14	12.10
4	**Meta-Curriculum (cls) + FT**	6.14	5.73	7.70	1.93	30.75	6.58	5.62	12.04
5	**RMLNMT w/o FT**	6.34	4.54	8.27	3.15	38.70	8.37	**6.12**	15.21
	RMLNMT + FT	**7.28**	**6.21**	**9.37**	**4.45**	**38.73**	**8.41**	6.08	**15.28**

Table 4: Domain Adaptability: BLEU scores on English → Chinese translation tasks.

domain parallel sentences. Therefore, we fine-tune the models on a small amount of domain-specific data.

Cross-Domain Robustness. To better show the cross-domain robustness of RMLNMT, we use the fine-tuned model of one specific domain to generate the translation for other domains. More formally, given k domains, we use the fine-tuned model M_J with the domain label of J to generate the translation of k domains. We calculate the BLEU score difference between the translations generated in the different domains and the vanilla baseline separately. The results are as shown in Appendix A.4.

5 Results

Domain Robustness. Tables 1 and 3 show the domain robustness of the models. As we can see, the word-level domain mixing model shows the best domain robustness compared with other models both in seen and unseen domains. In addition, the traditional meta-learning approach without fine-tuning is even worse than the standard transformer model. Note this setup differs from the previous work (Sharaf et al., 2020; Zhan et al., 2021) because we included the $\mathcal{D}_{\text{meta-train}}$ data to the vanilla system to insure all systems in the table use the same training data. Interestingly, the translation quality in the *WMT* domain decreases with the increasing robustness in other domains. We speculate this might be due overfiting of the vanilla system to the WMT domain.

Domain Adaptability. Tables 2 and 4 show the domain adaptability of the models. 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. One interesting observation is that RMLNMT does not improve much on seen domains after finetuning, because the meta-learning model without finetuning is already strong enough due to the domain robustness of word-level domain mixing.

The results of both domain robustness and domain adaptability are consistent for the chrF and COMET evaluation metrics (see more details in Tables 13 and 14 of Appendix A.5).

6 Analysis

In this section, we conduct additional experiments to better understand the strengths of RMLNMT. We analyze the contribution of different components in RMLNMT, through an ablation study.

Different classifiers. Tables 1, 2, 3 and 4 show that the classifier-based curriculum slightly outperforms the curriculum derived from language models. We evaluate the impact of different classifiers on translation performance. The main results are as shown in Table 5 (see more details in Appendix A.3). We observed that the performance of RMLNMT is not directly proportional to the accuracy of the classifier. In other words, slightly higher classification accuracy does not lead to better BLEU scores. This is because the accuracy of the classifier is close between BERT-based models

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**

Table 5: Different classifier: BLEU scores on the English → German translation task.

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**

Table 6: Different sampling strategy: BLEU scores on the English → German translation task.

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-unseen	**26.53**	**15.37**	**13.71**	31.97	**29.47**	**47.02**	**26.33**	**51.13**	**32.83**	**28.37**

Table 7: Different fine-tuning strategy: BLEU scores on the English → German translation task.

and the primary role of the classifier is to construct the curriculum for splitting the tasks. When we use a significantly worse classifier, i.e., the CNN in our experiments, the overall performance of RMLNMT is worse than the BERT-based classifier.

Balanced sampling vs. Token-based sampling. Plain meta-learning uses a token-based sampling strategy to split sentences into small tasks. However, the token-based strategy could cause unbalanced domain distribution in some tasks, especially in the early stage of training due to domain mismatches (see the discussion of balanced sampling in Section 3.3). To address this issue, we proposed to balance the domain distribution after splitting the task. Table 6 shows that our methods can result in small improvements in performance. For example, in the *Books* domain, BLEU was 12.70 with token-based sampling, but with the balanced sampling strategy BLEU was 12.79. We keep the same number of tasks to have a fair comparison with previous methods.

Different fine-tuning strategies. As described in Section 3.1, the model for each domain has its own multi-head and feed-forward layers. During fine-tuning on one domain corpus (support set of $\mathcal{D}_{\text{meta-test}}$), we devise four strategies: i) **FT-unseen**: fine-tuning using all unseen domain corpora; ii) **FT-seen**: fine-tuning using all seen domain corpora; iii) **FT-all**: fine-tuning using all out-of-domain corpora (seen and unseen domains); iv) **FT-specific**: using the specific domain corpus to fine-tune the specific models. The results are shown in Table 7. *FT-specific* obtains robust results among all the strategies. Although other strategies outperform *FT-specific* in some domains, *FT-specific* is robust across all domains. Furthermore, *FT-specific* is the fairest comparison because it uses only a specific domain corpus to fine-tune, which is the same as the baseline systems.

