A Two-Stage Curriculum Training Framework for NMT

Anonymous ACL submission

Abstract

Neural Machine Translation (NMT) models are typically trained on heterogeneous data that are concatenated and randomly shuffled. Curriculum training aims to present the data to the NMT systems in a meaningful order. In this work, we introduce a two-stage curriculum training framework for NMT where we fine-tune a base NMT model on subsets of data, selected by both deterministic scoring using pre-trained methods and online scoring that consider prediction scores of the emerging NMT model. Through extensive experiments on six language pairs comprising low- and high-resource languages from WMT'21, we have shown that our curriculum strategies consistently demonstrate better quality (up to +2.2 BLEU improvement) and faster convergence (approximately 50% fewer updates).

1 Introduction

The notion of a curriculum came from the human learning experience; we learn better and faster when the learnable examples are presented in a meaningful sequence rather than a random order (Newport, 1990). In the case of machine learning, curriculum training hypothesizes presenting the data samples in a meaningful order to machine learners during training such that it imposes structure in the task of learning (Bengio et al., 2009).

In recent years, Neural Machine Translation (NMT) has shown impressive performance in high-resource settings (Popel et al., 2020). Typically training data of the NMT systems are a heterogeneous collection from different domains, sources, topics, styles, and modalities, and of different quality and linguistic difficulty levels. However, not all of them may be useful, some examples may be redundant, and some data might even be noisy and detrimental to the final NMT system performance (Khayrallah and Koehn, 2018). So, NMT systems have the potential to benefit greatly from curriculum learning in terms of both speed and quality.

In this work, we propose a *two-stage* curriculum training framework for NMT — *model warm-up* and *model fine-tuning*. We initially train a base model in the warm-up stage on all available data. In the fine-tuning, we adapt the base model on subsets of the data based on data quality and/or usefulness at the current state of the model. We explore two sets of data selection curriculum strategies — *deterministic* and *online*. The deterministic curriculum uses external measures which require pretrained models for selecting the data subset at the beginning and continues training on the selected subset. In contrast, the online curriculum dynamically selects a subset of the data for each epoch without requiring any external measure. Specifically, it leverages the prediction scores of the emerging NMT model. Our online curriculum resembles self-paced learning (Kumar et al., 2010) which uses the emerging model hypothesis to select samples.

For picking the subset of the data in the online curriculum, we investigate two approaches of *data-selection window* – static and dynamic. Even though the size of the data-selection window is fixed throughout the training in the static approach, the samples in the selected subset vary from epoch to epoch due to the change in their prediction scores. In the dynamic approach, we either expand or shrink the data-selection window.

Experiments on 6 language pairs (12 translation directions) comprising low- and high-resource languages from WMT'21 demonstrate better performance compared to the baseline trained on all data (up to +2.2 BLEU). We observe bigger gains for the high-resource pairs compared to the low-resource ones. Interestingly, we find that the online curriculum approaches perform on par with the deterministic approaches while not using any external pretrained models. Our proposed curriculum training approaches not only exhibit better performance but also converge much faster requiring approximately 50% fewer updates compared to the baseline.

2 Background

Curriculum learning Inspired by human learners, Elman (1993) argues that optimization of neural network training can be accelerated by gradually increasing the difficulty of the concepts. Bengio et al. (2009) were the first to use the term "curriculum learning" to refer to the easy-to-hard training strategies in the context of machine learning. Using an easy-to-hard curriculum based on increasing vocabulary size in language model training, they achieved performance improvement. Recent work (Jiang et al., 2015; Hacohen and Weinshall, 2019; Zhou et al., 2020a) shows that manipulating the sequence of training data can improve both training efficiency and model accuracy. Several studies show the effectiveness of the difficulty-based curriculum learning in a wide range of NLP tasks (Cirik et al., 2016; Liu et al., 2018).

Curriculum learning in NMT The difficulty-based curriculum in NMT was first explored by Kocmi and Bojar (2017). Later, Zhang et al. (2018) adopt a probabilistic view of curriculum learning and investigate a variety of difficulty criteria based on human intuition, e.g., sentence length and word rarity. Platanios et al. (2019) connect the appearance of difficult samples with NMT model competence. Liu et al. (2020) propose a norm-based curriculum learning method based on the norm of word embedding. Zhou et al. (2020b) use a pre-trained language model to measure the word-level uncertainty. Xu et al. (2020) explore the effectiveness of curriculum learning for low-resource NMT.

Data selection strategy for NMT Joty et al. (2015) use domain adaptation by penalizing sequences similar to the out-domain data. Wang et al. (2018) propose a curriculum-based data selection strategy by using an additional trusted clean dataset to calculate noise level of a sample. Kumar et al. (2019) use reinforcement learning to learn a denoising curriculum jointly with the NMT system. Jiao et al. (2020) identify the inactive samples during training and re-label them for later use. Wang et al. (2021) find gradient alignments between a clean dataset and the training data to mask out noisy data.

Domain specific fine-tuning In a successful line of research NMT models are first trained on a large general-domain bitext and then fine-tuned on small in-domain data (Luong and Manning, 2015; Zoph et al., 2016). van der Wees et al. (2017) gradually decrease the training data size to a cleaner subset of the data estimated by some external scorers.

Summary Most curriculum learning methods in NMT focus on addressing the batch selection issue from the beginning of the training by using hand-designed heuristics (Zhao et al., 2020). In contrast, our proposed two-stage curriculum training framework for NMT fine-tunes the base model from the warm-up stage on a selected subset of data. Our curriculum training framework resembles the formal education system as discussed in §6.4.

3 Proposed Framework

Let s and t denote the source and target language respectively, and $\mathcal{D}_g = \{(x_i, y_i)\}_{i=1}^N$ denote the general-domain parallel training data containing N sentence pairs with x_i and y_i coming respectively from s and t languages. Also, let $\mathcal{D}_d \subseteq \mathcal{D}_g$ be the in-domain parallel training data and $\mathcal{M}$ is an NMT model that can translate sentences from s to t. The overall training objective of the NMT model is to minimize the total loss of the training data:

$$\mathcal{J}(\theta) = \sum_{i=1}^{N} \mathcal{L}(x_i, y_i, \theta) = \sum_{i=1}^{N} -\log P_\theta(y_i|x_i) \quad (1)$$

where $P_\theta(y_i|x_i)$ is the sentence-level translation probability of the target sentence y_i for the source sentence x_i with θ being the parameters of $\mathcal{M}$.

