Token-wise Curriculum Learning for Neural Machine Translation

Chen Liang[*][◇], **Haoming Jiang**[◇], **Xiaodong Liu**[†], **Pengcheng He**[‡],
Weizhu Chen[‡], **Jianfeng Gao**[†] **and Tuo Zhao**[◇]
[◇] Georgia Tech, [†] Microsoft Research, [‡] Microsoft Azure AI

Abstract

Existing curriculum learning approaches to Neural Machine Translation (NMT) require sampling sufficient amounts of "easy" samples from training data at the early training stage. This is not always achievable for low-resource languages where the amount of training data is limited. To address such a limitation, we propose a novel token-wise curriculum learning approach that creates sufficient amounts of easy samples. Specifically, the model learns to predict a short sub-sequence from the beginning part of each target sentence at the early stage of training. Then the subsequence is gradually expanded as the training progresses. Such a new curriculum design is inspired by the cumulative effect of translation errors, which makes the latter tokens more challenging to predict than the beginning ones. Extensive experiments show that our approach can consistently outperform baselines on five language pairs, especially for low-resource languages. Combining our approach with sentence-level methods further improves the performance of high-resource languages.

1 Introduction

Neural Machine Translation (NMT) has achieved significant progress in recent years (Sutskever et al., 2014; Bahdanau et al., 2014; Vaswani et al., 2017), mainly in the scenarios where the parallel training corpora are abundant. However, training corpora can be limited in some domains (e.g., spoken language (Cettolo et al., 2015)) and languages (e.g., African languages) due to the high cost of data acquisition. Koehn and Knowles (2017); Lample et al. (2018) show that NMT models do not perform well in such data-limited settings.

To improve NMT with limited data, researchers resort to large amounts of auxiliary data. One line of research leverages the knowledge from high-resource parallel corpora. For examples, some works pre-train NMT models on high-resource data, and then fine-tune them on low-resource data (Zoph et al., 2016; Chen et al., 2017; Kocmi and Bojar, 2018; Neubig and Hu, 2018; Nguyen and Chiang, 2017); others train Multilingual or Multitask NMT models jointly on both high-resource and low-resource datasets (Gu et al., 2018a,b; Aharoni et al., 2019; Jiang et al., 2019; Siddhant et al., 2020). The other line exploits high-resource monolingual data as auxiliary data to train NMT models in a semi-supervised manner (Sennrich et al., 2015; Currey et al., 2017; Cheng, 2019).

Aside from previous approaches, curriculum learning (Bengio et al., 2009) is proposed to address the data insufficiency issue via utilizing the limited data more efficiently (Zhang et al., 2019b). The idea of curriculum learning is to sample training data in an order of increasing difficulty. The "easy" samples of such a curriculum can be beneficial to the training of the models at the early stage. There have been multiple designs of curriculum for NMT in the recent literatures (Zhou et al., 2020; Liu et al., 2020; Ruiter et al., 2020; Platanios et al., 2019; Wang et al., 2019a,b; Kumar et al., 2019; Zhang et al., 2018). All these methods sample complete sentence pairs for training from a selected *subset*, which expands as training progresses. We refer to them as the "sentence-level" curricula.

However, such a sentence-level design is not necessarily effective for NMT when data is limited. In the early stage of training, the selected *subset* is usually limited to a small portion of total training samples. In the low-resource setting, this *subset* contains even fewer samples. To better measure this effect, we use Figure 1 to show the diversity of the samples selected in the early training stage under low-resource and high-resource settings [1].

---

[*]Work was done at Microsoft Azure AI.

[1]For all sentence-level curriculum experiments in Section 1, we adopt the design proposed in Zhou et al. (2020).

