# Dynamically Composing Domain-Data Selection with Clean-Data Selection by "Co-Curricular Learning" for Neural Machine Translation

**Wei Wang** and **Isaac Caswell** and **Ciprian Chelba**
Google Research
{wangwe,icaswell,cipriancheIba}@google.com

## Abstract

Noise and domain are important aspects of data quality for neural machine translation. Existing research focus separately on domain-data selection, clean-data selection, or their static combination, leaving the dynamic interaction across them not explicitly examined. This paper introduces a "co-curricular learning" method to compose dynamic domain-data selection with dynamic clean-data selection, for transfer learning across both capabilities. We apply an EM-style optimization procedure to further refine the "co-curriculum". Experiment results and analysis with two domains demonstrate the effectiveness of the method and the properties of data scheduled by the co-curriculum.

## 1 Introduction

Significant advancement has been witnessed in neural machine translation (NMT), thanks to better modeling and data. As a result, NMT has found successful use cases in, for example, domain translation and helping other NLP applications, e.g., (Buck et al., 2018; McCann et al., 2017). As these tasks start to scale to more domains, a challenge starts to surface: Given a source monolingual corpus, how to use it to improve an NMT model to translate same-domain sentences well? Data selection plays an important role in this context.

In machine translation, data selection has been a fundamental research topic. One idea (van der Wees et al., 2017; Axelrod et al., 2011) for this problem is to use language models to select parallel data out of a background parallel corpus, seeded by the source monolingual sentences. This approach, however, performs poorly on noisy data, such as large-scale, web-crawled datasets, because data noise hurts NMT performance (Khayrallah and Koehn, 2018). The lower learning curve in Figure 1 shows the effect of noise on domain-data selection.

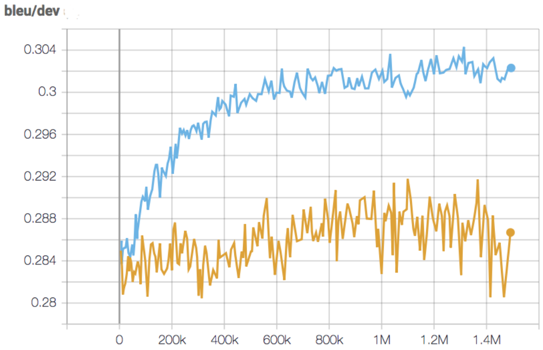

Figure 1: BLEU curves over NMT training steps: domain-data selection on Paracrawl English→French data (lower curve) vs. clean-data selection on the same data (upper curve). Setup available in the experiment section.

NMT community has realized the harm of data noise to translation quality, leading to efforts in data denoising (Koehn et al., 2018), as has been popular in computer vision (Hendrycks et al., 2018). The upper curve in Figure 1 shows the effect of clean-data selection on the same noisy data. These denoising methods, however, cannot be directly used for the problem in question as they require trusted parallel data as input.

We introduce a method to dynamically combine clean-data selection and domain-data selection. We treat them as independent curricula, and compose them into a "co-curriculum". We summarize our contributions as:

1. "Co-curricular learning", for transfer learning across data quality. It extends the single curriculum learning work in NMT and makes the existing domain-data selection method work better with noisy data.

2. A curriculum optimization procedure to refine the co-curriculum. While gaining some

*Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1282–1292
Florence, Italy, July 28 - August 2, 2019. ©2019 Association for Computational Linguistics

improvement with deep models, it surprisingly improves shallow model by 8-10 BLEU points – We find that bootstrapping seems to “regularize” the curriculum and make it easier for a small model to learn on.

3. We wish our work contributed towards better understanding of data, such as noise, domain, or “easy to learn”, and its interaction with NMT network.

## 2 Related Work

### 2.1 Measuring Domain and Noise in Data

Data selection for MT usually uses a scoring function to rank sentence pairs. Cross entropy difference (Moore and Lewis, 2010) between two language models is usually used for selecting domain sentences, e.g., (van der Wees et al., 2017; Axelrod et al., 2011). For a source sentence $x$ of length $|x|$, with a general-domain language model (LM), parameterized as $\widetilde{\vartheta}$, and an in-domain LM, $\widehat{\vartheta}$, the domain-relevance of $x$ is calculated as:[1]

$$\varphi\left(x;\widetilde{\vartheta},\widehat{\vartheta}\right)=\frac{\log P\left(x;\widehat{\vartheta}\right)-\log P\left(x;\widetilde{\vartheta}\right)}{|x|} \quad (1)$$

Alternative measures (Wang et al., 2017; Chen and Huang, 2016; Chen et al., 2016) also show effectiveness. With Eq. 1 to select data, the data distribution (domain quality) in the in-domain monolingual data used to train $P(x;\widehat{\vartheta})$ is transferred into the selected data through the scoring.

Data selection has also been used for data denoising (Junczys-Dowmunt, 2018; Wang et al., 2018b), by using NMT models and trusted data to measure the noise level in a sentence pair. One such a scoring function uses a baseline NMT, $\widetilde{\theta}$, trained on noisy data and a cleaner NMT, $\widehat{\theta}$, obtained by fine-tuning $\widetilde{\theta}$ on a small trusted parallel dataset, and measures quality in a sentence pair $(x, y)$:

$$\phi\left(x,y;\widetilde{\theta},\widehat{\theta}\right)=\frac{\log P\left(y|x;\widehat{\theta}\right)-\log P\left(y|x;\widetilde{\theta}\right)}{|y|} \quad (2)$$

Using NMT models for selection can also lead to faster convergence (Wang et al., 2018a). With Eq. 2, the distribution (data quality) in the trusted parallel data is transferred into the selected data. These scoring functions usually use smaller networks.

### 2.2 Curriculum Learning for NMT

Curriculum learning (CL) (Bengio et al., 2009) has been used to further improve traditional static selection. In CL, a curriculum, $\mathcal{C}$, is a sequence of training criteria over training steps. A training criterion, $Q_t(y|x)$, at step $t$ is associated with a set of weights, $W_t(x, y)$, over training examples $(x, y)$ in a dataset $D$, where $y$ is the translation for $x$. $Q_t(y|x)$ is a re-weighting of the training distribution $P(y|x)$:

$$Q_t\left(y|x\right) \propto W_t\left(x,y\right) P\left(y|x\right), \forall (x,y) \in D \quad (3)$$

Hence, for a training with $T$ maximum steps, $\mathcal{C}$ is a sequence:

$$\mathcal{C} = \langle Q_1, ..., Q_t, ..., Q_T \rangle \quad (4)$$

At $t$, an online learner samples data from $Q_t$ to train on, resulting in a task (or model), $m_t$. Therefore, $\mathcal{C}$ corresponds to a sequence of tasks, $\mathcal{M} = \langle m_1, ..., m_t ..., m_f \rangle$, where $m_f$ is the final task of interest. Intermediate tasks, $m_t$, are sorted in increasing relevance to $m_f$ as a series of “stepping stones” to $m_f$, making curriculum learning a form of transfer learning that transfers knowledge through $\mathcal{M}$ to benefit $m_f$. A performance metric $\mathcal{P}(\mathcal{C}, m_f)$ is used to evaluate $m_f$.