We propose a *two-stage* training curriculum where in the *model warm-up* stage we train $\mathcal{M}$ on general domain bitext $\mathcal{D}_g$ for K number of gradient updates; K is generally smaller than the total number of updates $\mathcal{M}$ requires for convergence. Then in *model fine-tuning* stage, we fine-tune $\mathcal{M}$ on the in-domain bitext $\mathcal{D}_d$ till it converges. Based on the intuition *"not all of the training data are useful or non-redundant, some samples might be irrelevant or even detrimental to the model"*, we hypothesize that there exists a $\mathcal{D}_s \subset \mathcal{D}_d$, fine-tuning on which $\mathcal{M}$ will exhibit an improved performance.

Our goal is to design a ranking of the training samples which will eventually help us extract $\mathcal{D}_s$ from $\mathcal{D}_d$. For this, we investigate two sets of data selection curriculum strategies – *deterministic* and *online*. Both strategies require a measure of data quality and/or usefulness at the current state of the model to extract $\mathcal{D}_s$. While the deterministic curriculum uses external measures that require pre-trained models, the online curriculum leverages the prediction scores of the emerging NMT models.

3.1 Deterministic Curriculum

In this strategy, we select a $\mathcal{D}_s \subset \mathcal{D}_d$ initially and do not change it during the model fine-tuning stage. We first score each sample in $\mathcal{D}_d$ using an external bitext scoring method. We experiment with three scoring methods as described below.

• **LASER** This approach utilizes the Language-Agnostic SEntence Representations (`LASER`) toolkit (Artetxe and Schwenk, 2019), which gives multilingual sentence representations using an encoder-decoder architecture trained on a parallel corpus . We use the sentence representations to *score the similarity* of a bitext using Cross-Domain Similarity Local Scaling (CSLS), which performs better than other similarity metrics in reducing the hubness problem (Conneau et al., 2017).

$$\mathcal{S}core_{\text{laser}}(x_i, y_i) = \text{CSLS}(\texttt{LASER}(x_i), \texttt{LASER}(y_i)) \quad (2)$$

Chaudhary et al. (2019) showed benefits of `LASER`-based ranking for low-resource corpus filtering.

• **Dual Conditional Cross-Entropy (DCCE)** Junczys-Dowmunt (2018) proposed this method, which requires two inverse translation models – one forward model (f) and one backward (b) model trained on the same parallel corpus. It then finds the score of a bitext (x_i, y_i) by taking the maximal symmetric agreement of the two models which exploits the conditional cross-entropy (H).

$$\mathcal{S}core_{\text{dcce}}(x_i, y_i) = |H_f - H_b| + \frac{1}{2}(H_f + H_b) \quad (3)$$
$$\text{where } H_f = -\log P_{\theta_f}(y_i|x_i);\ H_b = -\log P_{\theta_b}(x_i|y_i)$$

The absolute difference between the conditional cross-entropy in Eq.3 measures the agreement between the two conditional probability distributions. If the sentences in a bitext are equally probable (good) or equally improbable (bad/noisy), this part of the equation will have a low score. To differentiate between these two scenarios, we need the average cross-entropy score which scores higher for improbable sentence pairs.

• **Modified Moore-Lewis (MML)** MML ranks the bitext pairs based on domain relevance by calculating cross-entropy difference scores (Moore and Lewis, 2010; Axelrod et al., 2011). For this, we need to train four language models (LM): **in**- and **gen**eral-domain LMs in both **s**ource and **t**arget languages. Then we find the MML score of a bitext pair (x_i, y_i) as follows:

$$\begin{aligned}\mathcal{S}core_{\text{mml}}(x_i, y_i) = (H_{s,in}(x_i) - H_{s,gen}(x_i)) \\ + (H_{t,in}(y_i) - H_{t,gen}(y_i)) \quad (4)\end{aligned}$$
$$\text{where } H_{b,C}(z) = -\log P^{\text{LM}}_{b,C}(z)$$

Algorithm 1 Deterministic Curriculum Strategy

Input : General domain corpus $\mathcal{D}_g$, in-domain corpus $\mathcal{D}_d \subseteq \mathcal{D}_g$, external pretrained bitext scorer $\mathcal{S}$

Output : A trained translation model

1. `// model warm-up stage`
   Train a base model $\mathcal{M}$ on general domain corpus $\mathcal{D}_g$ for K number of updates
2. `// model fine-tuning stage`
   (a) Use $\mathcal{S}$ to score each bitext in $\mathcal{D}_d$
   (b) Rank $\mathcal{D}_d$ based on these scores
   (c) Find $\mathcal{D}_s \subset \mathcal{D}_d$ by selecting top $p\%$ of $\mathcal{D}_d$
   (d) **for** *n_epochs* **do**
   Fine-tune $\mathcal{M}$ on $\mathcal{D}_s$

end

Here, $b \in \{s, t\}$ refers to the bitext side and $C \in \{in, gen\}$ refers to the corpus domain. In our experiments, we use the *newscrawl* data as in-domain and *commoncrawl* combined with newscrawl data as general-domain for training the LMs.

LASER and DCCE assign scores based on denoising curriculum (*i.e.,* higher rank for good translation and lower rank for noisy ones) while MML performs domain similarity curriculum on the given data. After scoring each pair $(x_i, y_i) \in \mathcal{D}_d$ by any of the above methods, we rank $\mathcal{D}_d$ based on the scores, and pick $\mathcal{D}_s \subset \mathcal{D}_d$ by selecting top $p\%$ pairs as the better subset to fine-tune the base model $\mathcal{M}$ on $\mathcal{D}_s$. Algorithm 1 presents a pseudo-code of our deterministic curriculum strategy.