Specifically, we count the number of unique trigrams in the sentence pairs used for training up-till a certain training iteration. We observe that the selected samples in low resource setting are less diverse than in high-resource setting, especially in the early curriculum (i.e. up-till 25% of total updates in curriculum). Consequently, the sentence-level curriculum slows down the learning progress in low-resource setting although this is not an issue for high-resource setting, as shown in Figure 2.
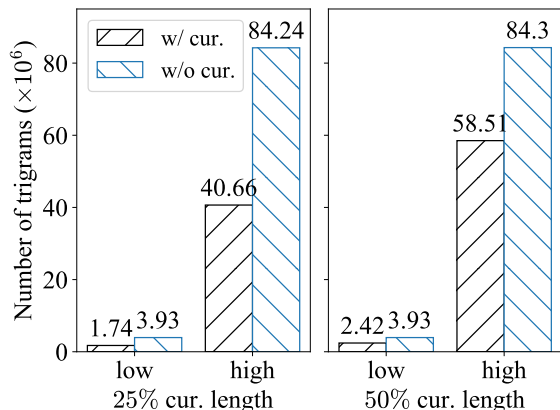

Figure 1: Sample diversity in early stage of training under low-resource and high-resource settings.
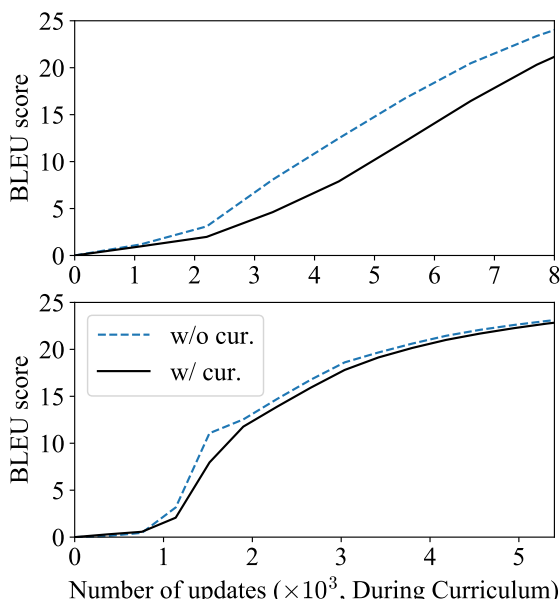

Figure 2: BLEU score (dev) in low-resource (upper) and high-resource (lower) settings.

This observation, that insufficient diversity in low-resource setting can affect the learning efficiency, motivates us to design a token-wise curriculum. During the curriculum, the model learns to

---

The dataset we use in low-resource setting is IWSLT14 De-En, and in high-resource setting is WMT16 En-De. We adopt Transformer-base (Vaswani et al., 2017) as the baseline model.

predict only a short sub-sequence from *each* target sentence at the early stage of training, and then the sub-sequence is gradually expanded as the training progresses. Comparing with the sentence-level curriculum, which only focuses on "easy" sentence pairs, token-wise curriculum can on purpose create much more partial and diverse samples to address the data insufficiency challenge.

The next question is, how to design an effective sub-sequence selection scheme, such that the difficulty of the selected sub-sequences follows an "easy-to-hard" schedule. Specifically, we consider the sub-sequence difficulty in the context of machine translation generation. In a left-to-right autoregressive generation setting, the generation of next word is dependent on previous generation on the left. In other words, wrong predictions in the early tokens would affect the accuracy of the latter ones during inference. This results in prediction error accumulation[2] (Zhang et al., 2019a), which indicates that the beginning tokens are easier to predict than the latter ones. Therefore, we design a scheduler to select sub-sequences from the *beginning* part of target sentences, and gradually expand them until the end of the sentences, as the training progresses.

Our experiments on several low-resource NMT datasets collected from IWSLT (Cettolo et al., 2015) show that the proposed curriculum outperforms existing baselines in both standard training and transfer learning settings. In addition, the experiments on a high-resource dataset WMT'16 En-De (Bojar et al., 2016) shows that the proposed curriculum can not only by itself, but also by combining with existing sentence-level curricula, benefit NMT model training. Finally, we show that the proposed token-wise curriculum is general for multiple sequence generation tasks. It shows superior performance on language modeling tasks, besides machine translation. Our codes are released at https://github.com/cliang1453/token-wise-curriculum-learning.