There has already been rich research in CL for NMT. Fine-tuning a baseline on in-domain parallel data is a good strategy (Thompson et al., 2018; Sajjad et al., 2017; Freitag and Al-Onaizan, 2016). van der Wees et al. (2017) introduce a domain curriculum. Wang et al. (2018b) define noise level and introduce a denoising curriculum. Kocmi and Bojar (2017) use linguistically-motivated features to classify examples into bins for scheduling. Kumar et al. (2019) use reinforcement learning to learn a denoising curriculum based on noise level of examples. Zhang et al. (2018) explore CL in general for NMT and observe faster training convergence. Zhang et al. (2019) use CL to adapt generic NMT models to a specific domain. Platanios et al. (2019) propose a CL framework to simplify and speed up training and achieve better results; a nice study in sampling schedules was carried out.

CL therefore is a natural formulation for dynamic online data selection. Our work is built on two types of dynamic data selection: Dynamic domain-data selection and dynamic clean-data selection. The former uses the neural LM (NLM)-based scoring function (Eq. 1), which we call

[1] We can use both source and target LMs, but we study the problem where only a source in-domain corpus is available..

**domain curriculum**, denoted by $\mathcal{C}_{\text{domain}}$. The later uses the NMT-based scoring function (Eq. 2), which we call **denoising curriculum**, denoted by $\mathcal{C}_{\text{denoise}}$. Ideally, we would have in-domain, trusted parallel data to design a **true curriculum**, $\mathcal{C}_{\text{true}}$, as an assessment oracle: with trusted in-domain parallel data, $\mathcal{C}_{\text{denoise}}$ is expected to simultaneously perform domain-data selection and clean-data selection, becoming $\mathcal{C}_{\text{true}}$.

Mini-batch sampling is important for CL. Several alternatives have been introduced to evolve the training criteria $Q_t$ over time (Zhang et al., 2018; Wang et al., 2018b; van der Wees et al., 2017; Kocmi and Bojar, 2017; Platanios et al., 2019). In these curricula, tasks in $\mathcal{M}$ are sequenced in order of increasing relevance. Earlier tasks are exposed to a diversity of examples and later tasks progressively concentrate on data subsets more relevant to the final task.

### 2.3 More Related Work

Junczys-Dowmunt (2018) introduces a practical and effective method to combine (static) features for data filtering. Mansour et al. (2011) combine an n-gram LM and IBM translation Model 1 (Brown et al., 1993) for domain data filtering. We compose different types of dynamic online selection rather than combining static features.

Back translation (BT), e.g., (Sennrich et al., 2016), is another important approach to using monolingual data for NMT. Here we use monolingual data to seed data selection, rather than generating parallel data directly from it. Furthermore, we study the use of source-language monolingual data, in which case BT cannot be applied directly.

## 3 Problem Setting

$\widetilde{D_{XY}}$ is a background parallel dataset between languages $X$ and $Y$. It may be crawled from the web: large (hundreds of millions of pairs), diverse and noisy.

$D^{\text{ID}}_X$ is an in-domain monolingual corpus in source language $X$. It contains thousands to millions of sentences and specifies the testing domain. With $D^{\text{ID}}_X$, we can train $\varphi$ (Eq. 1) to sort data by domain relevance into a domain curriculum. $D^{\text{ID}}_X$ can be small because we can use it to fine-tune $\widetilde{\vartheta}$ into $\widehat{\vartheta}$.

$\widehat{D^{\text{OD}}_{XY}}$ is a small, trusted, out-of-domain (OD) parallel dataset. It contains several thousands of pairs or fewer. With $\widehat{D^{\text{OD}}_{XY}}$, we can train the $\phi$ (Eq. 2) to sort data by noise level into a denoising curriculum.

3 en→zh sentence pairs:

1 (en) Where is the train station?
(zh-gloss) TRAIN STATION IS WHERE?
2 (en) I'd like to have two window seats.
(zh-gloss) PLS. BOOK ME TWO WINDOW SEATS.
3 (en) It usually infects people older than 60.
(zh-gloss) PEOPLE OLDER THAN 60 USUALLY ARE INFECTED BY IT.

| | $W_1 \rightarrow$ | $W_2 \rightarrow$ | $W_3 \rightarrow$ | $W_4$ |
|---|---|---|---|---|
| Travel domain curri.<br>$\varphi(3) < \varphi(2) < \varphi(1)$ | $\begin{pmatrix}1/3\\1/3\\1/3\end{pmatrix}$ | $\begin{pmatrix}1/3\\1/3\\1/3\end{pmatrix}$ | $\begin{pmatrix}1/2\\1/2\\\cancel{0.0}\end{pmatrix}$ | $\begin{pmatrix}1.0\\\cancel{0.0}\\\cancel{0.0}\end{pmatrix}$ |
| Denoising curri.<br>$\phi(2) < \phi(1) < \phi(3)$ | $\begin{pmatrix}1/3\\1/3\\1/3\end{pmatrix}$ | $\begin{pmatrix}1/2\\\cancel{0.0}\\1/2\end{pmatrix}$ | $\begin{pmatrix}1/2\\\cancel{0.0}\\1/2\end{pmatrix}$ | $\begin{pmatrix}1/2\\\cancel{0.0}\\1/2\end{pmatrix}$ |
| Co-curriculum<br>(Our goal) | $\begin{pmatrix}1/3\\1/3\\1/3\end{pmatrix}$ | $\begin{pmatrix}1/2\\\cancel{0.0}\\1/2\end{pmatrix}$ | $\begin{pmatrix}1.0\\\cancel{0.0}\\\cancel{0.0}\end{pmatrix}$ | $\begin{pmatrix}1.0\\\cancel{0.0}\\\cancel{0.0}\end{pmatrix}$ |

Table 1: Curriculum and co-curriculum examples generated from a toy dataset. Each is characterized by its re-weighting, $W_t$, over four steps, to stochastically order data to benefit a final task. $\varphi$: the domain scoring function (Eq. 1). $\phi$: the denoising scoring function (Eq. 2). Strikethrough marks discarded examples.

