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

Minh-Quang Pham
Uni. Paris-Saclay, CNRS, LISN
F-91405 Orsay
pham@limsi.fr

Josep Crego
SYSTRAN ,
5 rue Feydeau, F-75002 Paris
crego@systrangroup.com

François Yvon
Uni. Paris-Saclay, CNRS, LISN
F-91405 Orsay, France
yvon@limsi.fr

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 in the course of training. Based on the results of multiple experiments, we show that our method offer a generic framework to automatically handle several real-world situations, from multi-source or unsupervised domain adaptation to multidomain learning.

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 aspect of training is then to mitigate the detrimental effects of a mismatch between these distributions. Single-source and multi-source[1] 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 MT (Firat et al., 2016; Ha et al., 2016; Johnson et al., 2017; Aharoni et al., 2019), where the diversity 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 testing 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, or to the regularization effects of out-of-domain data (Miceli Barone et al., 2017). A static policy may also be suboptimal when some target domains or languages are 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 explore ways to instead consider *dynamic* data selection and sampling strategies: van der Wees et al. (2017) and Zhang et al. (2019) construct a static curriculum, while Wang et al. (2020a) and 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), a variant of Differentiable Data Selec-

[1] In this paper, multi-source DA means having multiple domains to adapt from; this setting differs from multi-source translation, where several *source languages* are considered.

tion (DDS) of Wang et al. (2020b), initially applied to multilingual NMT, that simultaneously accounts for the domain adaptation and the multidomain adaptation problems.

- We show that MDAC achieves performance that compare to fine-tuning strategies for DA (§ 5.1) and outperform some static data sampling strategies for multidomain settings (5.3).
- We show that our variant MDAC mitigates some failures of DDS in multidomain training.
- We illustrate the generality of differentiable data selection frameworks (both MDAC and DDS) on less common situations such as DA using unsupervised clustering (§ 5.5); DA using out-of-domain training data and small in-domain validation data (§ 5.4); and two-domain adaptation where the test distribution only mixes two of the training domain (§ 5.2).

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 other data sources in terms of textual genre, thematic content (Chen et al., 2016; Zhang et al., 2016), register, 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 may differ across domains.

It is usual in MT to opportunistically collect corpora 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, boldface $\boldsymbol{\lambda}$ denotes a vector with $\lambda(d)$ the d^{th} component of $\boldsymbol{\lambda}$.

The main challenge in this situation is to make the best of heterogeneous data, with the aim to achieve optimal performance for the target test conditions. These might correspond to data from just one of the training domains, as in standard supervised domain adaptation; a more difficult case is when the test data is from one domain unseen in training (unseen domain adaptation); in multidomain adaptation finally, the test distribution is itself a mixture of domains, some of which may also be observed in training. We thus 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 domain adaptation (see Figure 1).

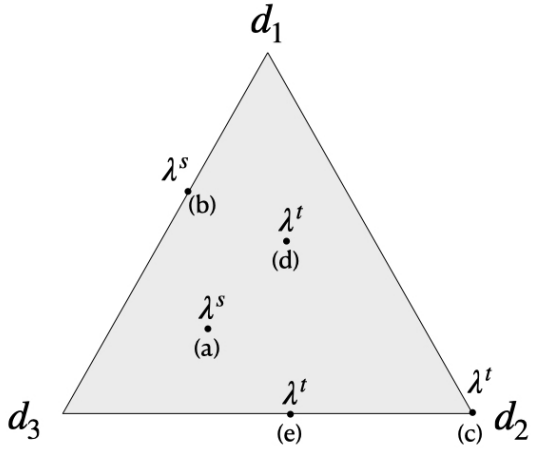


Figure 1: Training and testing with distribution mismatch. We consider three domains and represent $\boldsymbol{\lambda}^s$ and $\boldsymbol{\lambda}^t$ in the 3-dimensional simplex. Training with weights in (a) and testing with weights in (c) is supervised multi-source domain adaptation to domain 2 (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 settings (a)-(e) (training domain d_1 is not seen in test), and (b)-(d) (test domain d_2 is unseen in training).