2 Background

• **NMT** models the conditional probability of a target sentence $y = (y_1, ..., y_\ell)$ given a source sentence $x = (x_1, ..., x_m)$. The density function $p(y|x)$ is parameterized by an encoder-decoder

---

[2]We verify that the phenomenon of error accumulation exists in NMT in Section 4.8 Figure 4.

neural network, which generates the target sentence in an auto-regressive manner (Sutskever et al., 2014; Bahdanau et al., 2014). Specifically, the model predicts the probability of the t-th token by $p(y_t|y_{<t}, \boldsymbol{x}; \theta)$, where θ denotes the model parameters. It is trained by minimizing the sum of cross-entropy loss on all sentence pairs, where the loss on each sentence pair $(\boldsymbol{x}, \boldsymbol{y})$ is

$$L(\boldsymbol{x}, \boldsymbol{y}; \theta) = -\frac{1}{\ell} \sum_{t=1}^{\ell} \log p(y_t|y_{<t}, \boldsymbol{x}; \theta) \quad (1)$$

• **Curriculum Learning in NMT**. Research on curriculum learning in NMT mainly fall into two categories: measurement of sample difficulty and design of curriculum schedule (Kocmi and Bojar, 2017). In the first category, some research measure sample difficulty with features derived from lexical statistics, e.g., sentence length and word rarity (Zhang et al., 2018; Platanios et al., 2019). Others measure difficulty with features derived from pre-trained model, e.g., Liu et al. (2020) use the norm of pre-trained word embeddings, Wang et al. (2018) leverage a pre-trained NMT model to measure sample noise-level, and Zhou et al. (2020) use a pre-trained language model to measure the word-level perplexity (i.e. uncertainty). In the second category, most schedules select samples with difficulty under a threshold. Platanios et al. (2019) determine the threshold by a linear/square-root function of training step, Liu et al. (2020) design a function based on the norm of the encoder word embedding, and Zhou et al. (2020) design a function based on model uncertainty.

3 Method

We introduce a token-wise curriculum learning approach for NMT.

3.1 Hard Curriculum

We propose a token-wise curriculum based on sub-sequence selection. At each training iteration, the model is trained to predict a *sub-sequence* of target sentences. We remark that such prediction is conditioned on *complete* source sentences. Specifically, at the i-th iteration, the model is updated based on the loss computed on such sub-sequence only,

$$-\frac{1}{|S_i|} \sum_{t \in S_i} \log p(y_t|y_{<t}, \boldsymbol{x}; \theta), \quad (2)$$

where S_i is the set of the token indexes in the selected sub-sequence at the i-th iteration.

Left-to-Right Selection Scheme. The selection scheme of S_i can be described as follows:
• At the beginning of curriculum (0-th iteration), we select the sub-sequence from the beginning of each target sentence: $S_0 = [1, 2, 3, ..., \lfloor \lambda_0 \ell \rfloor]$, where ℓ is the length of the target sequence and $\lambda_0 \in (0, 1)$ is the initial sub-sequence percentage with respect to the total length.
• We then gradually expand each sub-sequence throughout the curriculum until it covers the whole sentence: $S_i = [1, 2, 3, ..., \ell_i]$ with the length ℓ_i determined by a linear function,

$$\ell_i = \lfloor \ell \cdot (\lambda_0 + \frac{i}{I} \cdot (1 - \lambda_0)) \rfloor \quad 0 < i < I, \quad (3)$$

where I is the number of updates in the curriculum.

With this selection scheme, the model can be updated by the SGD type algorithm (e.g., ADAM (Kingma and Ba, 2014)) with the stochastic gradients computed based on S_i. The loss of each sentence at the i-th iteration is computed by:

$$-\frac{1}{\ell_i} \sum_{t=1}^{\ell_i} \log p(y_t|y_{<t}, \boldsymbol{x}; \theta). \quad (4)$$

After the curriculum ends (i.e., $i \geq I$), the model continues with the standard training.