The setup, however, assumes that the in-domain, trusted parallel data, $\widehat{D^{\text{ID}}_{XY}}$, does not exist – Our goal is to use an easily available monolingual corpus and recycle existing trusted parallel data to reduce the cost of curating in-domain parallel data.

We are interested in a composed curriculum, $\mathcal{C}_{\text{co}}$, to improve either original curriculum:

$$\mathcal{P}\left(\mathcal{C}_{\text{co}}, m_f\right) > \mathcal{P}\left(\mathcal{C}_{\text{denoise}}, m_f\right) \tag{5}$$

$$\mathcal{P}\left(\mathcal{C}_{\text{co}}, m_f\right) > \mathcal{P}\left(\mathcal{C}_{\text{domain}}, m_f\right) \tag{6}$$

We hope $\mathcal{P}(\mathcal{C}_{\text{co}}, m_f) \approx \mathcal{P}(\mathcal{C}_{\text{true}}, m_f)$ as if a small in-domain, trusted parallel dataset were available.

## 4 Co-Curricular Learning

Table 1 illustrates the idea with a toy dataset of three examples. Source sentences (en) of examples 1 and 2 are in the travel domain. Example 2 is a noisy translation. Example 3 is well-translated but belongs to the medicine domain. A travel-domain curriculum follows its data re-weighting, $W_t$, and gradually discards (strikethrough) less in-domain examples, optimizing towards a travel-domain model. The denoising curriculum gradually discards noisy examples to improve general accuracy, without paying special attention to travel domain. We want to "fuse" these two partial curricula into a co-curriculum to train models progressively on both in-domain and clean examples. We call this co-curricular learning.

### 4.1 Curriculum Mini-Batching

To facilitate the definition of co-curricular learning and following (Platanios et al., 2019; Wang et al., 2018b), we define a dynamic data selection function, $\mathcal{D}^{\phi}_{\lambda}(t, D)$, to return the top $\lambda(t)$ of examples in a dataset $D$ sorted by a scoring function $\phi$ at a training step $t$. We use $\lambda(t) = 0.5^{t/H}$, $(0 < \lambda \leq 1)$, as a *pace function* to return a selection ratio value that decays over time controlled by a hyper-parameter $H$.[2] During training, $\mathcal{D}^{\phi}_{\lambda}(t, D)$ progressively evolves into smaller subdatasets that are more relevant to the final task using the scoring function. In practice, $\mathcal{D}^{\phi}_{\lambda}(t, D')$ can be applied on a small buffer $D'$ of random examples from the much bigger $D$, for efficient online training. It may also be desirable to set a floor value on $\lambda(t)$ to avoid potential data selection bias. This is how we implement a curriculum in experiments. We introduce two different co-curricula below.

### 4.2 Mixed Co-Curriculum ($\mathcal{C}^{\text{mix}}_{\text{co}}$)

Mixed co-curriculum, $\mathcal{C}^{\text{mix}}_{\text{co}}$, simply adds up the domain scoring function (Eq. 1) and the denoising function (Eq. 2). For a sentence pair $(x, y)$,

$$\psi(x, y) = \phi(x, y) + \varphi(x).$$

We then can constrain the re-weighting, $W_t(x, y)$, to assign non-zero weights only to examples in $D^{\psi}_{\lambda}(t, \widetilde{D_{XY}})$ at a training step. We use uniform sampling. The co-curriculum is thereby fully instantiated based on Eq. 3 and Eq. 4. However, values of $\phi$ and $\varphi$ may not be on the same scale or even from the same family of distributions. Therefore, despite its simplicity, $\mathcal{C}^{\text{mix}}_{\text{co}}$ may not be able to enforce either curriculum sufficiently.

### 4.3 Cascaded Co-Curriculum ($\mathcal{C}^{\text{cascade}}_{\text{co}}$)

Cascaded co-curriculum, $\mathcal{C}^{\text{cascade}}_{\text{co}}$, defines two selection functions and nests them. Let $\beta(t) = 0.5^{t/F}$ and $\gamma(t) = 0.5^{t/G}$ be two pace functions, implemented similarly to above $\lambda(t)$, with different hyper-parameters $F$ and $G$.[3] They control the data-discarding paces for clean-data selection and domain-data selection, respectively. At step $t$, $\mathcal{D}^{\phi}_{\beta}\left(t, \widetilde{D_{XY}}\right)$ retains the top $\beta(t)$ of background data $\widetilde{D_{XY}}$, sorted by scoring function $\phi(x, y)$. $\mathcal{D}^{\varphi}_{\gamma}\left(t, \mathcal{D}^{\phi}_{\beta}\left(t, \widetilde{D_{XY}}\right)\right)$ retains the top $\gamma(t)$ of $\mathcal{D}^{\phi}_{\beta}\left(t, \widetilde{D_{XY}}\right)$, re-sorted by scoring function $\varphi(x)$. That is,

$$\left(\mathcal{D}^{\varphi}_{\gamma} \circ \mathcal{D}^{\phi}_{\beta}\right)\left(t, \widetilde{D_{XY}}\right) = \mathcal{D}^{\varphi}_{\gamma}\left(t, \mathcal{D}^{\phi}_{\beta}\left(t, \widetilde{D_{XY}}\right)\right)$$

Then Eq. 3 is redefined into Eq. 4 with uniform sampling:[4]

$$W_t(x, y) = \begin{cases} \frac{1}{\left|\mathcal{D}^{\varphi}_{\gamma} \circ \mathcal{D}^{\phi}_{\beta}\right|} & \text{if } (x, y) \in \mathcal{D}^{\varphi}_{\gamma} \circ \mathcal{D}^{\phi}_{\beta} \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

Compared to $\mathcal{C}^{\text{mix}}_{\text{co}}$, $\mathcal{C}^{\text{cascade}}_{\text{co}}$ cascades $\mathcal{C}_{\text{denoise}}$ and $\mathcal{C}_{\text{domain}}$ per step.

At a time step, both pace functions, in their respective paces, discard examples that become less relevant to their own tasks. All surviving examples then have an equal opportunity to be sampled. Even though uniformly sampled, examples that are more relevant are retained longer in training and thus weighed more over time.

