

Multi-Domain Adaptation in Neural Machine Translation with Dynamic Sampling Strategies

First Author

Affiliation / Address line 1
Affiliation / Address line 2
Affiliation / Address line 3
email@domain

Second Author

Affiliation / Address line 1
Affiliation / Address line 2
Affiliation / Address line 3
email@domain

Abstract

Building effective Neural Machine Translation models often implies accommodating diverse sets of heterogeneous data so as to optimize performance for the domain(s) of interest. Such multi-source / multi-domain adaptation problems are typically approached through instance selection or reweighting strategies, based on a static assessment of the relevance of training instances with respect to the task at hand. In this paper, we study dynamic data selection strategies that are able to automatically re-evaluate the usefulness of data samples and to evolve a data selection policy in the course of training. Based on the results of multiple experiments, we show that such methods constitute a generic framework to automatically and effectively handle a variety of real-world situations, from multi-source domain adaptation to multi-domain learning and unsupervised domain adaptation.

1 Introduction

A typical setting in machine translation (MT) is to collect the largest possible collection of parallel data for the chosen language pair, with the intent to achieve optimal performance for the task of interest. In such situations, the training data distribution is opportunistic, while the test data distribution is chosen and fixed; a key component in training is then to mitigate the detrimental effects of a possible mismatch between these distributions. Single-

source and multi-source¹ domain adaptation (DA) is a well-studied instance of this setting (see Chu et al. (2017; Saunders (2021) for a review), and so is multi-domain (MD) learning (Chu and Dabre, 2018; Zeng et al., 2018; Jiang et al., 2020; Pham et al., 2021). A related situation is multilingual machine translation (Firat et al., 2016; Ha et al., 2016; Johnson et al., 2017; Aharoni et al., 2019), where the heterogeneity of training data not only corresponds to variations in the topic, genre, or register but also in language.

This problem is often approached by *static* instance selection or re-weighting strategies, where the available training data is used in proportion to its relevance for the test conditions (Moore and Lewis, 2010; Axelrod et al., 2011). Finding the optimal balance of training data is, however, a challenging task due, for instance, to the similarity between domains/languages, but also due to regularization effects of out-of-domain data (Miceli Barone et al., 2017). It may also be suboptimal, as some target domains or languages might be easier to train than others. Finally, improving the performance of the MT system in one domain will often hurt that of another (van der Wees et al., 2017; Britz et al., 2017), and improving model generalization across all domains (Koehn et al., 2018) may not achieve optimally for any particular domain.

Several recent proposals have explored ways to instead consider *dynamic* data selection and sampling strategies that surpass static strategies. Notably, van der Wees et al. (2017; Zhang et al. (2019) construct a static curriculum, while Graves et al. (2017; Platanios et al. (2019; Kumar et al.

© 2022 The authors. This article is licensed under a Creative Commons 3.0 licence, no derivative works, attribution, CC-BY-ND.

¹In this paper, multi-source DA means having multiple domains to adapt from and should not be confused with multi-source translation, where several *source languages* are considered.

(2019; Wang et al. (2020a; Wang et al. (2020b) build curricula that automatically adapt to the training data.

In this paper, we contribute to this line of research in several ways.

- First, we propose a novel framework (*Multi-Domain Automated Curriculum*, MDAC for short), which is a variant of Differentiable Data Selection (DDS) of Wang et al. (2020b), that simultaneously accounts for the domain adaptation and the multi-domain adaptation problems.
- We demonstrate MDAC’s equivalent performance compared to fine-tuning for domain adaptation; and better performance compared to classical choices of static data sampling strategies.
- We show that our variant MDAC mitigate some failures of DDS in handling multi-domain training.
- We demonstrate various exotic use cases of differentiable data selection frameworks (MDAC and original DDS), including, domain adaptation using unsupervised clustering instead of using domain label a priori which is barely available in real context; domain adaptation using out-of-domain training data and small in-domain validation data; and bi-domain adaptation in which we are only interested in two domain of the training mixture.

Based on these experimental results, our main conclusions are that (a) using MDAC often yields overall performance that is as good as the standard fine-tuning strategy for domain adaptation; (b) MDAC can effectively handle a variety of test situations, from targeting one single domain to the full multi-domain scenario.

2 Learning with multiple data sources

We conventionally define a domain d as a distribution $\mathcal{D}_d(x)$ over some feature space \mathcal{X} that is shared across domains (Pan and Yang, 2010): in machine translation, \mathcal{X} is the representation space for input sentences; each domain corresponds to a specific source of data, and may differ from the other data sources in terms of textual genre, thematic content (Chen et al., 2016; Zhang et al., 2016), register (Sennrich et al., 2016a), style (Niu et al., 2018), etc.

Translation in domain d is formalized by a translation function $h_d(y|x)$ pairing sentences in a source language with sentences in a target language $y \in \mathcal{Y}$. h_d is usually assumed to be deterministic (hence $y = h_d(x)$), but might differ from one domain to the other.

It is usual in MT to opportunistically collect training samples from several domains, which means that training instances are distributed according to a mixture \mathcal{D}^s such that $\mathcal{D}^s(x) = \sum_{d=1}^{n_d} \lambda^s(d) \mathcal{D}_d(x)$, with $\{\lambda^s(d), d = 1 \dots n_d\}$ the mixture weights satisfying $\sum_d \lambda^s(d) = 1$. In the sequel, we also use boldface to denote the vector $\boldsymbol{\lambda}$, and $\lambda(d)$ is used to denote a scalar corresponding to the d^{th} element of the vector $\boldsymbol{\lambda}$.