3.2 Soft Curriculum

In hard curriculum, the model is trained without regarding the loss upon $\{1, ..., \ell\} \setminus S_i$. However, those tokens may play important roles in sequence generation. For example, the model needs to learn how to end a sentence by predicting the $\langle EOS \rangle$ token. Therefore, we propose an alternative method – the soft curriculum, where we place geometrically decaying weights on the loss of all tokens. By allowing weights on all tokens, the model is able to learn more diverse samples. By placing decaying weight on end tokens that are difficult to learn, we maintain the sample easiness.

At the i-th iteration, the re-weighted loss on each target sentence with length ℓ is computed by:

$$-\frac{1}{\ell} \sum_{t=1}^{\ell} \gamma_i^{\alpha_{t,\ell}} \log p(y_t|y_{<t}, \boldsymbol{x}, \theta), \quad (5)$$

where γ_i and $\alpha_{t,\ell}$ are two factors controlling the rate of geometric decay. The decaying factor γ_i at the i-th iteration is computed by:

$$\gamma_i = \gamma_0 + \frac{i}{I} \cdot (1 - \gamma_0) \quad 0 < i < I, \quad (6)$$

where $0 \leq \gamma_0 < 1$ is a hyperparameter controlling the scale of initial weights placed on all tokens. The weights gradually increase as γ_i grows from γ_0 to 1 throughout the curriculum. We remark that while γ_i grows linearly, the weights change with different rates for tokens at different positions – we design the power factor $\alpha_{t,\ell}$ uniquely for the t-th token in a target sentence of length ℓ:

$$\alpha_{t,\ell} = \alpha_0 \cdot \frac{t-1}{\ell-1} \quad 0 < t \leq \ell. \qquad (7)$$

As illustrated in Figure 3, the weights on tokens gradually decay from the beginning to the end of the sentence, where $\alpha_0 > 0$ is a hyperparameter controlling this decaying rate.
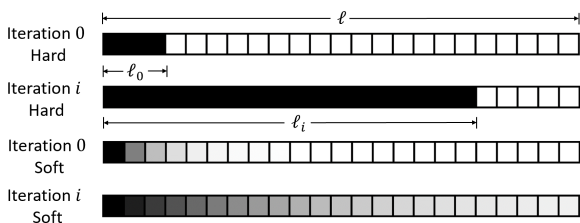

Figure 3: A comparison between sub-sequence selection in hard and soft curricula. Each row of grids represents a target sentence of tokens. The depth of color from light→dark represents the weight on the token loss from 0→1.

4 NMT Experiments

To demonstrate the effectiveness of our token-wise curriculum design, we present experimental results on NMT tasks.

4.1 Data Preparation & Preprocessing

We evaluate our method on widely used language pairs in both low-resource and high-resource datasets. Low-resource datasets include English-to-Vietnamese (En-Vi) from IWSLT15 (Cettolo et al., 2015)[3], German-to-English (De-En) from IWSLT14, French-to-English (Fr-En) from IWSLT16, and Romanian-to-English (Ro-En) from WMT16 (Bojar et al., 2016)[4]. The high-resource dataset is the WMT16 En-De. Table 1 shows the number of sentence pairs in each dataset. See A.1 for details on dev and test set used.

All datasets are encoded using byte-pair encoding (BPE, Sennrich et al. (2016)). For En-Vi and

Data	Train	Dev	Test
En-Vi	133K	768	1268
De-En	160K	7283	6750
Fr-En	224K	1080	1133
Ro-En	612K	1999	1999
En-De	4.5M	1061	1019

Table 1: The number of parallel sentences in datasets.