Table 1 shows a toy example of how two curricula are composed. At step 1, no example is discarded yet, and all examples have equal sampling opportunity ($W_1$'s). At step 2, the denoising curriculum discards the noisiest example 2, but the domain curriculum still keeps all; So only 1 and 3 are retained in the co-curriculum ($W_2$). In step 3, the domain curriculum discards the least in-domain example 3, so only 1 is left in the co-curriculum now ($W_3$). The denoising curriculum has a slower pace than the domain curriculum. Over the four steps, example 1 is kept longer thus weighed more.

### 4.4 Curriculum Optimization

We further improve the co-curriculum using an EM (Dempster et al., 1977) style optimization procedure in training, as shown in Figure 2. It aims specifically to iteratively improve the denoising selection, without losing quality on the domain selection.

[2] This is inspired by the exponential learning rate schedule. In the following notations, we omit $H$ for brevity, but the function name implies it.

[3] We will omit $F, G$ for brevity, but the function names can indicate them.

[4] Function nesting is asymmetrical, but the uniform sampling seems to make the nesting irrelevant to the nesting order. In experiments, we did not notice empirical differences between nesting one way or the other.

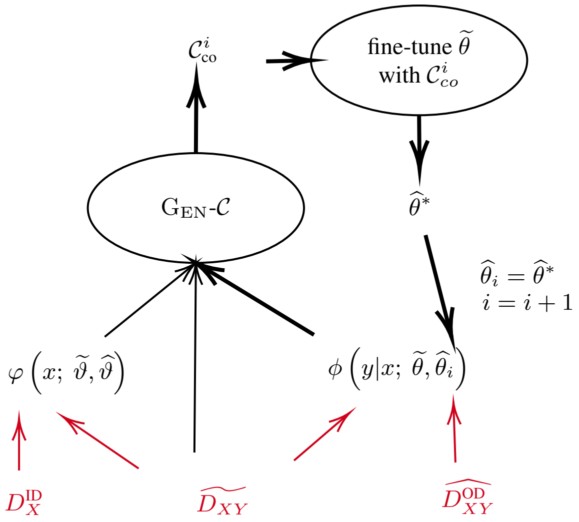
$\mathcal{C}_{\text{co}}^i$
fine-tune $\widetilde{\theta}$ with $\mathcal{C}_{co}^i$
GEN-$\mathcal{C}$
$\widehat{\theta}^*$
$\widehat{\theta}_i = \widehat{\theta}^*$
$i = i + 1$
$\varphi\left(x;\ \widetilde{\vartheta}, \widehat{\vartheta}\right)$
$\phi\left(y|x;\ \widetilde{\theta}, \widehat{\theta}_i\right)$
$D_X^{\text{ID}}$
$\widetilde{D_{XY}}$
$\widehat{D_{XY}^{\text{OD}}}$

Figure 2: Co-curricular learning with an EM-style optimization procedure. Thicker arrows form the bootstrapping loop.

With $\widetilde{D_{XY}}$ and $D_X^{\text{ID}}$, we train a domain scoring function, $\varphi(x; \widetilde{\vartheta}, \widehat{\vartheta})$. With $\widetilde{D_{XY}}$ and $\widehat{D_{XY}^{\text{OD}}}$, we train a denoising scoring function, $\phi(y|x; \widetilde{\theta}, \widehat{\theta})$. The in-domain component $\widehat{\vartheta}$ of $\varphi$ or the clean component $\widehat{\vartheta}$ of $\phi$ are obtained by fine-tuning $\widetilde{\vartheta}$ or $\widetilde{\theta}$ on the respective seed data. These *initialize* the procedure (iteration 0).

At iteration $i$, we generate a concrete co-curriculum using the dynamic re-weighting, $W_t$, as defined in Section 4. Let GEN-$\mathcal{C}$ denote the *curriculum generation* process:

$$\mathcal{C}_{\text{co}} = \text{GEN-}\mathcal{C}\left(\widetilde{D_{XY}}, \phi_i, \varphi\right) \tag{8}$$

Then, we fine-tune the original noisy NMT component, $\widetilde{\theta}$, of $\phi$ on $\mathcal{C}_{\text{co}}$:

$$\widehat{\theta}^* = \arg\max_{\widehat{\theta}} \mathcal{P}\left(\mathcal{C}_{\text{co}}, m_f\right) \tag{9}$$

$\widehat{\theta}^*$ is used to replace the clean component of $\phi$

$$\begin{aligned} \widehat{\theta}_i &= \widehat{\theta}^* \\ i &= i + 1 \end{aligned}$$

$\widehat{\theta}_i$ is then compared against the original $\widetilde{\theta}$ for scoring. The updated $\phi$ and the constant $\varphi$ work together to generate a new co-curriculum in the next iteration going back to Eq. 8. In this process, only the denoising function $\phi$ is iteratively updated, made more aware of the domain.

We call the procedure EM-style because $\widetilde{D_{XY}}$ is treated as incomplete without the (hidden) data order. The generated $\mathcal{C}_{\text{co}}$ in each iteration sorts the data and thus is viewed as complete. It is then used to train $\widehat{\theta}$ by maximizing the performance of the final task. $\widehat{\theta}$ and $\mathcal{C}_{\text{co}}$ bootstrap each other. The process finishes after a pre-defined number of iterations. We use shallow parameterization for scoring functions but we can train a deep model on the final $\mathcal{C}_{\text{co}}$. The process also uses fine-tuning, so it can be run efficiently.

In principle, the domain-data scoring function $\varphi$ can be updated in a similar manner, too, by updating its in-domain component, $\widehat{\vartheta}$. This may help when the in-domain monolingual corpus is very small. An alternating optimization process can be used to bootstrap both. We, however, do not investigate this.

## 5 Experiments

### 5.1 Setup

We consider two background datasets and two test domains, so we have four experiment configurations. Each configuration has as inputs a background dataset, an in-domain source-language corpus and a (small) trusted parallel dataset that is out-of-domain. The inputs of a configuration are shown in Figure 2.

As alternative background datasets, we use the English→French Paracrawl data,[5] (300 million pairs), and the WMT14 training data (40 million pairs). The former is severely noisier than the later. We adopt sentence-piece model and apply open-source implementation (Kudo, 2018) to segment data into sub-word units with a source-target shared 32000 sub-word vocabulary.