The main challenge in this situation is to make the best of these heterogeneous data, with the aim to achieve the optimal performance for the intended test conditions. These might correspond to data from just one of the training domains, as in standard supervised (multi-source) domain adaptation; a more difficult case is when the test data is from one domain unseen in training (unseen domain adaptation); in multi-domain adaptation finally, the test distribution is itself a mixture of domains, some of which may also be observed in training. Without loss of generality, one may then assume that the test distribution takes the form $\mathcal{D}^t(x) = \sum_d \lambda^t(d) \mathcal{D}_d(x)$ - with only one non-null component in the case of pure domain adaptation (see illustration on Figure 1).

These situations have been amply documented from a theoretical perspective (eg. Mansour et al. (2009b; Mansour et al. (2009a; Hoffman et al. (2018)). A general recommendation in the DA setting is to adjust the sampling distribution used to optimize the system so as to compensate for the mismatch between $\mathcal{D}^s(x)$ and $\mathcal{D}^t(x)$. This can be approximated by reweighting instances, or more conveniently domains, which are selected during training with a probability $\lambda^l(d)$, with $\lambda^l(d) \neq \lambda^s(d)$.

A widely used approach to supervised DA is *fine-tuning* Luong and Manning (2015; Freitag and Al-Onaizan (2016), where $\boldsymbol{\lambda}^l$ is allowed to vary during learning. With our notations, this approach amounts to first learning an initial parameter value with all the data ($\forall d, \lambda^l(d) = \lambda^s(d)$), then to continue training with only batches from the test domain d_t ($\lambda^l(d) = \mathbb{I}(d = d_t)$) with $\mathbb{I}(A)$ the indicator function for predicate A . Note that this strategy is potentially suboptimal, as some out-of-domain

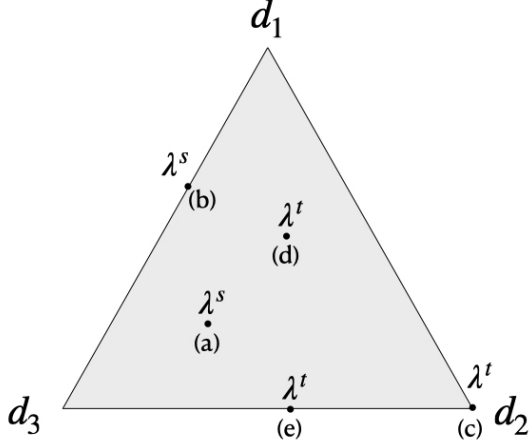


Figure 1: Training and testing with distribution mismatch. We consider just three domains, and represent vectors of mixture weights λ^s and λ^t in the 3-dimensional simplex. Training with weights in (a) and testing with weights in (c) is supervised multi-source domain adaptation to domain 2 (d_2), while (b)-(c) is the unsupervised version, with no training data from d_2 ; training with weights in (a) and testing with weights in (d) is multi-domain learning, also illustrated with configurations (a)-(e) (training domain d_1 is not seen in test), and (b)-(d) (test domain d_2 is unseen in training).

samples may contribute to the final performance due to eg. domain similarity. Optimizing the learning distribution in multi-domain settings is even more challenging, as the learner has to best take advantage of potential domains overlaps, and also of the fact that some domains might be easier to learn than others.

3 Multi-Domain Automated Curriculum

3.1 Basic principles

Assuming training data in each of the n_d domains $d_1 \dots d_{n_d}$, we denote the size of the training corpus from domain d as N_d^s , and $N^s = \sum_d N_d^s$ is the total number of training samples. We use $\widehat{\mathcal{D}}_d^l$ and $\widehat{\mathcal{D}}_d^t$ to denote the empirical train and test distributions for domain d , and $\widehat{\mathcal{D}}^u(x; \lambda^u) = \sum_d \lambda^u(d) \widehat{\mathcal{D}}_d^u(x)$ for $u \in \{l, t\}$. In our setting, λ^t , and hence $\widehat{\mathcal{D}}^t(x; \lambda^t)$ are fixed and predefined, approximated with an equivalent number of development corpora.

MDAC constructs an adaptative training distribution λ^l that optimizes the data selection policy along with the training of the NMT model. We parameterize λ^l by a differentiable function $\lambda^l(\psi)$, which will be described in detail in Section 4.4. We divide the training into many short sessions; in each session t , the model is trained with a static data distribution $\lambda^l(\psi_t)$. After one learning session, we update the data distribution using the REINFORCE

algorithm of Williams (1992). The evolution of ψ is thus defined by:

$$\psi_{t+1} = \psi_t + \text{lr}_1 * \sum_{d=1}^{n_d} R(d) * \frac{\partial \lambda^l(d; \psi_t)}{\partial \psi},$$

where the reward $R(d)$ is computed as:

$$R(d) = J^t(\theta_{t+k}, \lambda^t) - J^t(\theta_t, \lambda^t),$$

and where we also define:

$$\begin{aligned} \theta_{t+i} &= \text{Update}(\theta_{t+i-1}, [x_j^i, y_j^i]_{j=1}^N) \\ x_j^i, y_j^i &\sim \widehat{\mathcal{D}}_d^l(x) \\ J^t(\theta, \lambda^t) &= \sum_{d=1}^{n_d} \lambda^t(d) \sum_{x_d^t, y_d^t \in \widehat{\mathcal{D}}_d^t} l(\theta, x_d^t, y_d^t). \end{aligned}$$

In these equations, N denotes the size of a batch; lr_1 is the learning rate of the sampling distribution; $l(\theta, x, y)$ is the loss of the NMT model on sample (x, y) ; $J^t(\theta, \lambda^t)$ is the weighted loss aggregated over n_d dev-sets corresponding to the n_d domains.