Fr-En, we use a BPE trained with 32K merge operations and use sentences up to length 200 subword symbols, following Platanios et al. (2019). For Ro-En, we use a BPE trained with 40K merge operations and use sentences up to length 50 subword symbols as Gu et al. (2018a,b). We preprocess De-En data following *fairseq*[5]. We adopt the pre-processed En-De data released by Google[6].

4.2 Baselines

We compare our token-wise curriculum learning method (**TC**) with several state-of-the-art sentence-level methods (**SC**):

• **SC$_{\text{r-sqrt}}$** measures sample difficulty by word rarity, and uses a square-root function as curriculum schedule (Platanios et al., 2019).

• **SC$_{\text{norm}}$** measures sample difficulty based on norm of sentence embedding, and uses a threshold function of encoder word embedding norm as curriculum schedule (Liu et al., 2020).

• **SC$_{\text{unc}}$** measures sample difficulty by data uncertainty, and uses a threshold function of model uncertainty as curriculum schedule (Zhou et al., 2020).

4.3 Model & Training

For both **SC** and **TC** experiments, we adopt Transformer-base NMT model (Vaswani et al., 2017) as the baseline model. All implementations are based on *fairseq* (Ott et al., 2019) code-base with all experiments running with 32G NVIDIA V100 GPUs. For all datasets, we use ADAM (Kingma and Ba, 2014) as the optimizer with $\beta = (0.9, 0.98)$. For low-resource datasets, we use a learning rate of 5×10^{-4} with 8000 steps of warmup updates. For high-resource dataset En-De, we use a learning rate of 1×10^{-3} with 4000 steps of warmup updates. See training details in A.2.

We fix $\lambda_0 = 0.1$ in $\mathbf{TC}_{\text{hard}}$ experiments, and fix $\gamma_0 = 0.7$ and $\alpha_0 = 25$ in $\mathbf{TC}_{\text{soft}}$ experiments. We set the curriculum length $I = 8000, 7000, 6500, 1100, 5400$ for De-En, En-Vi, Fr-En, Ro-En and En-De. See hyperparameter selection details in A.3. For $\mathbf{SC}$ methods, we follow the recommended settings in the original papers with special configurations for the low-resource setting. See training details in A.4.

Consistent with previous practice, we use tokenized BLEU (Papineni et al., 2002) as the evaluation metrics. For all low-resource datasets, we report the BLEU score of the best checkpoint using a beam size of 5 and length penalty of 1. For high-resource dataset En-De, we report the average of the last 10 checkpoints with a beam size of 10 and length penalty of 0.6.

4.4 Main Results

We compare $\mathbf{TC}_{\text{hard}}$ and $\mathbf{TC}_{\text{soft}}$ with the baseline, and report the best testing BLEU among 5 runs with different random seeds in Table 2 and Table 3 (See A.5 for validation scores). As can be seen, $\mathbf{TC}_{\text{hard}}$ outperforms the baseline in all cases, and $\mathbf{TC}_{\text{soft}}$ further improves upon $\mathbf{TC}_{\text{hard}}$. This implies that $\mathbf{TC}_{\text{soft}}$ finds a better balance between sample diversity and sample easiness than $\mathbf{TC}_{\text{hard}}$.

In the low resource setting (Table 2), all $\mathbf{TC}$ methods uniformly outperform $\mathbf{SC}$ methods by around 0.5 BLEU scores, while $\mathbf{SC}$ methods can sometimes hurt the baseline (e.g., in En-Vi and De-En, the two smallest datasets). Under the high resource setting (Table 3), all $\mathbf{TC}$ methods outperform the baseline by around 0.4 BLEU scores. However, we observe that the performance of $\mathbf{TC}$ methods show no clear improvement upon $\mathbf{SC}$ methods. We conjecture the reason is that the selected samples in high resource setting are sufficiently diverse for $\mathbf{SC}$ method to be well-performed. To further improve performance in high resource setting, we combine $\mathbf{TC}$ and $\mathbf{SC}$ methods, expecting that this combination selects not only diverse, but also easier samples than those selected by any single method. In particular, we first use $\mathbf{SC}$ to select sentences, and then use $\mathbf{TC}$ to select beginning sub-sequences upon these sentences. As can be seen, both $\mathbf{TC}_{\text{soft}} + \mathbf{SC}_{\text{norm}}$ and $\mathbf{TC}_{\text{soft}} + \mathbf{SC}_{\text{unc}}$ can further improve upon the best single method.