We use two test domains: the English→French IWSLT15 test set, in spoken language domain; and the English→French WMT14 test set, in news domain. For IWSLT15, we use the English side of its provided parallel training data (220 thousand examples) as $D_X^{\text{ID}}$, but use the parallel version as $\widehat{D_{XY}^{\text{OD}}}$ for the WMT14 domain. The IWSLT14 test set is used for validation. For the WMT14 domain, the provided 28 million English sentences are used as $D_X^{\text{ID}}$. WMT 2010-2011 test sets are concatenated as $\widehat{D_{XY}^{\text{ID}}}$ for news[6], or as $\widehat{D_{XY}^{\text{OD}}}$ for the above

[5] https://paracrawl.eu

[6] Strictly speaking, though all are in news, the WMT 2014 monolingual data, the WMT 2011-2012 test sets and the 2014 test set are not necessarily in the exact same news domain. So this news test domain could be treated as a looser case than the IWSLT domain and examines the method at a slightly different position in the spectrum of the problem.

IWSLT15 test domain. So, the trusted data are reversely shared across the two test domains. Additionally, WMT 2012-2013 are used as the validation set for the WMT14 test domain. Our method does not require the in-domain trusted data, but we use it to construct bounds in evaluation.

We use RNN-based NMT (Wu et al., 2016) to train models. Model parameterization for $\theta$'s of $\phi$ (Eq 2) or $\vartheta$'s $\varphi$ (Eq 1) is 512 dimensions by 3 layers – NLMs are realized using NMT models with dummy source sentences (Sennrich et al., 2016). Deep models are 1024 dimensions by 8 layers. Unless specified, results are reported for deep models. We compute truecased, detokenized BLEU with `mteval-v14.pl`.

Training on Paracrawl uses Adam in warmup and then SGD for a total of 3 million steps using batch size 128, learning rate 0.5 annealed, at step 2 million, down to 0.05. Training on WMT 2014 uses batch size 96, dropout probability 0.2 for a total of 2 million steps, with learning rate 0.5 annealed, at step 1.2 million, down to 0.05, too. No dropout is used in Paracrawl training due to its large data volume.

For the pace hyper-parameters (Section 4), we empirically use $H = F = 400k$, $G = 900k$. Floor values set for $\lambda, \beta, \gamma$ are top $0.1, 0.2, 0.5$ selection ratios, respectively, such that in the cascaded co-curriculum case, the tightest effective percentile value would be the same $0.1 = 0.2 \times 0.5$, too. All single curriculum experiments use the same pace setting as $\mathcal{C}^{\text{mix}}$.

### 5.2 Baselines and Oracles

We build various systems below as baselines and oracles. Oracle systems use in-domain trusted parallel data.

Baselines:

1. $\mathcal{C}_{\text{random}}$ : Baseline model trained on background data with random data sampling.
2. $\mathcal{C}_{\text{domain}}$: Dynamically fine-tunes $\mathcal{C}_{\text{random}}$ with a domain curriculum (van der Wees et al., 2017).
3. $\mathcal{C}_{\text{denoise}}$: Dynamically fine-tunes $\mathcal{C}_{\text{random}}$ with a denoising curriculum (Wang et al., 2018b).

Oracles:

4. $\mathcal{C}_{\text{true}}$: Dynamically fine-tunes $\mathcal{C}_{\text{random}}$ with the true curriculum.
5. ID fine-tune $\mathcal{C}_{\text{random}}$: Simply fine-tunes $\mathcal{C}_{\text{random}}$ with in-domain (ID) parallel data.

| Models | Test BLEU | |
|---|---|---|
| | IWSLT15 | WMT14 |
| (P)aracrawl | | |
| P1: $\mathcal{C}_{\text{random}}$ | 34.6 | 31.6 |
| P2: $\mathcal{C}_{\text{domain}}$ | 35.7 | 32.4 |
| P3: $\mathcal{C}_{\text{denoise}}$ | 36.6 | 33.6 |
| P4: $\mathcal{C}_{\text{true}}$ | 37.2 | 34.2 |
| P5: ID fine-tune P1 | 38.5 | 34.0 |
| (W)MT | | |
| W1: $\mathcal{C}_{\text{random}}$ | 36.5 | 35.0 |
| W2: $\mathcal{C}_{\text{domain}}$ | 37.6 | 35.9 |
| W3: $\mathcal{C}_{\text{denoise}}$ | 37.4 | 36.0 |
| W4: $\mathcal{C}_{\text{true}}$ | 38.5 | 36.3 |
| W5: ID fine-tune W1 | 39.7 | 35.9 |

Table 2: Baseline and oracle models trained on Paracrawl data and WMT data, respectively. ID: in-domain. P2,3,4 (or W2,3,4) each dynamically fine-tunes P1 (or W1) with the respective curriculum. Except for P1 and W1, the two BLEU scores in each row are for *two different* training runs, each focusing on its own test domain (configuration).

We'll see if our method is better than either original curriculum and how close it is to the true curriculum oracle. In most experiments, we fine-tune a warmed-up (baseline) model to compare curricula, for quicker experiment cycles.

Baseline and oracle BLEU scores are shown in Table 2. Note that, except for P1 and W1, the two BLEU scores in a row are for *two different* training runs, each focusing on its own test domain. On either training dataset, domain curriculum, $\mathcal{C}_{\text{domain}}$, improves baseline, $\mathcal{C}_{\text{random}}$, by 0.8-1.1 BLEU (P3 vs P1, W3 vs W1). $\mathcal{C}_{\text{domain}}$ falls behind of $\mathcal{C}_{\text{denoise}}$ on the noisy Paracrawl dataset (P2 vs P3), but delivers matched performance on the cleaner WMT dataset (W2 vs W3) – noise compromises the domain capability. On the WMT training data, $\mathcal{C}_{\text{denoise}}$ improves baselines by about +1.0 BLEU on either test domain (W3 vs W1), and more on the noisier Paracrawl data: +2.0 on either test domain (P3 vs P1). The true curriculum (P4, W4) bounds the performance of $\mathcal{C}_{\text{domain}}$ and $\mathcal{C}_{\text{denoise}}$. Simple in-domain fine-tuning gives good improvements (P5 vs P1, W5 vs W1).

### 5.3 Co-Curricular Learning

**Cascading vs. mixing**. Table 3 shows per-step cascaded filtering can work better than flat mixing (P7 vs P6). So we use $\mathcal{C}_{\text{co}}^{\text{cascade}}$ for the remaining experiments.