To compute the reward $R(d)$ of using data from domain d to train the model, we simulate k training steps from the current checkpoint by training the model with k batches sampled from $\widehat{\mathcal{D}}_d^l(d)$ and compute the gain of the weighted dev-loss. This computation is inspired by the target prediction gain in Graves et al. (2017). However, where Graves et al. (2017) used accumulated gains from the past as rewards, we instead predict the usefulness of each domain for improving the future performance of the system given its current state. This is achieved by simulating a round of training with only the data from one domain. Furthermore, the choice of the parameterization of the sampling distribution in our method differs from that of Graves et al. (2017), although this choice is empirical.

The work of Wang et al. (2020b) is also related: it is based on the Bi-level Optimization framework, which aims to find an optimal static distribution λ^l that will result in the best NMT model with respect to a given target dev set at the end of training. These authors also derive a similar form of update for ψ . However, their reward is the cosine similarity between the gradient computed with the training data from one domain and the gradient computed with the dev set. We compare this approach with ours in the experiment section.

3.2 MDAC for (multi) domain adaptation

The setting developed in previous sections is quite general and can, in principle, accommodate the variety of situations mentioned above, and many more: basic domain adaption, multi-domain adaptation

with various target distributions, possibly including domains unseen in training. In our experiments, we would like to better assess the true potential of MDAC in these settings and seek to experimentally answer the following questions:

- is MDAC a viable alternative to conventional fine-tuning? In particular, does it enable to better take advantage of relevant data from other domains?
- is MDAC also a viable option in multi-domain adaptation scenarios?
- does MDAC also enable to perform *unsupervised* (multi-)domain adaptation?

These questions are further explored in Section 5. We now turn to our experimental conditions.

4 Experimental settings

4.1 Data and metrics

We experiment with translation from English into French and use texts initially originating from 6 domains, corresponding to the following data sources: the UFAL Medical corpus V1.0 (MED)²; the European Central Bank corpus (BANK) (Tiedemann, 2012); the JRC-Acquis Communautaire corpus (LAW) (Steinberger et al., 2006); documentations for KDE, Ubuntu, GNOME and PHP from the Opus collection (Tiedemann, 2009), merged in a IT-domain; TedTalks (TALK) (Cettolo et al., 2012), and the Koran (REL). Complementary experiments also use v12 of the News Commentary corpus (NEWS). Most corpora are available from the Opus website.³ These corpora were deduplicated and tokenized with in-house tools; statistics are in Table 1. To reduce the number of types and build open-vocabulary systems, we use Byte-Pair Encoding (Sennrich et al., 2016b) with 30,000 merge operations on a corpus containing all sentences in both languages. We randomly select in each corpus a development and a test set of 1,000 lines and keep the rest for training. Validation sets are used to choose the best model according to the average BLEU score (Papineni et al., 2002).⁴ Significance testing is performed using bootstrap resampling (Koehn, 2004), implemented in compare-

²https://ufal.mff.cuni.cz/ufal_medical_corpus. We only use the in-domain (medical) subcorpora: PATR, EMEA, CESTA, ECDC.

³<http://opus.nlpl.eu>

⁴We use truecasing and sacrebleu (Post, 2018).

mt⁵ (Neubig et al., 2019). We report significant differences at the level of $p = 0.05$.

4.2 Baseline systems

Our baselines are standard for multi-domain systems.⁶ Using Transformers (Vaswani et al., 2017) implemented in OpenNMT-tf⁷ (Klein et al., 2017), we build the following systems:

- Generic models trained with predefined mixtures of the training data taking the form:

$$\lambda_\alpha(d) = \frac{q_d^\alpha}{n_d} \quad q_d = \frac{|N_d^s|}{N^s} \quad (1)$$

$$\sum_{d=1} q_d^\alpha$$

with $\alpha \in \{0, 0.25, 0.5, 0.75, 1.0\}$. We denote these as Mixed- α below. Mixed-0 uses a uniform distribution, Mixed-1.0 the empirical distribution of domains.

- fine-tuned models (Luong and Manning, 2015; Freitag and Al-Onaizan, 2016), based on the Mixed-1.0 system, further trained on each domain for at most 50 000 iterations, with early stopping when the dev BLEU stops increasing for 5 successive iterations. The full fine-tuning (FT-Full) procedure may update all the parameters of the initial generic model, resulting in six systems, each adapted for one domain, with no parameter sharing across domains.
- systems trained with fixed data mixtures corresponding to $\lambda^l \in [\lambda_0, \lambda_{0.25}, \lambda_{0.5}, \lambda_{0.75}, \lambda_{1.0}]$; these are used in the multi-domain experiments of Section 5.3;
- our own implementations of recent dynamic sampling proposals from the literature: Curriculum Learning (CL) of Zhang et al. (2019) and Differential Data Selection (DDS) of Wang et al. (2020b) (see details below);

All models use embeddings and hidden layers of dimension 512. Transformer models contain 8 attention heads in each of the 6+6 layers; the inner feedforward layer contains 2048 cells. Training lasts for 200K iterations, with batches of 12,288 tokens, Adam with parameters $\beta_1 = 0.9$, $\beta_2 =$

⁵<https://github.com/neulab/compare-mt>

⁶We however omit domain-specific systems trained only with the corresponding subset of the data, which are always inferior to the mix-domain strategy (Britz et al., 2017).

⁷<https://github.com/OpenNMT/OpenNMT-tf>

	MED	LAW	BANK	IT	TALK	REL	NEWS
# lines	2609 (0.68)	501 (0.13)	190 (0.05)	270 (0.07)	160 (0.04)	130 (0.03)	260 (0)
# tokens	133 / 154	17.1 / 19.6	6.3 / 7.3	3.6 / 4.6	3.6 / 4.0	3.2 / 3.4	7.8 / 9.2
# types	771 / 720	52.7 / 63.1	92.3 / 94.7	75.8 / 91.4	61.5 / 73.3	22.4 / 10.5	-
# uniq	700 / 640	20.2 / 23.7	42.9 / 40.1	44.7 / 55.7	20.7 / 25.6	7.1 / 2.1	-

Table 1: Corpora statistics: number of parallel lines ($\times 10^3$) and proportion in the training domain mixture (which does not contain `NEWS`), number English and French tokens ($\times 10^6$), number English and French types ($\times 10^3$), number of types that only appear in a given domain ($\times 10^3$). `MED` is the largest domain, containing almost 70% of the sentences, while `REL` is the smallest, with only 3% of the data.