As $\mathbf{TC}_{\text{soft}}$ uniformly outperforms $\mathbf{TC}_{\text{hard}}$, we use $\mathbf{TC}_{\text{soft}}$ in the following experiments unless stated otherwise.

	En-Vi	De-En	Fr-En	Ro-En
w/o Cur.	31.43	34.33	37.21	32.10
$\mathbf{SC}_{\text{r-sqrt}}$	31.01	34.29	37.25	32.19
$\mathbf{SC}_{\text{norm}}$	31.05	34.24	37.28	32.25
$\mathbf{SC}_{\text{unc}}$	31.33	34.48	37.60	32.26
$\mathbf{TC}_{\text{hard}}$	31.85	34.88	38.22	32.45
$\mathbf{TC}_{\text{soft}}$	**31.94**	**34.91**	**38.28**	**32.52**

Table 2: BLEU scores (test) on low-resource datasets. The performance gains of $\mathbf{TC}_{\text{hard}}$ and $\mathbf{TC}_{\text{soft}}$ are significant compared with $\mathbf{SC}_{\text{unc}}$ – the results of 5 runs pass the unpaired student's t-test with p-value < 0.01.

	En-De
w/o Curriculum (Vaswani et al., 2017)	28.10
$\mathbf{SC}_{\text{r-sqrt}}$ (Platanios et al., 2019)	28.27
$\mathbf{SC}_{\text{norm}}$ (Liu et al., 2020)	28.51
$\mathbf{SC}_{\text{unc}}$ (Zhou et al., 2020)	28.55
$\mathbf{TC}_{\text{hard}}$	28.49
$\mathbf{TC}_{\text{soft}}$	28.54
$\mathbf{TC}_{\text{soft}} + \mathbf{SC}_{\text{norm}}$	28.62
$\mathbf{TC}_{\text{soft}} + \mathbf{SC}_{\text{unc}}$	**28.67**

Table 3: BLEU scores (test) on the high-resource datasets. The performance gains of $\mathbf{TC}_{\text{soft}}$, $\mathbf{TC}_{\text{hard}}$, $\mathbf{TC}_{\text{soft}} + \mathbf{SC}_{\text{norm}}$ and $\mathbf{TC}_{\text{soft}} + \mathbf{SC}_{\text{unc}}$ are significant compared with w/o Curriculum – the results of 5 runs pass the unpaired student's t-test with p-value < 0.01.

4.5 Transfer Learning with Curriculum

We show that our curriculum can be further combined with transfer learning to improve NMT performance. Instead of training from scratch, transfer learning considers fine-tuning a pre-trained model on the limited parallel data. Specifically, we consider the following transfer learning settings:

• **Domain Transfer Learning**. We consider transferring from a high-resource domain to a low-resource domain. Specifically, we fine-tune the Transformer-big NMT model pre-trained from News domain (WMT) [7] on TED domain (IWSLT). Table 4 shows that using our curriculum improves the domain transfer performance.

• **Pre-trained Multilingual Language Model Fine-tuning**. We also consider the case of transferring from high-resource monolingual data to low-resource parallel data. Specifically, we initialize

---

[7] The pre-trained model is trained on WMT16 En-De data and publicly available from github.com/pytorch/fairseq/tree/master/examples/translation.

an NMT model from XLM (Lample and Conneau, 2019), a multilingual language model pre-trained on extensive monolingual En and De data [8]. Then we fine-tune the NMT model on the En-De TED data. Table 4 shows that using our curriculum improves the fine-tuning performance on pre-trained multilingual language model.