7 Conclusion

We presented `RMLNMT`, a robust meta-learning framework for low-resource NMT domain adaptation reaching both high domain adaptability and domain robustness. Unlike previous methods which sacrifice the performance on other domains, our proposed methods keeps robustness on all domains (both in seen and unseen domains). We show consistent improvements in translation from English to German and English to Chinese. RMLNMT is recommended for those who would want systems that are domain-robust and domain adaptable in low-resource scenarios. Our future directions include extending RMLNMT to multilingual domain adaptation, reducing model parameters while ensuring domain robustness and adaptability.

References

Amittai Axelrod, Xiaodong He, and Jianfeng Gao. 2011. Domain adaptation via pseudo in-domain data selection. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 355–362, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations*.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48.

Chenhui Chu and Rui Wang. 2018. A survey of domain adaptation for neural machine translation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1304–1319, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135. PMLR.

Jiatao Gu, Yong Wang, Yun Chen, Victor O. K. Li, and Kyunghyun Cho. 2018. Meta-learning for low-resource neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3622–3631, Brussels, Belgium. Association for Computational Linguistics.

Junjie Hu, Mengzhou Xia, Graham Neubig, and Jaime Carbonell. 2019. Domain adaptation of neural machine translation by lexicon induction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2989–3001, Florence, Italy. Association for Computational Linguistics.

Haoming Jiang, Chen Liang, Chong Wang, and Tuo Zhao. 2020. 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*, pages 1823–1834, Online. Association for Computational Linguistics.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Rumeng Li, Xun Wang, and Hong Yu. 2020. Metamt, a meta learning method leveraging multiple domain data for low resource machine translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8245–8252.

Ruixuan Luo, Jingjing Xu, Yi Zhang, Xuancheng Ren, and Xu Sun. 2019. Pkuseg: A toolkit for multi-domain chinese word segmentation. *arXiv preprint arXiv:1906.11455*.

Nitika Mathur, Timothy Baldwin, and Trevor Cohn. 2020. Tangled up in BLEU: Reevaluating the evaluation of automatic machine translation evaluation metrics. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4984–4997, Online. Association for Computational Linguistics.

Robert C. Moore and William Lewis. 2010. Intelligent selection of language model training data. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 220–224, Uppsala, Sweden. Association for Computational Linguistics.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference*

on Empirical Methods in Natural Language Processing (EMNLP), pages 2685–2702, Online. Association for Computational Linguistics.

Simon Rieß, Matthias Huck, and Alex Fraser. 2021. A comparison of sentence-weighting techniques for NMT. In *Proceedings of the 18th Biennial Machine Translation Summit (Volume 1: Research Track)*, pages 176–187, Virtual. Association for Machine Translation in the Americas.

Shoetsu Sato, Jin Sakuma, Naoki Yoshinaga, Masashi Toyoda, and Masaru Kitsuregawa. 2020. Vocabulary adaptation for domain adaptation in neural machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4269–4279, Online. Association for Computational Linguistics.

Amr Sharaf, Hany Hassan, and Hal Daumé III. 2020. Meta-learning for few-shot NMT adaptation. In *Proceedings of the Fourth Workshop on Neural Generation and Translation*, pages 43–53, Online. Association for Computational Linguistics.

Dario Stojanovski and Alexander Fraser. 2019. Improving anaphora resolution in neural machine translation using curriculum learning. In *Proceedings of Machine Translation Summit XVII: Research Track*, pages 140–150, Dublin, Ireland. European Association for Machine Translation.

Liang Tian, Derek F. Wong, Lidia S. Chao, Paulo Quaresma, Francisco Oliveira, Yi Lu, Shuo Li, Yiming Wang, and Longyue Wang. 2014. UM-corpus: A large English-Chinese parallel corpus for statistical machine translation. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1837–1842, Reykjavik, Iceland. European Language Resources Association (ELRA).

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2214–2218, Istanbul, Turkey. European Language Resources Association (ELRA).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Rui Wang, Masao Utiyama, Lemao Liu, Kehai Chen, and Eiichiro Sumita. 2017. Instance weighting for neural machine translation domain adaptation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1482–1488, Copenhagen, Denmark. Association for Computational Linguistics.