0.98, Noam decay ($warmup_steps = 4000$), and a dropout rate of 0.1 in all layers.

4.3 CL and DDS’s re-implementation

We re-implement DDS in Tensorflow without any change in the choices of parameterization and hyper-parameters compared to the original code of Wang et al. (2020b).⁸ We also re-implement the approach of Zhang et al. (2019) according to the authors’ description. For each domain adaptation experiment, we combine the training data of all other domains into one corpus then compute the cross-entropy difference score of each source sentence of this combined dataset. We then sort and split the corpus into 9 shards and execute curriculum learning with 10 shards, using the in-domain corpus as the first shard.

4.4 MDAC systems

The behavior of MDAC only depends on (a) the initial domain distribution at the start of training $\lambda_{t=0}^l$, and (b) the targeted (dev/test) distribution λ^t . We thus report these systems as MDAC ($\lambda_{t=0}^l, \lambda^t$) and compare with DDS using the same settings.

In our work, we parameterize the distribution λ^l as follows (with $\beta = 2$ in all experiments):⁹

$$\lambda^l(d; \psi) = \frac{\psi[d]^\beta}{\sum_i \psi[i]^\beta}.$$

This parameterization avoids the “rich-get-richer” effect that we observe when using $\lambda(\psi) = \text{softmax}(\psi)$, which yields gradients wrt. $\psi[d]$ that are proportional to $\exp(\psi[d])$ (see also Figure 2). Additional settings for the hyper-parameters of our method include the number of simulation steps $k = 10$ and the learning rate $\text{lr}_{data} = 0.001$. We update the sampling distribution via 100 gradient descent iterations for almost all experimental

settings except that for adaptation with automatic clusters (Section 5.5), where we use 20 gradient descent iterations to avoid converging to degenerate distributions. We split the training into 100 short sessions that last 2000 training steps each. The choice of those hyper-parameters is mostly heuristic except for the learning rate lr_{data} which is optimized via grid search over a set of values $\{0.001, 0.0025, 0.005\}$.

The computational cost of our approach is due to the simulation step, which is conducted after every 2000 iterations to compute the reward of each domain. During the simulation step, we update the temporary checkpoint with k updates for each domain, which cost as much as k training updates. Therefore, we execute $k \times n_d$ updates in total after every 2000 iterations. Our algorithm approximately costs $1 + \frac{k \times n_d}{2000}$ times as much as a standard training.

4.5 Experimental tasks

We evaluate and compare our method with multiple baselines in the 5 following conditions. In the *supervised domain adaptation task*, given the data from 6 domains (`MED`, `BANK`, `LAW`, `IT`, `TALK`, `REL`), we aim to build distinct expert NMT models for each domain. To challenge the flexibility of the method, we also consider a *bi-domain adaptation task*, where given the same 6 domains, we only focus on adapting to a mixture of 2 domains.

In the *multi-domain adaptation task*, given the same 6 domains, we aim to build one single NMT model that would perform optimally, assuming a uniform distribution of domains during the test.

In a fourth experiment (*unsupervised domain adaptation*), given training data in 6 domains and a small development set of a new domain (`NEWS` in our case), we aim to build an NMT model which performs well for the unseen domain.

Finally, in the *unsupervised domain adaptation task*, we cluster all available training data into 30 clusters using the KNN algorithm in the same

⁸<https://github.com/cindyxyinyiawang/multiDDS>

⁹This setting corresponds to the *spherical softmax* of (de Brébisson and Vincent, 2016).

way as in (Tars and Fishel, 2018), then adapt these clusters to one of 6 domains using the corresponding in-domain dev set. We compare MDAC to DDS for each of our 6 test sets.

5 Results and discussion

5.1 Domain Adaptation

In this setting, we aim to build an NMT model for one single domain: we accordingly set λ^t to a deterministic distribution λ_d , where the target domain d has probability 1.

We consider three initializations for MDAC and DDS, using λ_0 , λ_1 and λ_d . According to Table 2, MDAC achieves the overall best performance when $\lambda_{t=0} = \lambda_0$. Doing so proves much better than initializing with λ_d for small domains: `TALK`, `BANK` and `IT`. Conversely, initializing with λ_d is beneficial when targeting large domains such as `MED` and `LAW`. The same conclusion holds for DDS.

We now compare the best MDAC system (using $\lambda_{t=0} = \lambda_0$) to full fine-tuning. According to Table 2, fine-tuning is better for large domains such as `MED` and `LAW`, while MDAC outperforms fine-tuning by approximately 1.2 BLEU for `BANK` and 1.0 BLEU for `REL`. This indicates that for small domains, out-of-domain data helps improve the generalization and that MDAC is able to exploit both the in-domain and the out-of-domain training data instead of edging out the out-of-domain training data as in fine-tuning. Results for DDS display similar trends but are always outperformed by MDAC. The behavior of CL, which does only well the large domain `MED` lag somewhat behind.

5.2 Bi-domain adaptation

In these control experiments, we showcase the flexibility of dynamic sampling and try to adapt to (arbitrary) pairs of target domains with equal weight and contrast our results with those of DDS. Results are in Table 3. Here, MDAC significantly outperforms DDS in two settings (`MED+IT` and `LAW+BANK`) while being surpassed in `TALK+REL`.