**Curriculum BLEU comparisons**. Table 4 shows the effectiveness of co-curricular learning. On Paracrawl, co-curriculum (P7) gives more than +2 BLEU on top of no CL (P1). It improves $\mathcal{C}_{\text{domain}}$ (P7 vs P2) by +1.4 BLEU on IWSLT15 and +1.6

| Co-Curriculum | Test BLEU | |
|---|---|---|
| | IWSLT15 | WMT14 |
| P6: $\mathcal{C}_{\text{co}}^{\text{mix}}$ | 36.2 | 33.8 |
| P7: $\mathcal{C}_{\text{co}}^{\text{cascade}}$ | **37.1** | **34.0** |

Table 3: Per-step cascading works better than mixing on Paracrawl data.

| Curriculum | Test BLEU | |
|---|---|---|
| | IWSLT15 | WMT14 |
| P1: $\mathcal{C}_{\text{random}}$ | 34.6 | 31.6 |
| P2: $\mathcal{C}_{\text{domain}}$ | 35.7 | 32.4 |
| P3: $\mathcal{C}_{\text{denoise}}$ | 36.6 | 33.6 |
| P7: $\mathcal{C}_{\text{co}}$ | **37.1** | **34.0** |
| $\mathcal{C}_{\text{co}} - \mathcal{C}_{\text{domain}}$ | $+1.4$ | $+1.6$ |
| $\mathcal{C}_{\text{co}} - \mathcal{C}_{\text{true}}$ | $-0.1$ | $-0.2$ |
| W1: $\mathcal{C}_{\text{random}}$ | 36.5 | 35.0 |
| W2: $\mathcal{C}_{\text{domain}}$ | 37.6 | 35.9 |
| W3: $\mathcal{C}_{\text{denoise}}$ | 37.4 | 36.0 |
| W7: $\mathcal{C}_{\text{co}}$ | **37.8** | **36.4** |
| $\mathcal{C}_{\text{co}} - \mathcal{C}_{\text{domain}}$ | $+0.2$ | $+0.5$ |
| $\mathcal{C}_{\text{co}} - \mathcal{C}_{\text{true}}$ | $-0.7$ | $+0.1$ |

Table 4: Co-curriculum improves either constituent curriculum and no CL, can be close to the true curriculum on noisy data.

BLEU on WMT14. It is better than either constituent curriculum (P2 or P3), close to the true curriculum (P4).

On the cleaner WMT training data, co-curriculum (W7) improves either constituent curricula (W2 and W3) by smaller gains than Paracrawl: +0.2 BLEU on IWSLT15 and +0.4 on WMT14. Compared to $\mathcal{C}_{\text{true}}$ W5, co-curriculum W7 falls behind (-0.7 BLEU) on IWSLT15 and matches (+0.1 BLEU) on WMT14.

So $\mathcal{C}_{\text{co}}$ outperforms either constituent curriculum, as we target in Section 3. In both background data cases, using in-domain trusted parallel data to build oracles (P5, W5) are more effective than selecting data in our setup.

### 5.4 Effect of Curriculum Optimization

We further bootstrap the co-curriculum with the EM-style optimization procedure (Figure 2) for three iterations for all four configurations.

**Shallow models**. We use the translation performance of the clean component $P(y|x;\widehat{\theta})$ in scoring function $\phi$ (Eq. 2) as an indicator to the quality of $\mathcal{C}_{\text{co}}$ per iteration. Figure 3 shows that the BLEU scores of $P(y|x;\widehat{\theta})$ steadily become better by iterations.[7] $\widehat{\theta}$ has 512 dimensions and 3 layers. Surprisingly, EM-3 improves baseline by +10 BLEU on IWSLT15, +8.2 BLEU on WMT14 and performs better than fine-tuning baseline with the clean, out-of-domain parallel data we have. They even reach the performance of $\mathcal{C}_{\text{random}}$ (P1) that uses a much deeper model (1024 dimensions x 8 layers) trained on the vanilla data.

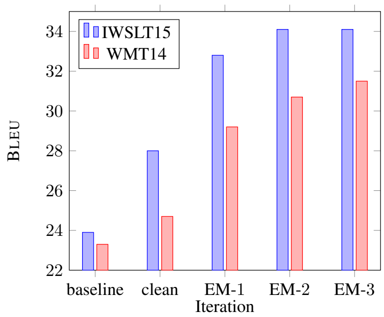

Figure 3: The EM-style optimization has a big impact on small-capacity models, measured in BLEU. Experiments were carried out on Paracrawl data.

| Curriculum | Test BLEU | |
|---|---|---|
| | IWSLT15 | WMT14 |
| P2: $\mathcal{C}_{\text{domain}}$ | 35.7 | 32.4 |
| P7: $\mathcal{C}_{\text{co}}$ | 37.1 | 34.0 |
| P8: P7+Optimization | **37.3** | **34.6** |
| P8 - $\mathcal{C}_{\text{domain}}$ | $+1.6$ | $+2.0$ |
| W2: $\mathcal{C}_{\text{domain}}$ | 37.6 | 35.9 |
| W7: $\mathcal{C}_{\text{co}}$ | **37.8** | 36.4 |
| W8: W7+Optimization | **37.8** | **36.5** |
| W8 - $\mathcal{C}_{\text{domain}}$ | $+0.2$ | $+0.6$ |

Table 5: EM-style optimization further improves domain curriculum. But, overall, it has a small impact on deep models.

**Deep models**. Table 5 shows the BLEUs of deep models (1024 dimensions x 8 layers) trained on the final co-curriculum. P8 performs slightly better than the non-bootstrapped version P7 on Paracrawl: +0.6 BLEU on WMT14 test and +0.2 on IWSLT15 test. The differences on the WMT data appear to be smaller (W8 vs. W7). So, curriculum bootstrapping has a small impact overall on deep models.

**Why the difference?** Why is there such a difference? We analyze the properties of the co-curriculum.

Each curve in Figure 4 corresponds to a single curriculum that simulates the online data selection from looser selection (left x-axis) to more-tightened selection (right x-axis). During the course of a single CL, the curriculum pushes "harder" examples with higher per-word loss (than

[7] They also include two initialization points: the noisy $\widetilde{\theta}$, and the initial clean $\widehat{\theta}$ obtained by fine-tuning $\widetilde{\theta}$ on the clean data.