These situations have been amply documented from a theoretical perspective (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), where $\boldsymbol{\lambda}^l$ varies during learning. With our notations, this approach first learns an initial parameter value with all the data ($\forall d, \lambda^l(d) = \lambda^s(d)$), then continues 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. This strategy is potentially suboptimal as some out-of-domain samples may contribute to the final performance due to e.g. domain similarity. Optimizing the learning distribution in multidomain settings is even more challenging as the learner needs to take advantage of possible 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 $d_1 \dots d_{n_d}$, the size of the training corpus in domain d is denoted N^s_d, and $N^s = \sum_d N^s_d$ is the total number of training samples. $\widehat{\mathcal{D}^l_d}$ and $\widehat{\mathcal{D}^t_d}$ denote the empirical train and test distributions for domain d and $\widehat{\mathcal{D}^u}(x; \boldsymbol{\lambda}^u) = \sum_d \lambda^u(d)\widehat{\mathcal{D}^u_d}(x)$ for $u \in \{l, t\}$. In our setting, $\boldsymbol{\lambda}^t$ and hence $\widehat{\mathcal{D}^t}(x; \boldsymbol{\lambda}^t)$ are fixed and predefined, approximated with an equivalent number of development corpora.

MDAC builds an adaptative training distribution $\boldsymbol{\lambda}^l$ that optimizes the data selection policy along with the training of the model. We parameterize $\boldsymbol{\lambda}^l$ by a differentiable function $\boldsymbol{\lambda}^l(\psi)$, which is described in § 4.4. We divide the training into many short sessions; in each session t, the model is trained with a static data distribution $\boldsymbol{\lambda}^l(\psi_t)$. After one learning session, we update the data distribution using the REINFORCE algorithm of Williams (1992). The evolution of ψ is thus defined by:

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

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

$$R(d) = J^t(\theta_{t+k}, \boldsymbol{\lambda}^t) - J^t(\theta_t, \boldsymbol{\lambda}^t), \qquad (1)$$

and where we also define:

$$\begin{aligned} \theta_{t+i} &= Update\big(\theta_{t+i-1}, [x^i_j, y^i_j]^N_{j=1}\big) \\ x^i_j, y^i_j &\sim \widehat{D^l_d}(x) \\ J^t(\theta, \boldsymbol{\lambda}^t) &= \sum_{d=1}^{n_d} \lambda^t(d) \sum_{x^t_d, y^t_d \in \widehat{D^t_d}} l(\theta, x^t_d, y^t_d). \end{aligned}$$

In these equations, N denotes the size of a batch; $\mathbf{lr}_1$ is the learning rate of the sampling distribution; $l(\theta, x, y)$ is the loss of the NMT model on sample (x, y); $J^t(\theta, \boldsymbol{\lambda}^t)$ is the weighted loss aggregated over n_d dev-sets corresponding to the n_d domains.

To compute the reward $R(d)$ associated to training the model with data from domain d, we simulate k training steps from the current checkpoint, using k batches sampled from $D^l(d)$ and computing the gain of the weighted dev-loss. This computation is inspired by the target prediction gain of 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. We also differ from these authors in the parameterization of the sampling distribution.

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 $\boldsymbol{\lambda}^l$ that will result in the best 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 DA, multidomain adaptation with various target distributions, possibly including domains unseen in training. In our experiments, we would like to better assess the potential of MDAC in these settings and seek to study the following questions:

- is MDAC a viable alternative to fine-tuning? In particular, does it enable to better take advantage of relevant data from other domains?
- is MDAC a viable option in multidomain adaptation scenarios?
- does MDAC 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 in 6 domains, corresponding to the following data sources: the UFAL Medical corpus V1.0 (MED)[2]; the European Central Bank corpus (BANK); the JRC-Acquis Communautaire corpus (LAW) (Steinberger et al., 2006); documentations for KDE, Ubuntu, GNOME and PHP from the Opus collection, merged in a IT-domain; TedTalks (TALK) (Cettolo et al., 2012), and the Koran (REL). Additional experiments use the News Commentary

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

	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 (exluding NEWS), number English and French tokens ($\times 10^6$), types and uniq types ($\times 10^3$): the latter are types that only appear in a given domain.

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, we use Byte-Pair Encoding (Sennrich et al., 2016) 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 chose 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-mt[5] (Neubig et al., 2019). We report significant differences at the level of $p = 0.05$.