4.6 Curriculum under Extremely Low-Resource Setting

We further show that our curriculum can improve NMT performance in both standard training (Table 5) and transfer learning (Table 4) under extremely low-resource setting. In standard training, the model is trained with a randomly sampled 50%/10% subset from all sentence pairs. In transfer learning, the model is finetuned with a randomly sampled 50%/10%/1% subset from all target domain sentence pairs. Table 4 and Table 5 show that $\mathbf{TC}_{soft}$ attains a steady performance gain as training/fine-tuning data becomes more scarce, e.g., the domain transfer learning improvement is over 3 BLEU scores under the 1% data setting.

4.7 Analysis

We first verify our assumption that the error accumulation makes beginning tokens easier to predict. Then we analyze whether our curriculum improves the sample diversity in the early stage of training, and further improves optimization.

• **Error Accumulation**. To verify that error accumulation is a prevailing phenomenon in machine translation generation, we conduct beam search with beam size of 5 using Transformer-base NMT model on De-En dataset. We compute the error rate of the predictions at different relative positions of sentences. Specifically, we compute the prediction error rate within 10 evenly-divided partitions in each sentence and average over all sentences. Since we choose λ_0 invariant to sentence length, we further verify that the error accumulation exists for sentences with different length. As shown in Figure 4, sentences with different length suffer from error accumulation. We further verify that the token-wise curriculum can effectively alleviate the error accumulation in A.6.

• **Sample Diversity**. We compare the diversity of samples selected/created by $\mathbf{SC}_{unc}$ and $\mathbf{TC}_{hard}$
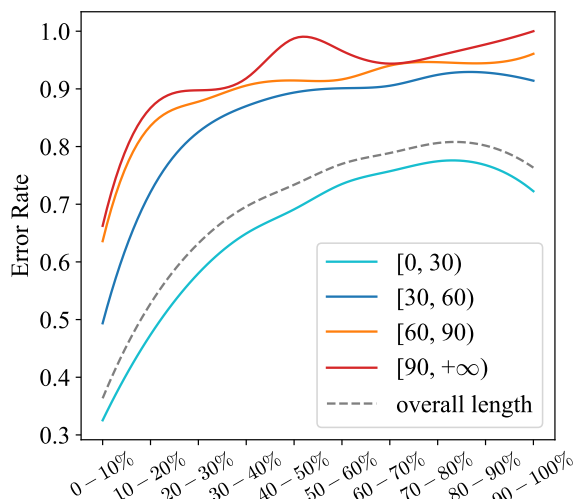

Figure 4: The averaged beam search error rate at different relative positions of sentences.

[9] on low-resource dataset De-En. Recall that the samples selected by sentence-level curriculum is a subset of all sentence pairs. In contrast, the samples created by token-wise curriculum consist of all source sentences as well as the selected subsequences from all target sentences. Up till a fixed training iteration (e.g., 25% of curriculum length), we measure the diversity by the number of unique trigrams summing over all selected/created sentences/sub-sequences. As shown in Figure 5, the samples created by $\mathbf{TC}_{hard}$ are more diverse at the early stage of training.
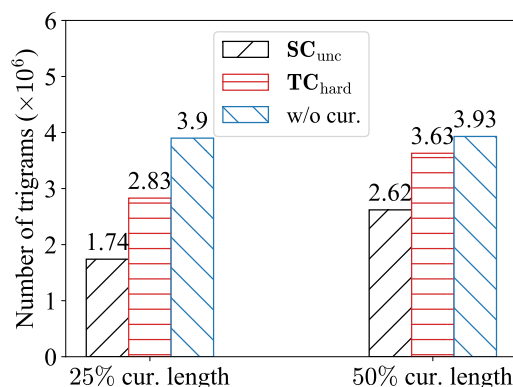

Figure 5: Sample diversity in early stage of training.