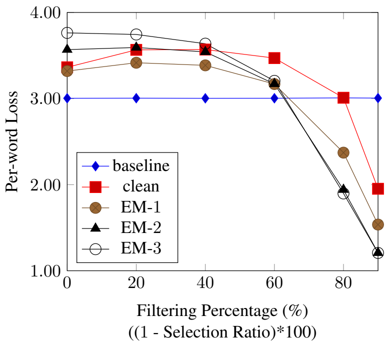

Figure 4: Curriculum learning and optimization push "easier-to-learn" (lower per-word loss) examples to late curriculum (right) and harder examples (higher per-word loss) to early curriculum (left).

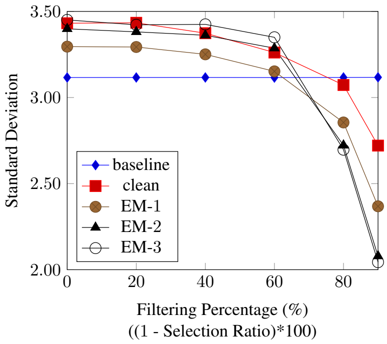

Figure 5: Curriculum learning and optimization push "regularized" (lower variance) examples to late curriculum and higher-variance examples to early curriculum.

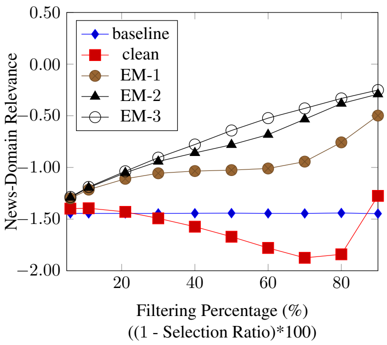

Figure 6: The denoising curriculum is made more aware of news-domain after iterations. Figure drawn for the (Paracrawl, news) configuration. Within a single curriculum, 'baseline' randomly shuffles data, thus flat curve. 'clean' uses the out-of-domain clean parallel data, thus not that much news relevance. All curves show negative news-domain relevance, indicating lack of news data in Paracrawl data.

| **Curriculum** | **Test BLEU** | |
|---|---|---|
| | **IWSLT15** | **WMT14** |
| P8: Fine-tune with $\mathcal{C}_{co}$ | 37.3 | 34.6 |
| P9: Retrain with $\mathcal{C}_{co}$ | **37.9** | **35.6** |
| W8: Fine-tune with $\mathcal{C}_{co}$ | 37.8 | **36.5** |
| W9: Retrain with $\mathcal{C}_{co}$ | **38.1** | 36.3 |

Table 6: Retraining with a curriculum may work better than fine-tuning with it, on a large, noisy dataset.

baseline) to the early curriculum phase (for exploration), and "easier-to-learn" examples with lower per-word loss to the late curriculum phase (for exploitation). Over iterations, a later-iteration curriculum schedules even easier examples than a previous iteration at late curriculum. The story happens reversely at early curriculum due to probability mass conservation. Figure 5 shows a similar story regarding per-word loss variance. So, curriculum optimization "regularizes" the curriculum and makes it easier-to-learn towards the end of CL.

These may be important for a small-capacity model to learn efficiently. The fact that the deep model is not improved as much means that 'clean' may have taken most of the headroom for deep models.

Meanwhile, according to Figure 6, each individual curriculum concentrates more on news in-domain examples as training progresses. Over iterations, bootstrapping makes the co-curriculum more news-domain aware. Due to the use of the denoising curriculum, data in curriculum becomes cleaner, too. So, although the co-curriculum schedules data from hard to easier-to-learn, which seems opposite to the general CL, it also schedules data from less in-domain to cleaner and more in-domain, which captures the spirit of CL.

### 5.5 Retraining

On Paracrawl, retraining NMT with co-curriculum improves dynamic fine-tuning, as shown in Table 6 (P9 vs. P8): +0.6 BLEU on IWSLT15 and +1.0 BLEU on WMT14. On WMT14 training data, retraining (W9) seems to perform similarly to fine-tuning on a warmed-up model (W8): +0.3 on IWSLT15 but -0.2 on WMT14; We speculate that this may be due to the smaller WMT training data size.

### 5.6 Dynamic vs. Static Data Selection

Co-curricular learning is dynamic. How does being dynamic matter? Table 7 shows that fine-tuning on the top 10% data[8] static selection (P10, W10) gives good improvements over baselines P1, W1, but co-curriculum (P9, W9) may do better.

[8] This is the ratio where the pace function reaches the floor value in training (see end of Section 5.1).

| Model | Test Bleu | |
|---|---|---|
| | IWSLT15 | WMT14 |
| P1: $\mathcal{C}_{\text{random}}$ | 34.6 | 31.6 |
| P9: Curriculum (Dynamic) | **37.9** | **35.6** |
| P10: Static selection | 36.8 | 34.6 |
| W1: $\mathcal{C}_{\text{random}}$ | 36.5 | 35.0 |
| W9: Curriculum (Dynamic) | **38.1** | 36.3 |
| W10: Static selection | 37.4 | 36.2 |

Table 7: Curriculum learning works slightly better than fine-tuning a warmed-up model with a top static selection.

| Model | Test Bleu | |
|---|---|---|
| | IWSLT15 | WMT14 |
| P9: Retrain with curriculum | **37.9** | **35.6** |
| P11: Retrain with static sel. | 37.1 | 34.6 |
| W9: Retrain with curriculum | **38.1** | **36.3** |
| W11: Retrain on static sel. | 34.0 | 31.7 |

Table 8: Curriculum learning works better than retraining with a static, top selection, especially when the training dataset is small.

This confirms findings by (van der Wees et al., 2017).

What if we retrain on the static data, too? In Table 8, W11 vs. W9 shows that retrained models on the static data is far behind for the WMT14 training – top 10% selection has only 4 million examples. On Paracrawl, P11 vs. P9 are closer, but retraining on co-curriculum performs still better. In all cases, co-curricular learning gives the best results. We may tune the static selection for better results, but then it is the exact point of CL, to evolve the data re-weighting without the need of a hard cutoff on selection ratio.

### 5.7 Discussion

**Evidence of data-quality transfer**. Figure 7 visualizes that CL in one domain (e.g., web) may enable CL in another. This is the foundation of our proposed method. To draw the figure, using a random sample of 2000 pairs from WMT training data and some additional in-domain parallel data, we sort examples by tightening the selection ratio according to a true web curriculum. The web curve shows the co-relation between selection ratio and data relevance to web. The same data order appears to yield increasing relevance to other domains, too, with bigger effect on a closer 'news' domain, but smaller effect on 'patent' and 'short' (sentences).