4.2 Baseline systems

Our baselines are standard for multidomain settings.[6] Using Transformers (Vaswani et al., 2017) implemented in OpenNMT-tf[7] (Klein et al., 2017), we build the following systems:

- Generic models trained with predefined mixtures of the training data taking the form:
$$\lambda_\alpha(d) = (\sum_{d=1}^{n_d} q_d^\alpha)^{-1}(q_d^\alpha) \quad q_d = \frac{|\ N_d^s\ |}{N^s} \quad (2)$$
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 based on Mixed−1.0, 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 fine-tuning (FT−Full) procedure updates all the parameters of the initial model, resulting in six systems, one per domain, with no parameter sharing across domains.
- systems trained with fixed mixtures with $\boldsymbol{\lambda}^l \in \left[\boldsymbol{\lambda}_0, \boldsymbol{\lambda}_{0.25}, \boldsymbol{\lambda}_{0.5}, \boldsymbol{\lambda}_{0.75}, \boldsymbol{\lambda}_{1.0}\right]$; these are used in the multidomain experiments of § 5.3;
- our implementations of 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 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 $\beta_1 = 0.9$, $\beta_2 = 0.98$, Noam decay ($warmup\_steps = 4000$), and a dropout rate of 0.1 in all layers.

4.3 CL and DDS re-implementations

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 DA 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 data 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 $\boldsymbol{\lambda}^l_{t=0}$, and (b) the target (dev/test) distribution $\boldsymbol{\lambda}^t$. We thus report these systems as MDAC($\boldsymbol{\lambda}^l_{t=0}$, $\boldsymbol{\lambda}^t$) and compare with DDS using the same settings.

In our work, we parameterize the distribution $\boldsymbol{\lambda}^l$ as follows (with $\beta = 2$ in all experiments):[9]

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

[3] http://opus.nlpl.eu

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

[5] https://github.com/neulab/compare-mt

[6] We 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

[8] https://github.com/cindyxinyiwang/multiDDS

[9] The *spherical softmax* in (de Brébisson and Vincent, 2016).

This parameterization avoids the "rich-get-richer" effect that we observe with $\boldsymbol{\lambda}(\boldsymbol{\psi}) = \mathrm{softmax}(\boldsymbol{\psi})$, 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 $\mathbf{lr}_{data} = 0.001$. We update the sampling distribution via 100 gradient descent iterations for almost all experimental settings except that for adaptation with automatic clusters (§ 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 $\mathbf{lr}_{data}$ 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 2,000 iterations to compute the reward of each domain (eq. (1)). During this step, we update the temporary checkpoint with k updates for each domain, which costs as much as k training updates. Therefore, we execute $k \times n_d$ updates after every 2,000 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 our method 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 expert NMT models for each domain. To challenge the flexibility of the method, we also consider a *two-domain adaptation task*, where given the same 6 domains, we focus on adapting to a mixture of 2 domains. In the *multidomain adaptation task*, we use the same 6 domains to build one single NMT model that should perform optimally, assuming a uniform distribution of domains during the test. A fourth experiment (*unseen domain adaptation*), adds to the training data for 6 domains a small dev set in a new domain (NEWS): our target is a 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 as in (Tars and Fishel, 2018), then learn mixture weights these clusters to one of 6 domains using the corresponding 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 $\boldsymbol{\lambda}^t$ to a deterministic distribution $\boldsymbol{\lambda}_d$, where the target domain d has probability 1.

We consider three initializations for MDAC and DDS, using $\boldsymbol{\lambda}_0$, $\boldsymbol{\lambda}_1$ and $\boldsymbol{\lambda}_d$. According to Table 2, MDAC achieves the overall best performance when $\boldsymbol{\lambda}_{t=0} = \boldsymbol{\lambda}_0$. Doing so proves much better than initializing with $\boldsymbol{\lambda}_d$ for small domains: TALK, BANK and IT. Conversely, initializing with $\boldsymbol{\lambda}_d$ 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 $\boldsymbol{\lambda}_{t=0} = \boldsymbol{\lambda}_0$) 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 suggests 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. Results for CL, which does only well the large domain MED, lag somewhat behind.

5.2 Two-domain adaptation

In these control experiments, we showcase the flexibility of dynamic sampling and adapt to (arbitrary) pairs of target domains with equal weight, contrasting MDAC with DDS in Table 3. Here, MDAC significantly outperforms DDS in two settings (MED+IT and LAW+BANK) out of three.