5.3 Multi-domain adaptation

We now turn to a more realistic scenario and consider multi-domain adaptation, which aims to train one single system with optimal performance averaged over 6 domains. This setting targets a uniform test distribution $\lambda^t = \lambda_0$. In this situation, CL (Zhang et al., 2019) does not apply. We therefore only contrast the performance of MDAC,

DDS and several fixed training data distribution $\lambda^l \in [\lambda_0, \lambda_{0.25}, \lambda_{0.5}, \lambda_{0.75}, \lambda_{1.0}]$, where λ_α is defined according to equation (1).

We again initialize MDAC and DDS with two distribution λ_0 and λ_1 . According to Table 4, MDAC achieves the best performance with initial (uniform) λ_0 . The same conclusion holds for DDS. For this configuration, MDAC outperforms static training distributions including $[\lambda_0, \lambda_{0.75}, \lambda_{1.0}]$ by a significant margin, and performs slightly better than $[\lambda_{0.25}, \lambda_{0.5}]$. This indicates that MDAC allows us to skip the somewhat heuristic choice of the optimal training mixture.

A second observation is that MDAC again outperforms DDS by a wide margin (+1.5 BLEU on average); the only domain where DDS does better is `MED`. Figure 2, which displays the evolution of the training mixture in the course of training, helps to understand the difference between the two methods. For DDS (Figure 2a), the sampling distribution quickly reaches a bi-domain mode in which only `MED` and `REL` have significant probability – hence the good performance on the former domain. In contrast, the distribution computed by MDAC evolves more smoothly; small domains such as `BANK`, `IT`, `TALK` and `REL` receive a larger proportion of training data in the early stages; their weights then slowly decrease as larger domains such as `MED` and `LAW` increase their share. This, however, only happens at the end of training, when some NMT models might already be close to their best performance for the small domains.

5.4 Unseen domain

The left part of Table 5 displays the performance on the unseen domain `NEWS` for NMT systems trained with the mixtures $\lambda^l \in [\lambda_0, \lambda_{0.25}, \lambda_{0.5}, \lambda_{0.75}, \lambda_{1.0}]$ and with dynamic data selection (MDAC and DDS). These systems have insignificant differences in BLEU, showing that dynamic mixture does not improve the robustness of the NMT system against unseen domains. However, the fact that the performance of MDAC and DDS is close to the best performance suggests that they also apply in such challenging situations.

5.5 Automatic clustering

The right part of Table 5 reports the performance of NMT systems adapted to each domain.¹⁰ In comparison to Section 5.1, the training data is dis-

¹⁰A more detailed analysis is in the supplementary material.

domain $d =$	MED	LAW	BANK	TALK	IT	REL	avg.
FT-Full(d)	40.3	63.8	54.4	38.5	52.0	91.0	56.7
CL (d)	40.2	60.2	53.7	36.5	51.1	91.1	55.5
DDS (λ_0, λ_d)	39.6	60.1	55.0	38.5	52.5	92.0	56.3
MDAC (λ_0, λ_d)	39.6	62.5**	55.6*	38.5	52.4	92***	56.8
DDS (λ_1, λ_d)	39.7	53.9	49.6	37.9	43.1	64.3	48.1
MDAC (λ_1, λ_d)	40.2	59.9	52.6	38.5	50.7	79.8	53.6
DDS (λ_d, λ_d)	39.9	63.9	54.5	35.4	51.2	91.8	56.1
MDAC (λ_d, λ_d)	40.6	63.9	54.5	35.6	51.3	92.3	56.4

Table 2: Domain adaptation experiments. We report BLEU scores of each method for 6 target domains and their average: each column corresponds to a distinct system. (*) MDAC is significantly better than CL, fine-tuning and DDS with $p < 0.05$. (**) MDAC is significantly better than CL and DDS with $p < 0.05$. (***) MDAC is significantly better than CL, fine-tuning with $p < 0.05$.

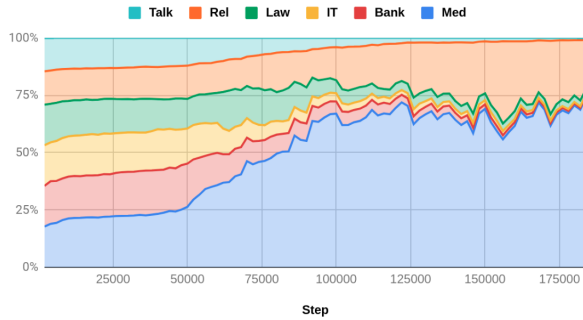
domain $d =$	MED	LAW	BANK	TALK	IT	REL
DDS (λ_0, λ_2)	39.5	-	-	-	50.1	-
MDAC (λ_0, λ_2)	39.1	-	-	-	51.8*	-
DDS (λ_0, λ_2)	-	60.8	53.3	-	-	-
MDAC (λ_0, λ_2)	-	61.9*	54.5*	-	-	-
DDS (λ_0, λ_2)	-	-	-	37.9	-	91.3
MDAC (λ_0, λ_2)	-	-	-	36.9	-	90.4

Table 3: Adapting to two domains. For a given line, non empty columns correspond to the pair of target domains. (*) MDAC is significantly better than DDS with $p < 0.05$.

domain $d =$	MED	LAW	BANK	TALK	IT	REL	mean
Mixed-0	38.6	59.3	53.7	37.3	51.0	90.4	55.1
Mixed-0.25	38.9	59.6	53.3	37.6	50.5	90.6	55.1
Mixed-0.5	39.0	60.2	52.5	38.5	51.9	90.3	55.4
Mixed-0.75	39.4	59.9	51.9	38.8	50.0	87.6	54.6
Mixed-1	40.3	59.5	49.8	36.4	49.0	80.0	52.5
DDS (λ_0, λ_0)	40.1	56.9	50.7	37.4	46.8	92.0	54.0
MDAC (λ_0, λ_0)	38.5	60.3**	54.4*	37.3	51.3**	91.4*	55.5**
DDS (λ_1, λ_0)	40.6	55.5	48.0	36.2	46.9	60.1	47.9
MDAC (λ_1, λ_0)	40.2	59.3**	51.0**	36.9**	48.6**	80.7**	52.8**