**Regularizing data without a teacher**. The analysis in Section 5.4 shows that the denoising scoring function and its bootstrapped versions tend to regularize the late curriculum and make the scheduled data easier for small models to learn on. One potential further application of this data property may be in learning a multitask curriculum where regular data may be helpful for multiple task distributions to work together in the same model. This has been achieved by knowledge distillation in existing research (Tan et al., 2019), by regularizing data with a teacher – We could instead regularize data by example selection, without a teacher. We leave this examination for future research.

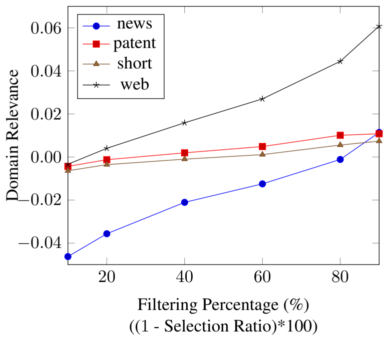

Figure 7: Curriculum learning in one domain may enable curriculum learning in another.

**Pace function hyper-parameters**. In experiments, we found that data-discarding pace functions seem to work best when they simultaneously decay down to their respective floors. Adaptively adjusting them seems an interesting future work.

## 6 Conclusion

We present a co-curricular learning method to make domain-data selection work better on noisy data, by dynamically composing it with clean-data selection. We show that the method improves over either constituent selection and their static combination. We further refine the co-curriculum with an EM-style optimization procedure and show its effectiveness, in particular on small-capacity models. In future, we would like to extend the method to handle more than two curricula objectives.

## Acknowledgments

The authors would like to thank Yuan Cao for his help and advice, the three anonymous reviewers for their constructive reviews, Melvin Johnson for early discussions, Jason Smith, Orhan Firat, Macduff Hughes for comments on an earlier draft, and Wolfgang Macherey for his early work on the topic.

## References

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

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26 th International Conference on Machine Learning*, page 8696, Montreal, Canada.

Peter F. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Comput. Linguist.*, 19(2):263–311.

Christian Buck, Jannis Bulian, Massimiliano Ciaramita, Wojciech Pawe Gajewski, Andrea Gesmundo, Neil Houlsby, and Wei Wang. 2018. Ask the right questions: Active question reformulation with reinforcement learning.

Boxing Chen and Fei Huang. 2016. Semi-supervised convolutional networks for translation adaptation with tiny amount of in-domain data. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning (CoNLL)*, pages 314–323.

Boxing Chen, Roland Kuhn, George Foster, Colin Cherry, and Fei Huang. 2016. Bilingual methods for adaptive training data selection for machine translation. In *AMTA*.

A. P. Dempster, N. M. Laird, and D. B. Rubin. 1977. Maximum likelihood from incomplete data via the em algorithm. *JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B*, 39(1):1–38.

Markus Freitag and Yaser Al-Onaizan. 2016. Fast domain adaptation for neural machine translation. *CoRR*, abs/1612.06897.

Dan Hendrycks, Mantas Mazeika, Duncan Wilson, and Kevin Gimpel. 2018. Using trusted data to train deep networks on labels corrupted by severe noise. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 10456–10465. Curran Associates, Inc.

Marcin Junczys-Dowmunt. 2018. Dual conditional cross-entropy filtering of noisy parallel corpora. In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 901–908, Belgium, Brussels. Association for Computational Linguistics.

Huda Khayrallah and Philipp Koehn. 2018. On the impact of various types of noise on neural machine translation. *CoRR*, abs/1805.12282.

Tom Kocmi and Ondřej Bojar. 2017. Curriculum learning and minibatch bucketing in neural machine translation. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 379–386. INCOMA Ltd.

Philipp Koehn, 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*, Belgium, Brussels. Association for Computational Linguistics.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75. Association for Computational Linguistics.

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

Saab Mansour, Joern Wuebker, and Hermann Ney. 2011. Combining translation and language model scoring for domain-specific data filtering. In *International Workshop on Spoken Language Translation*, pages 222–229.

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in translation: Contextualized word vectors. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 6294–6305. Curran Associates, Inc.

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

Emmanouil Antonios Platanios, Otilia Stretcu, Graham Neubig, Barnabás Póczos, and Tom M. Mitchell. 2019. Competence-based curriculum learning for neural machine translation. In *2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics.

Hassan Sajjad, Nadir Durrani, Fahim Dalvi, Yonatan Belinkov, and Stephan Vogel. 2017. Neural machine translation training in a multi-domain scenario. *arXiv preprint arXiv:1708.08712v2*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving Neural Machine Translation Models with Monolingual Data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages

86–96, Berlin, Germany. Association for Computational Linguistics.

Xu Tan, Yi Ren, Di He, Tao Qin, and Tie-Yan Liu. 2019. Multilingual neural machine translation with knowledge distillation. In *International Conference on Learning Representations*.

Brian Thompson, Huda Khayrallah, Antonios Anastasopoulos, Arya D. McCarthy, Kevin Duh, Rebecca Marvin, Paul McNamee, Jeremy Gwinnup, Tim Anderson, and Philipp Koehn. 2018. Freezing subnetworks to analyze domain adaptation in neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 124–132. Association for Computational Linguistics.

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

Rui Wang, Masao Utiyama, and Eiichiro Sumita. 2018a. Dynamic sentence sampling for efficient training of neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 298–304, Melbourne, Australia. Association for Computational Linguistics.

Wei Wang, Taro Watanabe, Macduff Hughes, Tetsuji Nakagawa, and Ciprian Chelba. 2018b. Denoising neural machine translation training with trusted data and online data selection. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 133–143. Association for Computational Linguistics.

Marlies van der Wees, Arianna Bisazza, and Christof Monz. 2017. Dynamic data selection for neural machine transaltion. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1400–1410.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, ukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Xuan Zhang, Gaurav Kumar, Huda Khayrallah, Kenton Murray, Jeremy Gwinnup, Marianna J. Martindale, Paul McNamee, Kevin Duh, and Marine Carpuat. 2018. An empirical exploration of curriculum learning for neural machine translation. *CoRR*, abs/1811.00739.

Xuan Zhang, Pamela Shapiro, Gaurav Kumar, Paul McNamee, Marine Carpuat, and Kevin Duh. 2019. Curriculum learning for domain adaptation in neural machine translation. In *2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics*.