5.3 Multi-domain NMT

We now turn to a more realistic scenario and consider multidomain NMT, which aims to train one single system with optimal performance averaged over 6 domains and targets a uniform test distribution $\boldsymbol{\lambda}^t = \boldsymbol{\lambda}_0$. In this situation, CL (Zhang et al., 2019) does not apply: we only contrast the performance of MDAC, DDS and several fixed training data distribution $\boldsymbol{\lambda}^l \in \left[\boldsymbol{\lambda}_0, \boldsymbol{\lambda}_{0.25}, \boldsymbol{\lambda}_{0.5}, \boldsymbol{\lambda}_{0.75}, \boldsymbol{\lambda}_{1.0}\right]$, where $\boldsymbol{\lambda}_\alpha$ is defined according to equation (2).

We again initialize MDAC and DDS with two distribution $\boldsymbol{\lambda}_0$ and $\boldsymbol{\lambda}_1$. According to Table 4, MDAC achieves the best performance with initial (uniform) $\boldsymbol{\lambda}_0$. The same conclusion holds for DDS. For this

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 ($\boldsymbol{\lambda}_0, \boldsymbol{\lambda}_d$)	39.6	60.1	55.0	38.5	52.5	92.0	56.3
MDAC ($\boldsymbol{\lambda}_0, \boldsymbol{\lambda}_d$)	39.6	62.5**	55.6*	38.5	52.4	92***	56.8
DDS ($\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_d$)	39.7	53.9	49.6	37.9	43.1	64.3	48.1
MDAC ($\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_d$)	40.2	59.9	52.6	38.5	50.7	79.8	53.6
DDS ($\boldsymbol{\lambda}_d, \boldsymbol{\lambda}_d$)	39.9	63.9	54.5	35.4	51.2	91.8	56.1
MDAC ($\boldsymbol{\lambda}_d, \boldsymbol{\lambda}_d$)	40.6	63.9	54.5	35.6	51.3	92.3	56.4

Table 2: Single domain adaptation. 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 $p < 0.05$.

configuration, MDAC outperforms static training distributions including $\left[\boldsymbol{\lambda}_0, \boldsymbol{\lambda}_{0.75}, \boldsymbol{\lambda}_{1.0}\right]$ by a significant margin, and performs slightly better than $\left[\boldsymbol{\lambda}_{0.25}, \boldsymbol{\lambda}_{0.5}\right]$. Using MDAC thus dispenses with the empirical search of an 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 plots the evolution of the mixture weights during training, helps to understand the difference between the two methods. For DDS (Figure 2a), the sampling distribution quickly reaches a bi-modal regime 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 part of training data in the early stages; their weights then slowly decrease as larger domains such as MED and LAW increase their share. This only happens at the end of training, when some NMT models might already be close to their peak performance for the small domains.

5.4 Unseen domain

The left part of Table 5 displays the performance on the unseen domain NEWS for systems trained with mixtures $\boldsymbol{\lambda}^l \in \left[\boldsymbol{\lambda}_0, \boldsymbol{\lambda}_{0.25}, \boldsymbol{\lambda}_{0.5}, \boldsymbol{\lambda}_{0.75}, \boldsymbol{\lambda}_{1.0}\right]$ and with dynamic data selection (MDAC and DDS). These systems have insignificant differences in BLEU, suggesting that dynamic mixtures do not improve the robustness of NMT systems against unseen domains. However, the performance of MDAC and DDS remains close to the best performance, showing that they also apply in such settings.

5.5 Automatic clustering

The right part of Table 5 reports the performance of NMT systems adapted to each domain. In comparison to Section 5.1, the training data is distributed in 30 automatic clusters instead of the 6 original domains. Splitting the train data into small groups gives the learner extra degrees of freedom when selecting the best distribution. However, as these clusters are built automatically, they are noisier in nature. According to 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. With a uniform initialization, this cluster starts with a 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 et al., 2017), where the authors distinguish between data-centric and model-centric DA, a view also adopted in the recent survey of Saunders (2021). Our approach to DA in this paper falls under the former category. We refer readers interested in DA to these papers.

Multidomain NMT (MDMT) aims to develop systems that simultaneously bode well for several domains. Like for DA, techniques for supervised MDMT 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 MDMT is studied in (Farajian et al., 2017), as an instance of unsupervised DA.