Shen Yan, Leonard Dahlmann, Pavel Petrushkov, Sanjika Hewavitharana, and Shahram Khadivi. 2019. Word-based domain adaptation for neural machine translation. *arXiv preprint arXiv:1906.03129*.

Jiali Zeng, Jinsong Su, Huating Wen, Yang Liu, Jun Xie, Yongjing Yin, and Jianqiang Zhao. 2018. Multi-domain neural machine translation with word-level domain context discrimination. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 447–457, Brussels, Belgium. Association for Computational Linguistics.

Runzhe Zhan, Xuebo Liu, Derek F. Wong, and Lidia S. Chao. 2021. Meta-curriculum learning for domain adaptation in neural machine translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 14310–14318.

Xuan Zhang, Pamela Shapiro, Gaurav Kumar, Paul McNamee, Marine Carpuat, and Kevin Duh. 2019. Curriculum learning for domain adaptation in neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1903–1915, Minneapolis, Minnesota. Association for Computational Linguistics.

A Appendix

A.1 Datasets

For the OPUS corpus used in English → German translation task, we deduplicated the corpus, which is different from (Zhan et al., 2021) and is the main reason that we cannot reproduce the results in the original paper. The statistics of the original OPUS are shown in Table 8. The seen domains (EMEA, Globalvoices, JRC, KDE, WMT) contain a lot of duplicated sentences. The scores in the original paper are too high because the $\mathcal{D}_{\text{meta-train}}$ dataset overlaps with some sentences in $\mathcal{D}_{\text{meta-test}}$.

Corpus	Original	Deduplicated
Covid	3,325	3,312
Bible	62,195	61,585
Books	51,467	51,106
ECB	113,116	113,081
TED	143,830	142,756
EMEA	1,103,807	360,833
Globalvoices	71,493	70,519
JRC	717,988	503,789
KDE	223,672	187,918
WMT	45,913	34,727

Table 8: Data statistic of the original corpus for English→German translation task

For the meta-learning phase, to have a fair comparison with previous methods, we use the same setting. We random split 160 tasks and 10 tasks respectively in $\mathcal{D}_{\text{meta-train}}$ and $\mathcal{D}_{\text{meta-test}}$ to simulate the low-resource scenarios. For each task, the token amount of support set and query set is a strict limit to $8K$ and $16K$. $\mathcal{D}_{\text{meta-dev}}$ corpus is limited to 5000 sentences for each domain. Table 9 and Table 10 shows the detailed statistics of English → German and English → Chinese tasks.

A.2 Model Configuration

We use the `Transformer Base` architecture (Vaswani et al., 2017) as implemented in fairseq (Ott et al., 2019). We use the standard Transformer architecture with dimension 512, feed-forward layer 2048, 8 attention heads, 6 encoder layers and 6 decoder layers. For optimization, we use the Adam optimizer with a learning rate of $5 \cdot 10^{-5}$. To prevent overfitting, we applied a dropout of 0.3 on all layers. The number of warm-up steps was set to 4000. At the time of inference, a beam search of size 5 is used to balance the decoding time and accuracy of the search.

	$\mathcal{D}_{\text{meta-train}}$		$\mathcal{D}_{\text{meta-test}}$	
	Support	**Query**	**Support**	**Query**
Covid	/	/	309	612
Bible	/	/	280	548
Books	/	/	304	637
ECB	/	/	295	573
TED	/	/	390	772
EMEA	14856	29668	456	975
Globalvoices	11686	23319	368	699
JRC	7863	15769	254	519
KDE	24078	48284	756	1510
WMT	10939	21874	334	704

Table 9: Data statistic of the meta-learning stage for English→German translation task

	$\mathcal{D}_{\text{meta-train}}$		$\mathcal{D}_{\text{meta-test}}$	
	Support	**Query**	**Support**	**Query**
Education	/	/	395	785
Microblog	/	/	358	721
Science	/	/	392	852
Subtitles	/	/	612	1219
Laws	6379	13001	197	416
News	9004	18362	281	536
Spoken	18270	36569	571	1148
Thesis	8914	17883	298	547

Table 10: Data statistic of the meta-learning stage for English→Chinese translation task

For the word-level domain-mixing model, we use the same setting as Jiang et al. (2020). The number of parameters of our model is dynamically adjusted with the domain numbers and k times higher than standard model architecture, since every domain has its multi-head attention layer and feed-forward layer.