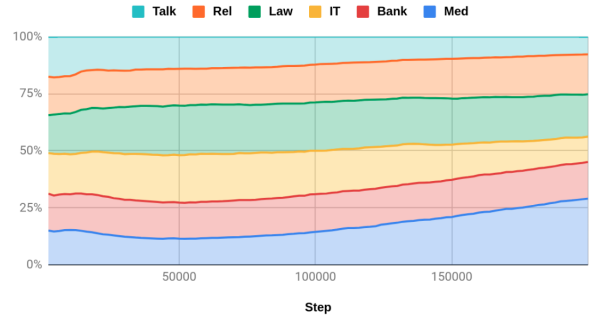
Table 4: Multi domain adaptation. For a given line, all the columns correspond to the same multi-domain system. (*) MDAC is significantly better than Mixed- α with $p < 0.05$. (**) MDAC is significantly better than DDS with $p < 0.05$.

DDS



(a) DDS

MDAC



(b) MDAC

Figure 2: Evolution of the sampling distribution during training.

domain	$d =$	NEWS
<i>Unseen domain</i>		
Mixed-0		25.7
Mixed-0.25		25.8
Mixed-0.5		26.5
Mixed-0.75		26.8
Mixed-1		26.9
DDS ($\lambda_0, \lambda_{news}$)		26.3
MDAC ($\lambda_0, \lambda_{news}$)		26.3

domain	$d =$	MED	LAW	BANK	TALK	IT	REL	mean
<i>Training with 30 clusters</i>								
DDS (λ^*, λ_d)		38.3	60.1	50.3	35.8	49.1	90.1	53.9
MDAC (λ^*, λ_d)		39.2*	61.6*	52.0*	38.2*	49.1	89.7	55.0*

Table 5: Unseen domain adaptation (left) and automatic clustering adaptation (right). For a given line, each column corresponds to one distinct system. (*) MDAC is significantly better than DDS.

tributed in 30 automatic clusters instead of the 6 original domains. Splitting the train data into small groups provides the learner with extra degrees of freedom when selecting the best distribution. However, as these clusters are built automatically, they are noisier in nature. According to the results in Table 5, this scenario is hard both for DDS and MDAC, which performs much worse than for the supervised DA setting. This again signals the importance of initialization: analyzing the clustering, we find that the data for REL mostly correspond to one single cluster. When using a uniform initialization, this cluster starts with a very small weight and never succeeds in matching the (good) performance observed in the DA setting.

6 Related Work

Domain adaptation is an old problem that has been studied from many angles, both for SMT and NMT. A survey of supervised and unsupervised DA for NMT is in Chu and Wang (2018), where the authors distinguish between data-centric and model-centric DA, a view also adopted in the more recent survey of Saunders (2021). Our approach to DA in this paper falls under the former category. We refer interested readers to these references and do not discuss DA any further.

Multi-domain adaptation (MDA) aims to develop systems that simultaneously bode well for several domains. Like for DA, techniques for supervised MDA combine one or several ingredients: (a) the specialization of data representations (Kobus et al., 2017) or of sub-networks (Pham et al., 2019) to differentiate the processing of each domain; (b) the use of adversarial techniques to neutralize differences between domains (Britz et al., 2017; Zeng et al., 2018); (c) the use of automatic domain identification e.g. Jiang et al. (2020). Unsupervised MDA is studied in Farajian et al. (2017), as an instance of unsupervised DA.

Most approaches to adaptive/dynamic data se-

lection take inspiration from Bengio et al. (2009), where the notion of curriculum learning is introduced. CL relies on the notion of the “easiness” of a sample to schedule the presentation of training data so that the easiest examples are presented first and the hardest last. Various ways to automate CL using the framework of multi-armed bandits are explored in (Graves et al., 2017), which has been an inspiration for our implementation. While the initial aim was primarily to improve and speed up training, CL has also proven useful for domain adaptive / multi-domain / multilingual MT, based on alternative definitions of “easiness”. For instance, Zhang et al. (2019) study supervised DA and propose a curriculum approach which progressively augments the training data: in the early stages, only in-data is used, while shards containing less relevant¹¹ data are introduced in later stages. This is somehow opposite to the recommendations of van der Wees et al. (2017), whose *gradual fine-tuning* progressively focuses on the in-domain data.

Kumar et al. (2019) use reinforcement techniques (deep Q-learning) to learn the curriculum strategy: in this work, complexity corresponds to difficulty levels which are binned using contrastive data selection. The reward is based on the increase of the development set loss that results from the actual data selection strategy. The same technique is recently applied to multilingual NMT in (Kumar et al., 2021). Zhou et al. (2020) propose another curriculum-based approach which instead relies on *instance uncertainty* as a measure of their difficulty and presents the data sample starting with the easiest (more predictable) first. Another contribution of this paper is an alternative stopping criterium. Closest to our problems, Wang et al. (2020a) adapt CL for multi-domain adaptation, where an optimal instance weighting scheme is found using Bayesian optimization techniques. Each step consists of (a)

¹¹Domain distance is computed with Lewis-Moore scores (based on the cross-entropy of in-domain LM).

weighting instances based on relevance features, (b) fine-tuning a pretrained model using the weighted training set, and is applied iteratively to train a sequence of models. The one that maximizes the development set performance is finally retained.