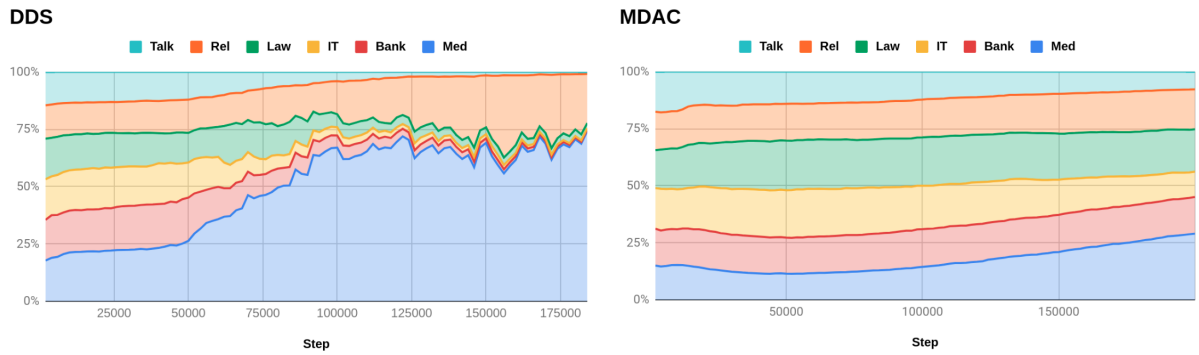
Most approaches to adaptive/dynamic data selection take inspiration from Bengio et al. (2009), where the notion of curriculum learning is introduced. CL relies on the notion of the "easiness" of

domain $d=$	MED	LAW	BANK	TALK	IT	REL
DDS $(\boldsymbol{\lambda}_0, \boldsymbol{\lambda}_2)$	39.5	-	-	-	50.1	-
MDAC $(\boldsymbol{\lambda}_0, \boldsymbol{\lambda}_2)$	39.1	-	-	-	51.8*	-
DDS $(\boldsymbol{\lambda}_0, \boldsymbol{\lambda}_2)$	-	60.8	53.3	-	-	-
MDAC $(\boldsymbol{\lambda}_0, \boldsymbol{\lambda}_2)$	-	61.9*	54.5*	-	-	-
DDS $(\boldsymbol{\lambda}_0, \boldsymbol{\lambda}_2)$	-	-	-	37.9	-	91.3
MDAC $(\boldsymbol{\lambda}_0, \boldsymbol{\lambda}_2)$	-	-	-	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 $(\boldsymbol{\lambda}_0, \boldsymbol{\lambda}_0)$	40.1	56.9	50.7	37.4	46.8	92.0	54.0
MDAC $(\boldsymbol{\lambda}_0, \boldsymbol{\lambda}_0)$	38.5	60.3**	54.4*	37.3	51.3**	91.4*	55.5**
DDS $(\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_0)$	40.6	55.5	48.0	36.2	46.9	60.1	47.9
MDAC $(\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_0)$	40.2	59.3**	51.0**	36.9**	48.6**	80.7**	52.8**

Table 4: Multidomain 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 $p < 0.05$.



(a) DDS **(b)** MDAC

Figure 2: Evolution of the sampling distribution during training.

domain $d=$	NEWS
Unseen domain	
Mixed-0	25.7
Mixed-0.25	25.8
Mixed-0.5	26.5
Mixed-0.75	26.8
Mixed-1	26.9
DDS $(\boldsymbol{\lambda}_0, \boldsymbol{\lambda}_{news})$	26.3
MDAC $(\boldsymbol{\lambda}_0, \boldsymbol{\lambda}_{news})$	26.3

domain $d=$	MED	LAW	BANK	TALK	IT	REL	mean
Training with 30 clusters							
DDS $(\boldsymbol{\lambda}^*, \boldsymbol{\lambda}_d)$	38.3	60.1	50.3	35.8	49.1	90.1	53.9
MDAC $(\boldsymbol{\lambda}^*, \boldsymbol{\lambda}_d)$	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.

a sample to schedule the presentation of training data so as to start with the easiest examples and end with the hardest. Various ways to automate CL in 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 multidomain and 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: early stages only use in-data, while less relevant[10] data are introduced in later stages. This is opposite to the policy of van der Wees et al. (2017), whose *gradual fine-tuning* progressively increases the focus on in-domain data.

Kumar et al. (2019) use reinforcement 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 devset loss that results from the current data selection strategy. This technique is applied to multilingual NMT in (Kumar et al., 2021). Zhou et al. (2020) propose another CL-based approach which relies on *instance uncertainty* as a measure of their difficulty and presents data samples starting with the easiest. Another contribution of this work is a new stopping criterium. Closest to our problems, Wang et al. (2020a) adapt CL for multidomain NMT, where an optimal instance weighting scheme is found using Bayesian optimization techniques. Each step consists of (a) weighting instances based on relevance features, (b) fine-tuning a pretrained model using the weighted training set, and is applied to train a sequence of models. The one that maximizes the devset performance is finally retained.