A.3 Different classifiers

With a general in-domain corpus and some out-of-domain corpora, we train five classifiers. We experiment with two different labeling schemes: `2-labels` where we distinguish only two classes: *out-of-domain* and *in-domain*; `many-labels` where sentences are labeled with the respective domain labels. Further, we experiment with two variants of the BERT model: first, we use monolingual English BERT on the source side only, and second, we use multilingual BERT (mBERT) to classify the parallel sentence pairs. For further comparison, we include also a CNN-based classifier (Kim, 2014). We present the accuracy of the English-German domain classifier in Table 11.

Classifier	Acc(%)
CNN	74.91%
BERT: many-labels	96.12%
BERT: 2-labels	95.35%
mBERT: many-labels	95.41%
mBERT: 2-labels	95.26%

Table 11: The accuracy of the different classifiers.

A.4 Cross-Domain Robustness

Table 12 reports the average difference of $k \times k$ BLEU scores; a larger positive value means a more robust model. We observed that RMLNMT shows its robustness on all domains and that the model performance fine-tuned in one specific domain is not sacrificed in other domains.

In Figure 3 we show the detailed results ($k \times k$ scores). We observed that the plain meta-learning based methods have a negative value, which means the performance gains in the specific domains come at the cost of performance decreases in other domains. In other words, the model is not domain robust enough. In contrast, RMLNMT has a positive difference with the vanilla system, showing that the model is robust.

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

Table 12: The average improvement over vanilla baseline.

A.5 Evaluations

In addition to BLEU, we also use chrF (Popović, 2015) and COMET (Rei et al., 2020) as evaluation metrics. Table 13 and Table 14 show the results. Consistently with the BLEU score (Tables 1 and 2), we observed that RMLNMT is more effective than all previous methods.

	Models	Unseen					Seen				
		Covid	Bible	Books	ECB	TED	EMEA	Globalvoices	JRC	KDE	WMT
1	**Vanilla**	0.550	0.418	0.385	0.538	0.542	0.599	0.536	0.614	0.525	0.558
	Plain FT	0.555	0.423	0.388	0.540	0.548	0.600	0.536	0.618	0.528	0.558
2	**Meta-MT w/o FT**	0.545	0.410	0.382	0.498	0.538	0.532	0.531	0.610	0.464	0.553
	Meta-MT + FT	0.566	0.432	0.390	0.542	0.556	0.582	0.538	0.613	0.522	0.552
3	**Meta-Curriculum (LM) w/o FT**	0.548	0.412	0.384	0.523	0.543	0.560	0.536	0.611	0.521	0.554
	Meta-Curriculum (LM) + FT	0.567	0.434	0.395	0.544	0.548	0.572	0.539	0.615	0.522	0.553
4	**Meta-Curriculum (cls) w/o FT**	0.549	0.414	0.385	0.518	0.546	0.559	0.536	0.609	0.516	0.550
	Meta-Curriculum (cls) + FT	0.558	0.447	0.394	0.547	0.562	0.574	0.540	0.615	0.527	0.553
5	**Word-level Adaptive**	0.560	0.418	0.387	0.557	0.551	**0.662**	0.541	0.705	0.555	0.555
6	**RMLNMT w/o FT**	0.555	0.405	0.388	0.557	0.544	0.656	0.552	0.702	0.574	0.561
	RMLNMT + FT	**0.562**	**0.451**	**0.395**	**0.558**	**0.560**	0.656	**0.552**	**0.702**	**0.574**	**0.561**

Table 13: chrF scores on the English → German translation task.

	Models	Unseen					Seen				
		Covid	Bible	Books	ECB	TED	EMEA	Globalvoices	JRC	KDE	WMT
1	**Vanilla**	0.4967	-0.1250	-0.2225	0.3276	0.3400	0.3096	0.3199	0.5430	0.1836	0.4326
	Plain FT	0.5066	-0.1105	-0.1985	0.3315	0.3553	0.3177	0.3276	0.5492	0.1813	0.4392
2	**Meta-MT w/o FT**	0.4850	-0.1454	-0.2228	0.0953	0.3506	0.0524	0.2985	0.5319	0.1304	0.4137
	Meta-MT + FT	0.5175	-0.0650	-0.1878	0.3466	0.3824	0.2678	0.3189	0.5509	0.1316	0.4161
3	**Meta-Curriculum (LM) w/o FT**	0.4879	-0.1365	-0.2122	0.2568	0.3751	0.1968	0.3273	0.5246	0.0962	0.4206
	Meta-Curriculum (LM) + FT	0.5193	-0.0604	-0.1773	0.3460	0.3729	0.2366	0.3141	0.5430	0.1467	0.4128
4	**Meta-Curriculum (cls) w/o FT**	0.4861	-0.1331	-0.2141	0.2496	0.3637	0.1758	0.3171	0.5193	0.0849	0.4120
	Meta-Curriculum (cls) + FT	0.5163	-0.0763	-0.1757	0.3421	0.3801	0.2435	0.3235	0.5452	0.1564	0.4174
5	**Word-level Adaptive**	0.5070	-0.1408	-0.2149	0.3544	0.3678	**0.4296**	0.3410	0.6838	0.2610	0.4106
6	**RMLNMT w/o FT**	0.4943	-0.1956	-0.2179	0.3580	0.3394	0.4026	0.3769	0.6797	0.3014	0.4255
	RMLNMT + FT	**0.5302**	**-0.0543**	**-0.1610**	**0.3547**	**0.3867**	0.4046	**0.3771**	**0.6797**	**0.3015**	**0.4256**