3.2 Online Curriculum

Unlike deterministic curriculum, in this strategy the selected subset $\mathcal{D}_s$ changes dynamically in each epoch of the fine-tuning stage through instantaneous feedback from the current model. Specifically, we rank the samples by leveraging the prediction scores from the emerging NMT model which assigns a probability to each token in the target sentence y_i. We then take the average of the token-level probabilities to get the sentence-level probability score which is regarded as the prediction score for the bitext pair (x_i, y_i). Formally,

$$P_\theta(y_i|x_i) = \frac{1}{\ell}\sum_{t=1}^{\ell} p_\theta(y_{i,t}|y_{i,<t}, x_i) \quad (5)$$

This bitext prediction score indicates the confidence of the model to generate the target sentence y_i from the source sentence x_i. Intuitively, if the model can predict the target sentence of a sample

Algorithm 2 Online Curriculum Strategy

Input : General corpus $\mathcal{D}_g$, in-domain corpus $\mathcal{D}_d \subseteq \mathcal{D}_g$

Output : A trained translation model

1. `// model warm-up stage`

Train a base model $\mathcal{M}$ on general domain corpus $\mathcal{D}_g$ for K number of updates

2. `// model fine-tuning stage`

for *n_epochs* **do**

(*a*) Find prediction score for each bitext in $\mathcal{D}_d$

(*b*) Rank $\mathcal{D}_d$ based on these scores

(*c*) Find $\mathcal{D}_s \subset \mathcal{D}_d$ by picking a data-selection window

(*d*) Fine-tune $\mathcal{M}$ on $\mathcal{D}_s$

end

with higher confidence, it indicates that the sample is *too easy* for the model and might not contain useful information to improve the model further at that state. On the other hand, if a target sentence is predicted with lower confidence, it indicates that the sample might be *too hard* for the model at that state or it might be a noisy sample. Subsequently, including such hard or noisy samples in training might degrade the model performance.

Algorithm 2 presents the pseudo-code of our proposed online curriculum strategy. After warm-up stage, we fine-tune $\mathcal{M}$ for n_epochs on $\mathcal{D}_s$ which is selected in every epochs. In the beginning of each fine-tuning epoch, we find the prediction score for each bitext pair in $\mathcal{D}_d$. We rank $\mathcal{D}_d$ based on these scores and select $\mathcal{D}_s \subset \mathcal{D}_d$ by picking a *data-selection window* in the ranked in-domain data. Finally, we fine-tune $\mathcal{M}$ on $\mathcal{D}_s$ for that epoch. We present the conceptual demonstration of our online curriculum strategy in Figure 1. For picking the data-selection window in the ranked $\mathcal{D}_d$, we investigate two methods:

• **Static Data-selection Window** Here in each epoch, we discard a constant amount of easy and hard samples from $\mathcal{D}_d$ based on the prediction scores and select the rests as $\mathcal{D}_s$. Even though in this method the size of the data-selection window is fixed through out the fine-tuning stage, unlike deterministic strategy the samples in $\mathcal{D}_s$ varies from epoch to epoch due to the change in their prediction scores by the emerging $\mathcal{M}$.

• **Dynamic Data-selection Window** Unlike the static approach, here we change the data-selection window size in subsequent epochs. This can be done in two ways:

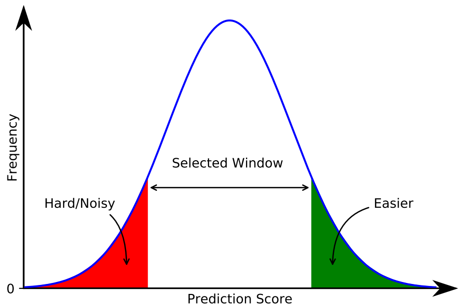

Figure 1: Conceptual demonstration of online curriculum. We rank the bitext pairs based on the prediction scores of the emerging model and pick a data-selection window which discards easy and hard/noisy ones.

(i) *Expansion:* Begin fine-tuning with smaller window ($|\mathcal{D}_s| << |\mathcal{D}_d|$) and gradually increase the window to a maximum size λ_{max}.

(ii) *Shrink:* Begin fine-tuning with a larger window ($|\mathcal{D}_s| \sim |\mathcal{D}_d|$) and gradually decrease the window to a minimum size λ_{min}.

To change the data-selection window size, we experiment with *linear scheduler* which can be regarded as a function $\lambda(t)$ to map the current training epoch t to a scalar. This scalar value will be the data-selection window size at epoch t. Formally,

$$\lambda_{\text{exp}}(t) = \begin{cases} \lambda_{init} + l_{inc} * t, & \text{if } \lambda_{\text{exp}}(t) < \lambda_{max} \\ \lambda_{max}, & \text{otherwise} \end{cases}$$
$$\lambda_{\text{shr}}(t) = \begin{cases} \lambda_{init} - l_{dec} * t, & \text{if } \lambda_{\text{shr}}(t) > \lambda_{min} \\ \lambda_{min}, & \text{otherwise} \end{cases} \tag{6}$$

Where λ_{init} is the initial window size which is smaller for *expansion* and larger for *shrink*, and l_{inc}, l_{dec} are the hyperparameters of the schedulers.

4 Experimental Setup

Datasets We conduct experiments on six language pairs: three high-resource including English (En) to/from German (De), Hungarian (Hu) and Estonian (Et), and three low-resource including English (En) to/from Hausa (Ha), Tamil (Ta) and Malay (Ms). We use the dataset provided in WMT 2021[1] — De and Ha are from *News shared task*, while the remaining four pairs are from *Large-Scale Multilingual MT shared task*. For En↔De, we use newstest2019 as validation set and report test results on newstest2020. For En↔Ha, we randomly split the provided dev set into valid and test set. For the other language pairs, we use the official evaluation data (dev and devtest) as validation and test sets. Table 1 presents the dataset

[1] http://www.statmt.org/wmt21/

Pair	Train		Validation	Test
	All-data	In-domain		
En-De	89,893,260	2,152,577	1997	1418
De-En	89,893,260	2,152,577	2000	785
En-Hu	53,219,023	647,106	997	1012
En-Et	19,685,308	869,537	997	1012
En-Ms	1,694,311	–	997	1012
En-Ta	1,064,032	–	997	1012
En-Ha	685,780	–	1000	1000

Table 1: Number of sentence pairs for each dataset after cleaning and deduplication.

statistics after cleaning and deduplication. For high-resource pairs, we consider formal bitext corpora sources as in-domain ($\mathcal{D}_d \subset \mathcal{D}_g$), while for low-resource pairs, we do not differentiate between general-domain and in-domain corpus ($\mathcal{D}_d := \mathcal{D}_g$).

Model Settings We use the Transformer (Vaswani et al., 2017) implementation in Fairseq (Ott et al., 2019); details of our model architecture settings are given in Appendix B. We use sentencepiece library[2] to learn joint Byte-Pair-Encoding (BPE) of size 32,000 and 16,000 for En↔De and En↔Ha, respectively. For other language pairs, we use official sentencepiece model provided in *Large-Scale Multilingual MT shared task*. We filter out bitext with length longer than 250 tokens during training. All experiments are evaluated using SacreBLEU (Post, 2018).

For LM training in modified Moore-Lewis method (§3.1), we use the implementation in Fairseq. For in-domain LM training, we use 5M sentences from newscrawl, while we combine 10M commoncrawl data with newscrawl totaling 15M sentences to train the general-domain LM.