7 Conclusion and outlook

In this study, we have presented a generic framework to perform a variety of adaptation tasks for machine translation, ranging from conventional supervised domain adaptation to multi-domain adaptation and unseen domain adaptation. By experimenting with all these settings, we have shown that the same algorithm, aimed at automatically finding an effective data sampling scheme during the course of training, could be used in all these situations. This algorithm, we believe, provides us with a more sound approach to (multi-domain) DA than existing heuristics and dispenses with the costly search of optimal meta-parameters. Another contribution of our work is an experimental comparison of recent approaches to dynamic data selection.

In the future, we intend to continue developing this approach and improve its effectiveness. One issue that we have left unaddressed is reward normalization, which is especially important in the early stages of training (Kumar et al., 2019). Another area where we need to progress is the unsupervised learning setting of Section 5.5, where our results still lag way behind that of supervised DA - this might be due to the inability of our simplistic optimization strategy to handle situations where the number of clusters is too large.

References

- Aharoni, Roei, Melvin Johnson, and Orhan Firat. 2019. Massively multilingual 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 3874–3884, Minneapolis, Minnesota, June. Association for Computational Linguistics.
- Axelrod, Amittai, Xiaodong He, and Jianfeng Gao. 2011. Domain adaptation via pseudo in-domain data selection. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP '11*, pages 355–362, Edinburgh, United Kingdom.
- Bengio, Yoshua, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning, ICML '09*, page 41–48, New York, NY, USA.
- Britz, Denny, Quoc Le, and Reid Pryzant. 2017. Effective domain mixing for neural machine translation. In *Proceedings of the Second Conference on Machine Translation*, pages 118–126, Copenhagen, Denmark. Association for Computational Linguistics.
- Cettolo, Mauro, Christian Girardi, and Marcello Federico. 2012. Wit³: Web inventory of transcribed and translated talks. In *Proceedings of the 16th Conference of the European Association for Machine Translation (EAMT)*, pages 261–268, Trento, Italy, May.
- Chen, Wenhui, Evgeny Matusov, Shahram Khadivi, and Jan-Thorsten Peter. 2016. Guided alignment training for topic-aware neural machine translation. In *Proceedings of the Twelfth Biennial Conference of the Association for Machine Translation in the Americas, AMTA 2012*, Austin, Texas.
- Chu, Chenhui and Raj Dabre. 2018. Multilingual and multi-domain adaptation for neural machine translation. In *Proceedings of the 24th Annual Meeting of the Association for Natural Language Processing, NLP 2018*, pages 909–912, Okayama, Japan.
- Chu, Chenhui and Rui Wang. 2018. A survey of domain adaptation for neural machine translation. In *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018*, pages 1304–1319, Santa Fe, New Mexico, USA.
- Chu, Chenhui, Raj Dabre, and Sadao Kurohashi. 2017. An empirical comparison of domain adaptation methods for neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, ACL 2017, pages 385–391, Vancouver, Canada.
- de Brébisson, Alexandre and Pascal Vincent. 2016. An exploration of softmax alternatives belonging to the spherical loss family. In Bengio, Yoshua and Yann LeCun, editors, *Proceedings of the 4th International Conference on Learning Representation, ICLR*, San Juan, Puerto Rico.
- Farajian, M. Amin, Marco Turchi, Matteo Negri, and Marcello Federico. 2017. Multi-domain neural machine translation through unsupervised adaptation. In *Proceedings of the Second Conference on Machine Translation*, pages 127–137, Copenhagen, Denmark, September.

- Firat, Orhan, Kyunghyun Cho, and Yoshua Bengio. 2016. Multi-way, multilingual neural machine translation with a shared attention mechanism. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 866–875. Association for Computational Linguistics.
- Freitag, Markus and Yaser Al-Onaizan. 2016. Fast domain adaptation for neural machine translation. *CoRR*, abs/1612.06897.
- Graves, Alex, Marc G. Bellemare, Jacob Menick, Rémi Munos, and Koray Kavukcuoglu. 2017. Automated curriculum learning for neural networks. In Precup, Doina and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1311–1320. PMLR, 06–11 Aug.
- Ha, Thanh-He, Jan Niehues, and Alex Waibel. 2016. Toward multilingual neural machine translation with universal encoder and decoder. In *Proceedings of the 13th International Workshop on Spoken Language Translation*, IWSLT 2016, Vancouver, Canada.
- Hoffman, Judy, Mehryar Mohri, and Ningshan Zhang. 2018. Algorithms and theory for multiple-source adaptation. In Bengio, S., H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 8246–8256. Curran Associates, Inc.
- Jiang, Haoming, 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, July. Association for Computational Linguistics.
- Johnson, Melvin, Mike Schuster, Quoc Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google’s multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.
- Klein, Guillaume, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, Vancouver, Canada. Association for Computational Linguistics.
- Kobus, Catherine, Josep Crego, and Jean Senellart. 2017. Domain control for neural machine translation. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 372–378, Varna, Bulgaria.
- Koehn, Philipp, Huda Khayrallah, Kenneth Heafield, and Mikel L. Forcada. 2018. Findings of the WMT 2018 shared task on parallel corpus filtering. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 726–739, Belgium, Brussels, October. Association for Computational Linguistics.
- Koehn, Philipp. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388–395, Barcelona, Spain, July. Association for Computational Linguistics.
- Kumar, Gaurav, George Foster, Colin Cherry, and Maxim Krikun. 2019. Reinforcement learning based curriculum optimization for 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 2054–2061, Minneapolis, Minnesota, June. Association for Computational Linguistics.
- Kumar, Gaurav, Philipp Koehn, and Sanjeev Khudanpur. 2021. Learning policies for multilingual training of neural machine translation systems. *CoRR*, abs/2103.06964.
- Luong, Minh-Thang and Christopher D. Manning. 2015. Stanford neural machine translation systems for spoken language domain. In *Proceedings of the International Workshop on Spoken Language Translation*, IWSLT, Da Nang, Vietnam.
- Mansour, Yishay, Mehryar Mohri, and Afshin Rostamizadeh. 2009a. Domain adaptation with multiple sources. In Koller, D., D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 1041–1048. Curran Associates, Inc.
- Mansour, Yishay, Mehryar Mohri, and Afshin Rostamizadeh. 2009b. Multiple source adaptation and the rényi divergence. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence, UAI 2009*, pages 367–374, December.
- Miceli Barone, Antonio Valerio, Barry Haddow, Ulrich Germann, and Rico Sennrich. 2017. Regularization techniques for fine-tuning in neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1489–1494, Copenhagen, Denmark, September. Association for Computational Linguistics.
- Moore, Robert C. 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.