7 Conclusion and outlook

In this study, we have presented a generic framework to perform multiple adaptation tasks for machine translation, ranging from supervised domain adaptation to multidomain NMT and unseen domain adaptation. In our experiments, we have shown that the same algorithm, aimed at automatically finding an effective data sampling scheme during the course of training, can 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.

Our future work will 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 § 5.5, where our results lag behind supervised DA. This might be due to the inability of our simplistic optimization strategy to handle situations where the number of clusters is large.

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

References

Aharoni, Roee, Melvin Johnson, and Orhan Firat. 2019. Massively multilingual neural machine translation. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, pages 3874–3884, Minneapolis, USA.

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*, pages 355–362, Edinburgh, UK.

Bengio, Yoshua, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th International Conference on Machine Learning*, page 41–48, Montréal, Canada.

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.

Cettolo, Mauro, Christian Girardi, and Marcello Federico. 2012. Wit^3: Web inventory of transcribed and translated talks. In *Proceedings of the 16th Conference of the European Association for Machine Translation*, pages 261–268, Trento, Italy.

Chen, Wenhu, Evgeny Matusov, Shahram Khadivi, and Jan-Thorsten Peter. 2016. Guided alignment training for topic-aware neural machine translation. In *Proceedings of the Conference of the Association for Machine Translation in the Americas*, Austin, USA.

Chu, Chenhui and Raj Dabre. 2018. Multilingual and multi-domain adaptation for neural machine translation. In *Proceedings of the 24st Meeting of the Association for Natural Language Processing*, pages 909—912, Okayama, Japan.

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, 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, Y. and Y. LeCun, editors, *Proceedings of the 4th International Conference on Learning Representation*, 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.

Firat, Orhan, Kyunghyun Cho, and Yoshua Bengio. 2016. Multi-way, multilingual neural machine translation with a shared attention mechanism. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, pages 866–875.

Graves, Alex, Marc G. Bellemare, Jacob Menick, Rémi Munos, and Koray Kavukcuoglu. 2017. Automated curriculum learning for neural networks. In Precup, D. and Y.-W. Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 1311–1320.

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*, 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.

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.

Johnson, Melvin, Mike Schuster, Quoc Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernand a 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*, pages 67–72, Vancouver, Canada.

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*, 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*, pages 726–739, Belgium, Brussels.

Koehn, Philipp. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 388–395, Barcelona, Spain.

Kumar, Gaurav, George Foster, Colin Cherry, and Maxim Krikun. 2019. Reinforcement learning based curriculum optimization for neural machine translation. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, pages 2054–2061, Minneapolis, USA.

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*, 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.

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*, pages 367–374.

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 Conference on Empirical Methods in Natural Language Processing*, pages 1489–1494, Copenhagen, Denmark.

Moore, Robert C. and William Lewis. 2010. Intelligent selection of language model training data. In *Proceedings of ACL*, pages 220–224, Uppsala, Sweden.

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, E., L. Derczynski, and P. Isabelle, editors, *Proceedings of the International Conference on Computational Linguistics*, pages 1008–1021, Santa Fe, 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*, pages 311–318.

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*, page 9p, Hong-Kong, China.

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.

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

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. 2016. 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.

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*, Genoa, Italy.

Tars, Sander and Mark Fishel. 2018. Multi-domain neural machine translation. In Pérez-Ortiz, J.-A., F. Sánchez-Martínez, M. Esplà-Gomis, M. Popović, C. Rico, A. Martins, J. Van den Bogaert, and M. Forcada, editors, *Proceedings of the 21st Annual Conference of the European Association for Machine Translation*, pages 259–269, Alicante, Spain.

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.

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.

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.

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.

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 Conference on Empirical Methods in Natural Language Processing*, pages 447–457, Brussels, Belgium.

Zhang, Jian, Liangyou Li, Andy Way, and Qun Liu. 2016. Topic-informed neural machine translation. In *Proceedings of the International Conference on Computational Linguistics*, pages 1807–1817, Osaka, Japan.

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 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 1903–1915, Minneapolis, USA.

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 Meeting of the Association for Computational Linguistics*, pages 6934–6944, Online.