Baselines We compare our methods with the **converged model**, which is a standard NMT model trained on all the general-domain data ($\mathcal{D}_g$) until convergence. Additionally, we compare both the deterministic and online curriculum approaches with the **traditional fine-tuning** where we fine-tune the base model from the warm-up stage with all the in-domain train data ($\mathcal{D}_d$) until convergence.

5 Results

The main results for the low- and high-resource languages are shown in Tables 2 and 3, respectively. For low-resource languages, we train the warm-up stage models for 20K updates, while the converged models are trained for 50K updates. For high-resource languages, we train for 50K and 100K updates for the warm-up and converged models, respectively. In traditional fine-tuning (*All Data* row in the Tables), we use all the available in-domain data ($\mathcal{D}_d$) in each fine-tuning epoch. On the other hand, for both deterministic and online curricula, we use at most 40% of the available in-domain data ($\mathcal{D}_s \subset \mathcal{D}_d$) in each fine-tuning epoch. We discuss a detailed performance comparison of traditional fine-tuning with *Converged Model* in Appendix C.

5.1 Performance of Deterministic Curricula

First, we consider the performance of deterministic curriculum approaches on low-resource languages. From Table 2, we see that training on the data subset ($\mathcal{D}_s$) selected by `LASER` outperforms the baseline (*Converged Model*) on five out of six translation tasks with a +2.2 BLEU gain in Ha-En. For the other two scoring methods, dual conditional cross-entropy (`DCCE`) and modified Moore-Lewis (`MML`), we also see a better or similar performance on 5/6 translation tasks. Compared to the traditional fine-tuning (*All Data* row in Table 2), the deterministic approaches perform better in most of the tasks - on average +0.5, +0.4, +0.2 BLEU gains for `LASER`, `DCCE`, and `MML`, respectively.

In Table 3, we see a similar trend of improved performance for the deterministic curricula over the converged model on high-resource languages. Specifically, data selection by utilizing the scoring of both `LASER` and `DCCE` performs better on four out of six translation tasks, while the `MML`-based method achieves a better performance on three tasks. The margin of improved performances for the high-resource languages are higher compared to the low-resource languages: +1.4, +0.9, +0.7 BLEU gains on average for `DCCE`, `LASER`, and `MML`, respectively over the baseline. If we compare with traditional fine-tuning (*All In-domain Data* row in Table 3), the deterministic curriculum approaches perform better in most of the tasks - on average +1.2, +0.8, +0.4 BLEU scores better for `DCCE`, `LASER`, and `MML`, respectively.

To observe the better performance of the deterministic curriculum approaches more clearly, we fine-tune the base model with different percentages of ranked data selected by the bitext scoring methods. Figure 2 shows the results. We observe that there exist multiple subsets of data ($\mathcal{D}_s \subset \mathcal{D}_d$), fine-tuning the base model from warm-up stage

[2] https://github.com/google/sentencepiece

Type	Setting	%data-used in each ep.	En-Ha →	En-Ha ←	En-Ms →	En-Ms ←	En-Ta →	En-Ta ←
	Warm-up Stage Model	100%	13.5	14.7	30.8	27.3	8.6	15.6
Baseline	Converged Model	100%	14.3	15.3	31.4	27.9	8.9	15.8
		Warm-up Stage Model Fine-tuning						
	All Data	100%	14.4 +0.1	15.6 +0.3	31.5 +0.1	28.0 +0.1	8.8 −0.1	15.7 −0.1
Det. Curricula	LASER	40%	14.6 +0.3	**17.5** +2.2	31.7 +0.3	28.2 +0.3	8.7 −0.2	15.9 +0.1
	Dual Cond. CE (DCCE)	40%	14.3 +0.0	16.3 +1.1	31.4 +0.0	28.2 +0.3	8.5 −0.4	16.0 +0.2
	Mod. Moore-Lewis (MML)	40%	**14.9** +0.6	15.6 +0.3	31.6 +0.2	28.1 +0.2	9.0 +0.1	15.7 −0.1
Online Curricula	Static Window	40%	14.7 +0.4	16.1 +0.8	31.6 +0.2	28.3 +0.4	**9.1** +0.2	**16.2** +0.4
	Dynamic Window							
	Expansion	<40%	14.8 +0.5	16.6 +1.3	**31.8** +0.4	**28.4** +0.5	9.0 +0.1	16.0 +0.2
	Shrink	<40%	14.7 +0.4	15.9 +0.6	31.4 +0.0	28.3 +0.4	8.8 −0.1	16.0 +0.2
Det. + Online	Hybrid	15-20%	14.7 +0.4	16.4 +1.1	31.5 +0.1	28.2 +0.3	**9.1** +0.2	15.9 +0.1

Table 2: Main results for **low-resource** languages. Here, the data-percentage represents *general-domain data* ($\mathcal{D}_g$) and we do not differentiate between general-domain and in-domain corpus ($\mathcal{D}_d \coloneqq \mathcal{D}_g$). Subscript values denote the BLEU score differences from the respective converged model.

Type	Setting	%data-used in each ep.	En-De →	En-De ←	En-Hu →	En-Hu ←	En-Et →	En-Et ←
	Warm-up Stage Model	100%+OOD	34.9	41.0	33.9	36.0	35.7	37.1
Baseline	Converged Model	100%+OOD	36.1	41.2	35.9	**36.7**	36.7	**38.2**
		Warm-up Stage Model Fine-tuning						
	All In-domain Data	100%	36.5 +0.4	40.7 −0.5	35.7 −0.2	35.5 −1.2	37.6 +0.9	37.4 −0.8
Det. Curricula	LASER	40%	37.6 +1.5	42.4 +1.2	36.0 +0.1	35.9 −0.8	37.6 +0.9	37.8 −0.4
	Dual Cond. CE (DCCE)	40%	37.9 +1.8	43.0 +1.8	**36.4** +0.5	35.4 −1.3	**38.1** +1.4	37.3 −0.9
	Mod. Moore-Lewis (MML)	40%	37.1 +1.0	41.7 +0.6	35.8 −0.1	35.2 −1.5	37.3 +0.6	37.4 −0.8
Online Curricula	Static Window	40%	37.3 +1.2	41.4 +0.2	36.1 +0.2	35.4 −1.3	37.9 +1.2	37.7 −0.5
	Dynamic Window							
	Expansion	<40%	37.3 +1.2	41.3 +0.1	36.2 +0.3	35.4 −1.3	38.0 +1.3	37.8 −0.4
	Shrink	<40%	37.0 +0.9	41.2 +0.0	36.0 +0.1	35.7 −1.0	**38.1** +1.4	37.6 −0.6
Det. + Online	Hybrid	15-20%	**38.1** +2.0	**43.3** +2.1	36.1 +0.2	35.6 −1.1	37.9 +1.2	37.3 −0.9