Table 14: COMET scores on the English → German translation task.

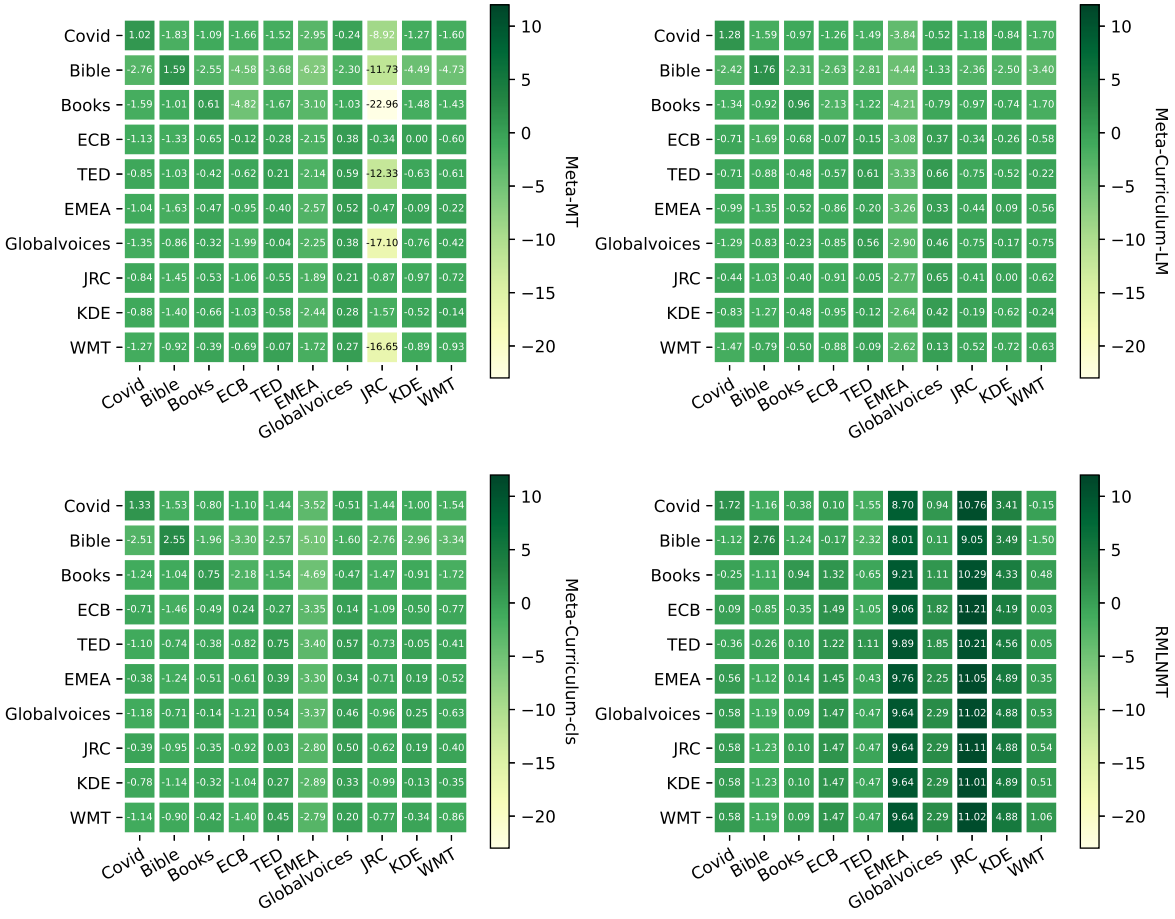

Figure 3: BLEU scores for one specific finetuned model on other domains for en2de translation.