- Neubig, Graham, Zi-Yi Dou, Junjie Hu, Paul Michel, Danish Pruthi, Xinyi Wang, and John Wieting. 2019. compare-mt: A tool for holistic comparison of language generation systems. *CoRR*, abs/1903.07926.
- Niu, Xing, Sudha Rao, and Marine Carpuat. 2018. Multi-task neural models for translating between styles within and across languages. In Bender, Emily M., Leon Derczynski, and Pierre Isabelle, editors, *Proceedings of the 27th International Conference on Computational Linguistics, COLING*, pages 1008–1021, Santa Fe, New Mexico, USA.
- Pan, Sinno Jialin and Qiang Yang. 2010. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359.
- Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, ACL '02*, pages 311–318, Stroudsburg, PA, USA.
- Pham, Minh Quang, Josep-Maria Crego, Jean Senellart, and François Yvon. 2019. Generic and Specialized Word Embeddings for Multi-Domain Machine Translation. In *Proceedings of the 16th International Workshop on Spoken Language Translation, IWSLT*, page 9p, Hong-Kong, CN.
- Pham, Minh Quang, Josep Crego, and François Yvon. 2021. Revisiting multi-domain machine translation. *Transactions of the Association for Computational Linguistics*, 9(0):17–35.
- Platanios, Emmanouil Antonios, Otilia Stretcu, Graham Neubig, Barnabas Poczos, and Tom Mitchell. 2019. Competence-based curriculum learning for 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 1162–1172, Minneapolis, Minnesota, June. Association for Computational Linguistics.
- Post, Matt. 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, October. Association for Computational Linguistics.
- Saunders, Danielle. 2021. Domain adaptation and multi-domain adaptation for neural machine translation: A survey. *CoRR*, abs/2104.06951.
- Sennrich, Rico, Barry Haddow, and Alexandra Birch. 2016a. Controlling politeness in neural machine translation via side constraints. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 35–40, San Diego, California. Association for Computational Linguistics.
- Sennrich, Rico, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany, August.
- Steinberger, Ralf, Bruno Pouliquen, Anna Widiger, Camelia Ignat, Tomaž Erjavec, Dan Tufis, and Dániel Varga. 2006. The JRC-Acquis: A multilingual aligned parallel corpus with 20+ languages. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation, LREC'06*, Genoa, Italy, may. European Language Resources Association (ELRA).
- Tars, Sander and Mark Fishel. 2018. Multi-domain neural machine translation. In Pérez-Ortiz, Juan Antonio, Felipe Sánchez-Martínez, Miquel Esplà-Gomis, Maja Popović, Celia Rico, André Martins, Joachim Van den Bogaert, and Mikel L. Forcada, editors, *Proceedings of the 21st Annual Conference of the European Association for Machine Translation, EAMT*, pages 259–269, Alicante, Spain. EAMT.
- Tiedemann, Jörg. 2009. News from OPUS - A collection of multilingual parallel corpora with tools and interfaces. In Nicolov, N., K. Bontcheva, G. Angelova, and R. Mitkov, editors, *Recent Advances in Natural Language Processing*, volume V, pages 237–248. John Benjamins, Amsterdam/Philadelphia, Borovets, Bulgaria.
- Tiedemann, Jörg. 2012. Parallel data, tools and interfaces in OPUS. In Chair), Nicoletta Calzolari (Conference, Khalid Choukri, Thierry Declerck, Mehmet Ugur Dogan, Bente Maegaard, Joseph Mariani, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Eight International Conference on Language Resources and Evaluation, LREC'12*, Istanbul, Turkey, may. European Language Resources Association (ELRA).
- van der Wees, Marlies, Arianna Bisazza, and Christof Monz. 2017. Dynamic data selection for neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1400–1410, Copenhagen, Denmark, September. Association for Computational Linguistics.
- Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In Guyon, I., U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Wang, Wei, Ye Tian, Jiquan Ngiam, Yinfei Yang, Isaac Caswell, and Zarana Parekh. 2020a. Learning a multi-domain curriculum for neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7711–7723, Online, July. Association for Computational Linguistics.

Wang, Xinyi, Yulia Tsvetkov, and Graham Neubig. 2020b. Balancing training for multilingual neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8526–8537, Online, July. Association for Computational Linguistics.

Williams, Ronald J. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256.

Zeng, Jiali, 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.

Zhang, Jian, Liangyou Li, Andy Way, and Qun Liu. 2016. Topic-informed neural machine translation. In *Proceedings of the 26th International Conference on Computational Linguistics: Technical Papers*, COLING 2016, pages 1807–1817, Osaka, Japan. The COLING 2016 Organizing Committee.

Zhang, Xuan, 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, June. Association for Computational Linguistics.

Zhou, Yikai, Baosong Yang, Derek F. Wong, Yu Wan, and Lidia S. Chao. 2020. Uncertainty-aware curriculum learning for neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6934–6944, Online, July. Association for Computational Linguistics.