Table 3: Main results for **high-resource** languages. Here, the data-percentage represents only *In-domain data* ($\mathcal{D}_d$) from Table 1 and *100%+OOD* denotes *All-data* ($\mathcal{D}_g$). Subscript values denote the BLEU score differences from respective converged model.

on those subsets exhibit a better performance compared to the baseline (*Converged Model*) and traditional fine-tuning. For De-En, traditional fine-tuning (on 100% data) reduces the BLEU score by 0.3 from the base model, while most of the subsets selected by the deterministic curricula exhibit improved performances. For Hu-En, traditional fine-tuning reduces the performance of the base model by 0.5 BLEU. Unlike De-En, here we do not find a subset by the deterministic curricula which improves the performance of the base model.

5.2 Performance of Online Curricula

Our online curriculum approaches perform on par with the deterministic curricula for both low- and high-resource languages as shown in Tables 2 and 3, respectively. Unlike deterministic, here we exploit the emerging model's prediction scores without using any external pretrained scoring methods. In our static window approach, we discard the top 30% and bottom 30% sentence pairs from the ranked $\mathcal{D}_d$ and fine-tune the base model on the remaining 40% data ($\mathcal{D}_s$). The selected data in $\mathcal{D}_s$ vary dynamically from epoch to epoch due to the change in the prediction scores of the emerging model. From the results (Tables 2, 3), we notice that the data-selection by static window method outperforms the baseline (*Converged Model*) on ten out of twelve translation tasks and the BLEU scores are comparable to the deterministic curriculum approaches.

In our dynamic window approach, we either expand or shrink the window size, where the selected window is restricted to the range of 30% to 70% of the ranked $\mathcal{D}_d$, i.e., $\mathcal{D}_s$ is at most 40% of $\mathcal{D}_d$. In window expansion, we start $\mathcal{D}_s$ with 10% of $\mathcal{D}_d$ and linearly increase it to 40% in the subsequent epochs, while in the window shrink method we start $\mathcal{D}_s$ with 40% and linearly decrease to 10% of $\mathcal{D}_d$. With dynamic window expansion, we achieve slightly better (in range of +0.5 to +0.1 BLEU) or similar performance on 9 out of 12 translation tasks compared to the static window method. On

Type	Setting	%data-used in each ep.	En-De →	En-De ←
	Warm-up Model	100%	33.3	39.1
Baseline	Converged Model	100%	34.6	40.0
	Warm-Up Model Fine-tuning			
	All in-domain data	100%	34.0 −0.6	41.6 +1.6
Det. Curricula	LASER	40%	34.4 −0.2	43.2 +3.2
	Dual Cond. CE (DCCE)	40%	**35.1** +0.5	**44.4** +4.4
	Mod. Moore-Lewis (MML)	40%	34.5 −0.1	41.6 +1.6
Online Curricula	Static Window	40%	34.1 −0.5	41.9 +1.9
	Dynamic Window			
	Expansion	<40%	34.4 −0.2	42.2 +2.2
	Shrink	<40%	34.3 −0.3	42.0 +2.0

Table 4: Results for En↔De on **noisy ParaCrawl corpus** of 10M bitext pairs. Here, the data-percentage corresponds to all 10M bitext ($\mathcal{D}_g$) and $\mathcal{D}_d \coloneqq \mathcal{D}_g$. Subscript values denote the BLEU score difference from the respective converged model.

Scoring Method	Top data%	En-Ha →	En-Ha ←	En-Ms →	En-Ms ←	En-Ta →	En-Ta ←
LASER	10%	$14.1_{8.3}$	$17.3_{10.1}$	$30.9_{18.9}$	$27.9_{15.1}$	$8.1_{0.7}$	$15.8_{1.6}$
	40%	$14.6_{13.1}$	$17.5_{16.5}$	$31.7_{30.2}$	$28.2_{25.2}$	$8.7_{5.9}$	$15.9_{10.7}$
Dual Cond. CE	10%	$13.0_{1.3}$	$16.3_{8.0}$	$31.0_{18.4}$	$28.0_{15.5}$	$8.0_{0.0}$	$15.2_{0.2}$
	40%	$14.3_{12.9}$	$16.3_{15.3}$	$31.4_{29.5}$	$28.2_{25.0}$	$8.5_{5.3}$	$16.0_{11.0}$
Modified Moore-Lewis	10%	$14.4_{5.9}$	$15.1_{4.7}$	$31.8_{19.6}$	$27.9_{15.3}$	$8.5_{0.0}$	$15.2_{0.6}$
	40%	$14.9_{13.3}$	$15.6_{13.6}$	$31.6_{30.8}$	$28.1_{24.9}$	$9.0_{5.9}$	$15.7_{10.5}$

Table 5: Results for **fine-tuning vs. training from scratch** on top 10% and 40% of selected data ranked by three bitext scoring methods (§3.1). Main values denote the results of fine-tuning, while subscript values represent results when model is trained from a random state on the same data subset.

the other hand, the dynamic window shrink method performs slightly worse than window expansion in most of the translation tasks.

6 Discussion and Analysis

6.1 Hybrid Curriculum

To benefit from both deterministic and online curricula, we combine the two strategies. Specifically, we consider three subsets of data comprising of the top 50% of $\mathcal{D}_d$ ranked by each of the three bitext scoring methods in §3.1 and keep the common bitext pairs (intersection of three subsets). We then apply the static window data-selection curriculum on these bitext pairs, where we discard the top 10% and bottom 10% pairs (ranked by the emerging model's prediction scores) and fine-tune the base model on the remaining bitext. Depending on the language pairs, the data percentage for fine-tuning ($\mathcal{D}_s$) becomes 15-20% of $\mathcal{D}_d$. Despite a smaller subset of data for fine-tuning, performances of the hybrid curriculum strategy are better on 10 out of 12 translation tasks compared to the baseline (Table 2, 3). Notably, for En-De and De-En, the hybrid curriculum achieves +2.0 and +2.1 BLEU gains compared to the converged baseline model.

6.2 Performance on Noisy Data

We further evaluate our framework on noisy data. We randomly selected 10M bitext pairs from the En-De ParaCrawl corpus. We keep the experimental settings similar to §5 and present the results in Table 4. Fine-tuning on the data subset (D_s) selected by DCCE method outperforms the baseline (*Converged Model*) on both directions with a +4.4 BLEU gain in De-En. All the other deterministic and online curriculum methods perform better than the converged model on the De-En direction with a sizable margin. Compared to the traditional fine-tuning, all the curriculum methods perform better in both En to/from De.

6.3 Do We Need the Warm-up Stage?

For the online curricula, we exploit the model $\mathcal{M}$ for selecting $\mathcal{D}_s$ based on the prediction scores, while in the deterministic curricula, we do not use the emerging model for selecting the data subset. One might ask – do we need a base model in the deterministic curricula? Can we get rid of the warm-up stage? To answer these questions, we perform another set of experiments where we train $\mathcal{M}$ from a randomly initialized state on the top p% of the selected data (p ={10, 40}) ranked by the three bitext scoring methods (§3.1) and compare the results with the base model $\mathcal{M}$ fine-tuned on the same data subset. From the results in Table 5, we see that for all the tasks our proposed two-stage curriculum framework outperforms the training from the scratch method by a sizable margin.

6.4 Are All Data Useful Always?

Our proposed curriculum training framework uses all the data ($\mathcal{D}_g$) in the warm-up stage and then utilizes a subset of in-domain data ($\mathcal{D}_s$) in the fine-tuning stage. This resembles the formal education system where students first learn the general subjects with the same weights and later concentrate more on a selected subset of specialized subjects. The first stage teaches base knowledge which is useful in the later stage. We observe the same in our experiments. From Table 6, we see that the performance of the NMT model using only the in-domain data is worse than using all general-domain data (-8.1 BLEU on average). Moreover, the gains of our proposed framework in most of the translation tasks over the converged model which uses all the data throughout the training, suggests that not

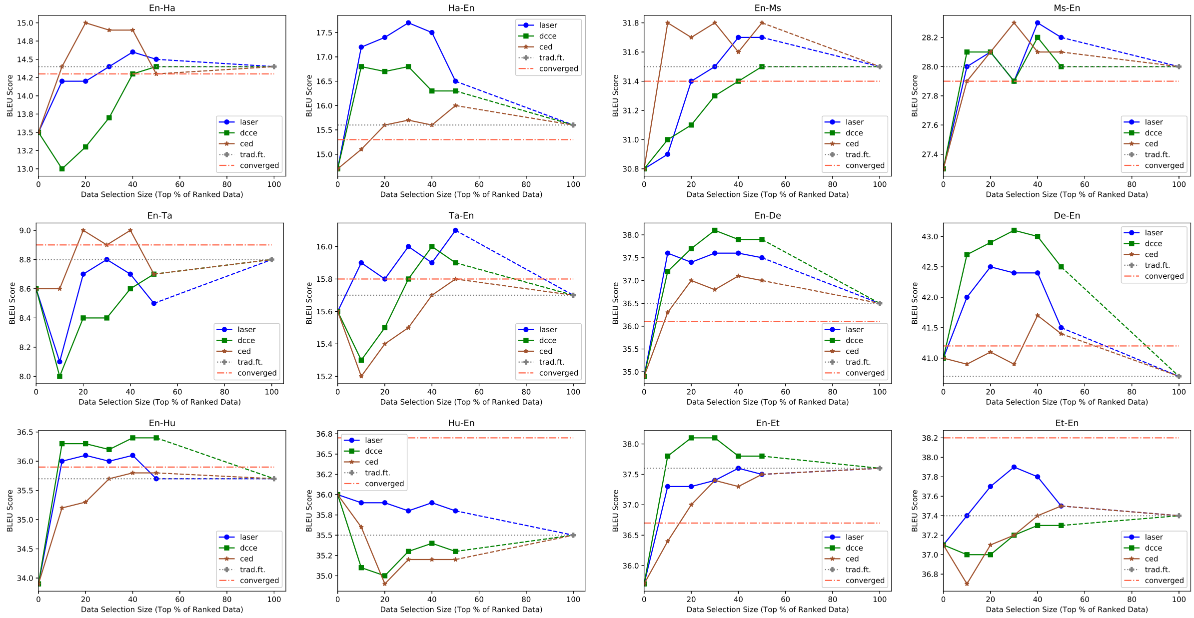

Figure 2: Fine-tuned *warm-up stage model* using different sizes of ranked data (deterministic curricula).

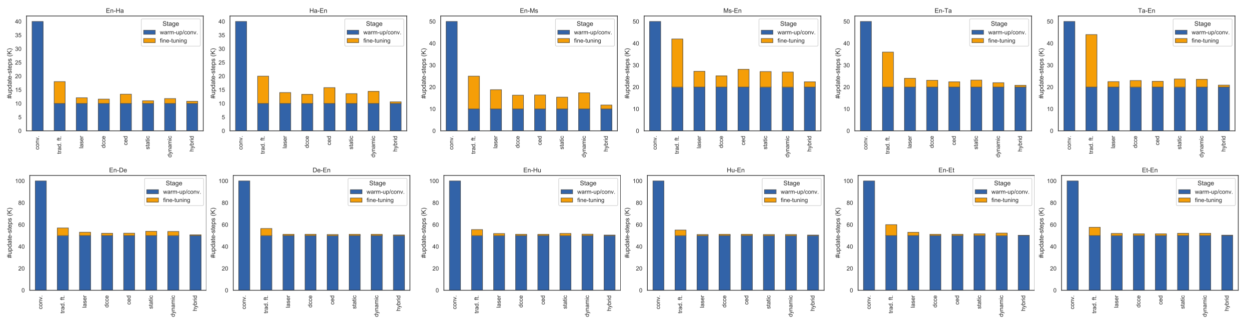

Figure 3: Number of update steps required for each setting of Tables 2, 3. We keep batch size same in each setting.

Corpus	En-De		En-Hu		En-Et	
	→	←	→	←	→	←
All-data	36.1	41.2	35.9	36.7	36.7	38.2
In-domain	32.6	33.5	25.5	23.6	30.6	30.3

Table 6: Results for high-resource languages on all-data ($\mathcal{D}_g$) vs. in-domain data ($\mathcal{D}_d$) when trained from scratch until convergence.

all data are useful all the time. Additionally, Figure 2 shows that most selected data subsets outperform traditional fine-tuning which uses all the data. This observation validates our intuition that some data samples are not only redundant but also detrimental to the NMT model's performance.

6.5 Comparing Required Update Steps

Our proposed curriculum training approaches not only exhibit better performance but also converge faster compared to the baseline and traditional fine-tuning method. In Figure 3, we plot the number of update steps required by each of the settings in Table 2 and 3. On average, we need about 50% fewer updates compared to the converged model. For high-resource languages, we need much fewer updates in the fine-tuning steps. For all the language pairs, the hybrid curriculum strategy requires the fewest updates as the size of selected subsets is much lower compared to other approaches.

7 Conclusion

We have presented a two-stage curriculum training framework for NMT where we apply a data selection curriculum in the fine-tuning stage. Our novel online curriculum strategy utilizes the emerging models' prediction scores for the selection of a better data subset. Experiments on 6 low- and high-resource language pairs show the efficacy of our proposed framework. Our curriculum training approaches exhibit better performance as well as converge much faster by requiring fewer updates.

References

Mikel Artetxe and Holger Schwenk. 2019. Margin-based parallel corpus mining with multilingual sentence embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3197–3203, Florence, Italy. Association for Computational Linguistics.

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

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, page 41–48, New York, NY, USA. Association for Computing Machinery.

Vishrav Chaudhary, Yuqing Tang, Francisco Guzmán, Holger Schwenk, and Philipp Koehn. 2019. Low-resource corpus filtering using multilingual sentence embeddings. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 261–266, Florence, Italy. Association for Computational Linguistics.

Volkan Cirik, Eduard H. Hovy, and Louis-Philippe Morency. 2016. Visualizing and understanding curriculum learning for long short-term memory networks. *CoRR*, abs/1611.06204.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2017. Word translation without parallel data. *arXiv preprint arXiv:1710.04087*.

Jeffrey L. Elman. 1993. Learning and development in neural networks: the importance of starting small. *Cognition*, 48(1):71–99.

Guy Hacohen and Daphna Weinshall. 2019. On the power of curriculum learning in training deep networks. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2535–2544. PMLR.

Lu Jiang, Deyu Meng, Qian Zhao, Shiguang Shan, and Alexander G. Hauptmann. 2015. Self-paced curriculum learning. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, page 2694–2700. AAAI Press.

Wenxiang Jiao, Xing Wang, Shilin He, Irwin King, Michael Lyu, and Zhaopeng Tu. 2020. Data Rejuvenation: Exploiting Inactive Training Examples for Neural Machine Translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2255–2266, Online. Association for Computational Linguistics.

Shafiq Joty, Hassan Sajjad, Nadir Durrani, Kamla Al-Mannai, Ahmed Abdelali, and Stephan Vogel. 2015. How to avoid unwanted pregnancies: Domain adaptation using neural network models. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1259–1270, Lisbon, Portugal. Association for Computational Linguistics.

Marcin Junczys-Dowmunt. 2018. Dual conditional cross-entropy filtering of noisy parallel corpora. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 888–895, Belgium, Brussels. Association for Computational Linguistics.

Huda Khayrallah and Philipp Koehn. 2018. On the impact of various types of noise on neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 74–83, Melbourne, Australia. Association for Computational Linguistics.

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, Varna, Bulgaria. INCOMA Ltd.

Gaurav Kumar, George Foster, Colin Cherry, and Maxim Krikun. 2019. Reinforcement learning based curriculum optimization for neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2054–2061, Minneapolis, Minnesota. Association for Computational Linguistics.

M. Kumar, Benjamin Packer, and Daphne Koller. 2010. Self-paced learning for latent variable models. In *Advances in Neural Information Processing Systems*, volume 23. Curran Associates, Inc.

Cao Liu, Shizhu He, Kang Liu, and Jun Zhao. 2018. Curriculum learning for natural answer generation. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 4223–4229. International Joint Conferences on Artificial Intelligence Organization.

Xuebo Liu, Houtim Lai, Derek F. Wong, and Lidia S. Chao. 2020. Norm-based curriculum learning for neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 427–436, Online. Association for Computational Linguistics.

Minh-Thang Luong and Christopher D. Manning. 2015. Stanford neural machine translation systems for spoken language domains.

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

Elissa L. Newport. 1990. Maturational constraints on language learning. *Cognitive Science*, 14(1):11–28.

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

Emmanouil Antonios Platanios, Otilia Stretcu, Graham Neubig, Barnabas Poczos, and Tom Mitchell. 2019. Competence-based curriculum learning for neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1162–1172, Minneapolis, Minnesota. Association for Computational Linguistics.

Martin Popel, Marketa Tomkova, Jakub Tomek, Łukasz Kaiser, Jakob Uszkoreit, Ondřej Bojar, and Zdeněk Žabokrtskỳ. 2020. Transforming machine translation: a deep learning system reaches news translation quality comparable to human professionals. *Nature Communications*, 11(1):1–15.

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

Marlies van der Wees, 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. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pages 5998–6008. Curran Associates, Inc.

Wei Wang, Taro Watanabe, Macduff Hughes, Tetsuji Nakagawa, and Ciprian Chelba. 2018. 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, Brussels, Belgium. Association for Computational Linguistics.

Xinyi Wang, Ankur Bapna, Melvin Johnson, and Orhan Firat. 2021. Gradient-guided loss masking for neural machine translation. *CoRR*, abs/2102.13549.

Chen Xu, Bojie Hu, Yufan Jiang, Kai Feng, Zeyang Wang, Shen Huang, Qi Ju, Tong Xiao, and Jingbo Zhu. 2020. Dynamic curriculum learning for low-resource neural machine translation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3977–3989, Barcelona, Spain (Online). International Committee on Computational Linguistics.

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. *arXiv preprint arXiv:1811.00739*.

Mingjun Zhao, Haijiang Wu, Di Niu, and Xiaoli Wang. 2020. Reinforced curriculum learning on pre-trained neural machine translation models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):9652–9659.

Tianyi Zhou, Shengjie Wang, and Jeffrey Bilmes. 2020a. Curriculum learning by dynamic instance hardness. In *Advances in Neural Information Processing Systems*, volume 33, pages 8602–8613. Curran Associates, Inc.

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

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer learning for low-resource neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1568–1575, Austin, Texas. Association for Computational Linguistics.

Appendix

A In-domain Corpora List

For high-resource language pairs, we consider formal bitext corpora sources as in-domain ($\mathcal{D}_d \subset \mathcal{D}_g$), while for low-resource pairs, we do not differentiate between general-domain and in-domain corpus ($\mathcal{D}_d \coloneqq \mathcal{D}_g$). Table 7 presents the in-domain corpora list for high-resource language pairs.

Pair	In-domain Corpora
En-De	Europarl, News Commentary
En-Hu	EUconst, Europarl, GlobalVoices, Wikipedia, WikiMatrix, WMT-News
En-Et	EUconst, Europarl, WikiMatrix, WMT-News

Table 7: In-domain corpora for high-resource language pairs.

B Model Architecture Settings

For En↔Ha, we use a smaller Transformer architecture with five layers, while for the other language pairs we use larger Transformer architecture with six encoder and decoder layers. We present the number of attention heads, embedding dimension, and inner-layer dimension of both settings in Table 8.

Settings	En↔Ha	Other Pairs
Transformer Layers	5	6
#Attention Heads	8	16
Embedding Dimension	512	1024
Inner-layer Dimension	2048	4096

Table 8: Model architecture settings.

C Traditional Fine-tuning Vs. Converged Model Performance

Comparing the performance of traditional fine-tuning (*All Data* in Table 2) with the *Converged Model* for low-resource languages, we see that both of these perform on par. This is not surprising as both approaches use all the train data ($\mathcal{D}_g$) during the whole training (for low-resource languages $\mathcal{D}_d \coloneqq \mathcal{D}_g$). The only difference between the two approaches is —- while the converged model continues to train the base model from warm-up stage, the traditional fine-tuning approach resets the base model's meta-parameters (e.g., learning-rate, lr-scheduler, dataloader, optimizer) and continue the training.

For high-resource languages in Table 3, while we fine-tune the base model only on the in-domain training data ($\mathcal{D}_d \subset \mathcal{D}_g$) in traditional fine-tuning (*All In-domain Data* in the Table), the converged model continues to train the base model on all the general-domain data ($\mathcal{D}_g$). Here, traditional fine-tuning performs better than the converged model on En-De (+0.4) and En-Et (+0.9), while exhibits worse performance on the other four directions by 0.7 BLEU score on an average.

D Overlap of Selected Data Subset

We compare the data percentage overlap of the ordered data between any two methods of §3.1 in Figure 4. From the plots, we see that the overlaps between the data subsets are quite low. Let us consider En-De for an example: if we take the top 40% data ranked by both `LASER` and dual `DCCE` methods, the overlap between these two subsets is 47%. But both of the subsets perform pretty well compared to the converged model and traditional fine-tuned model (Table 3). We observe the similar phenomena in almost all the cases (Figure 2, 4). These observations suggest that there can be multiple subsets of data for each language pair, fine-tuning the base model on which exhibits better performance compared to the traditional fine-tuning that uses all the data.

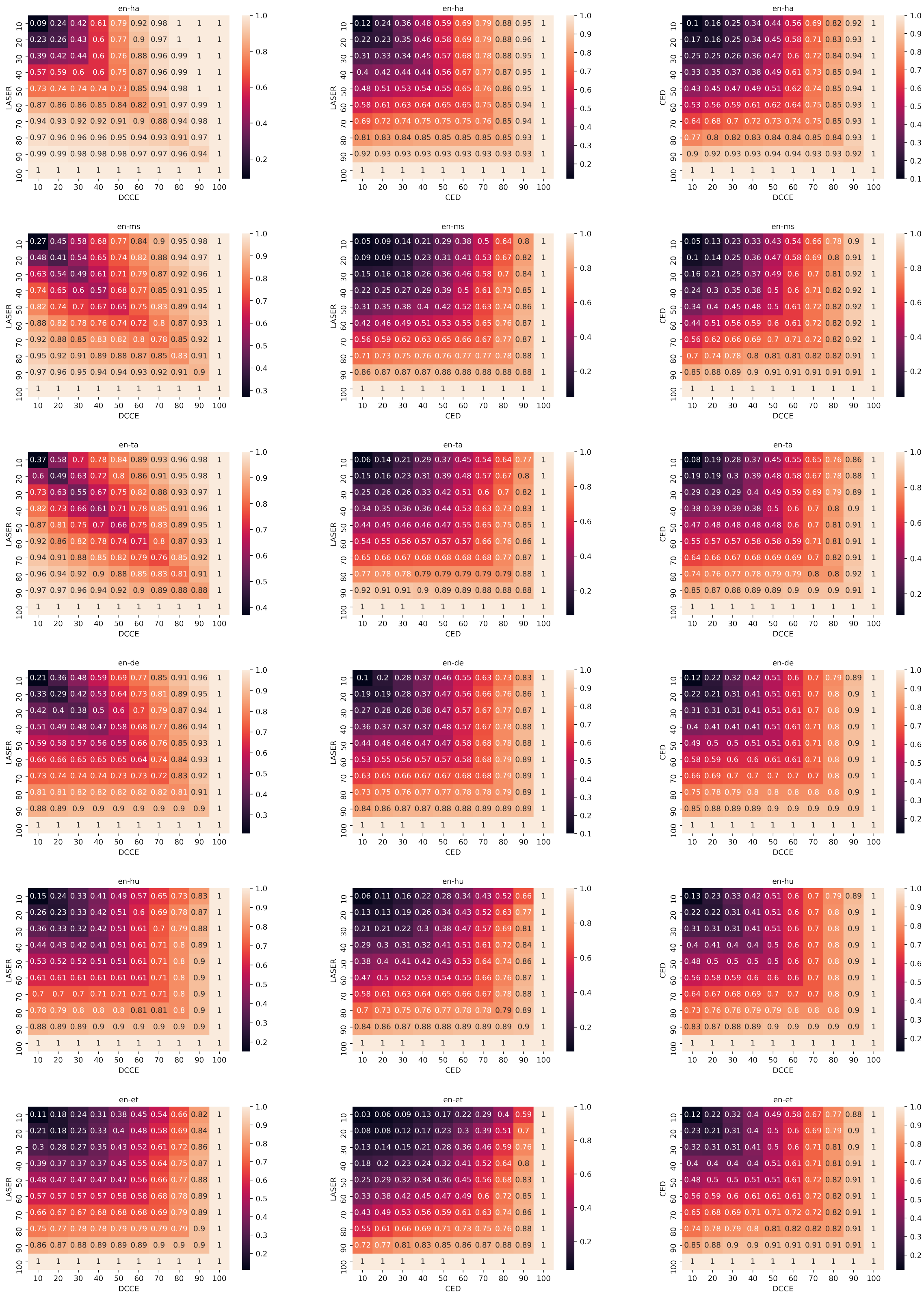

Figure 4: Overlap percentage of *ranked data* between any two methods {`LASER`, `DCCE`, `CED`}.