• **Learning Curve**. Figure 6 shows the validation performance of $\mathbf{SC}_{unc}$ and $\mathbf{TC}_{soft}$ in both early and later stages of training. As can be seen, the BLEU score under the token-wise curriculum increases

---

3663

Pre-trained Model	Translation Model				Multilingual Language Model (XLM)			
Source	**En-De News** (Parallel Data)				**En-De News** (Monoligual Data)			
Target	**En-De TED**				**En-De TED**			
Size	full	50%	10%	1%	full	50%	10%	1%
Transfer w/o Curriculum	32.88	32.42	31.30	26.66	31.86	28.22	14.52	9.15
TC_{soft}	**33.17**	**32.79**	**31.77**	**29.69**	**33.22**	**29.94**	**17.17**	**10.75**

Table 4: BLEU scores (test) in transfer learning, finetuned with full/50%/10%/1% subsets of target domain data.

	De-En			**Fr-En**			**Ro-En**		
Size	full	50%	10%	full	50%	10%	full	50%	10%
w/o Curriculum	34.33	31.04	16.33	37.21	34.25	21.36	32.10	29.99	21.96
TC_{soft}	**34.91**	**31.55**	**16.83**	**38.28**	**34.66**	**21.62**	**32.52**	**30.22**	**22.25**

Table 5: BLEU scores (test) on NMT, trained with full/50%/10% subsets of training sentence pairs.

faster and more smoothly than sentence-level curriculum in the early stage. Furthermore, the model trained with the token-wise curriculum achieves a better generalization performance, while the model trained with sentence-level curriculum shows signs of over-fitting. We conjecture that such improvement comes from training with more diverse samples in the early stage.

4.8 Ablation Study

We ablate some crucial designs of our curriculum, including the design of selecting consecutive tokens and the design of expanding the sub-sequence from beginning to the end of the sentence (referred as *left-to-right*). We only consider TC_{hard} in this section, as TC_{soft} is the improved version of TC_{hard}.

• **Consecutive Tokens vs. Random Tokens.** Here we study if selecting consecutive tokens is necessary for token-wise curriculum. It is natural to compare it with random sub-sequence curriculum, which uniformly samples the same number of tokens as TC_{hard} (but not necessarily consecutive). Table 6 shows that the random curriculum does not show improvement upon the baseline.

	De-En	Fr-En	Ro-En
w/o Curriculum	34.33	37.21	32.10
Random Sub-seq.	34.46	37.30	32.20
TC_{hard}	**34.88**	**38.22**	**32.45**

Table 6: BLEU score (test) comparison to random sub-sequence curriculum.

• **Teacher-forcing Loss vs. Beam Search Error Rate.** The *left-to-right* design is motivated by the error accumulation of beam search decoding which makes the latter tokens more difficult to predict (Figure 4). Recall that, unlike beam search, the NMT models are trained in a teacher-forcing way. Therefore, we would like to know whether the teacher-forcing training loss can characterize sample difficulty. To answer this question, we select the sub-sequence with the lowest average teacher-forcing loss and the same number of tokens as TC_{hard}. Table 7 shows that the selection based on teacher-forcing loss outperforms the baseline, but does not work as well as the *left-to-right* design.

	De-En	Fr-En	Ro-En
w/o Curriculum	34.33	37.21	32.10
Subseq w/ Low. Loss	34.63	37.53	32.29
TC_{hard}	**34.88**	**38.22**	**32.45**

Table 7: BLEU score (test) comparison to selecting based on sub-sequence average training loss.

• **Relative Positions of Sub-sequences**. We further explore whether choosing a sub-sequence expansion direction misaligned with the *left-to-right* decoding order can also improve the performance. We select initial sub-sequence not from the beginning of each sentence, instead, in the range of $30-40\%$, $60-70\%$ and $90-100\%$ of each sentence with the same expansion schedule. For example, by selecting the initial range as $90-100\%$, the expansion is in the *right-to-left* direction. By selecting the initial range as $30-60\%$, the sub-sequence
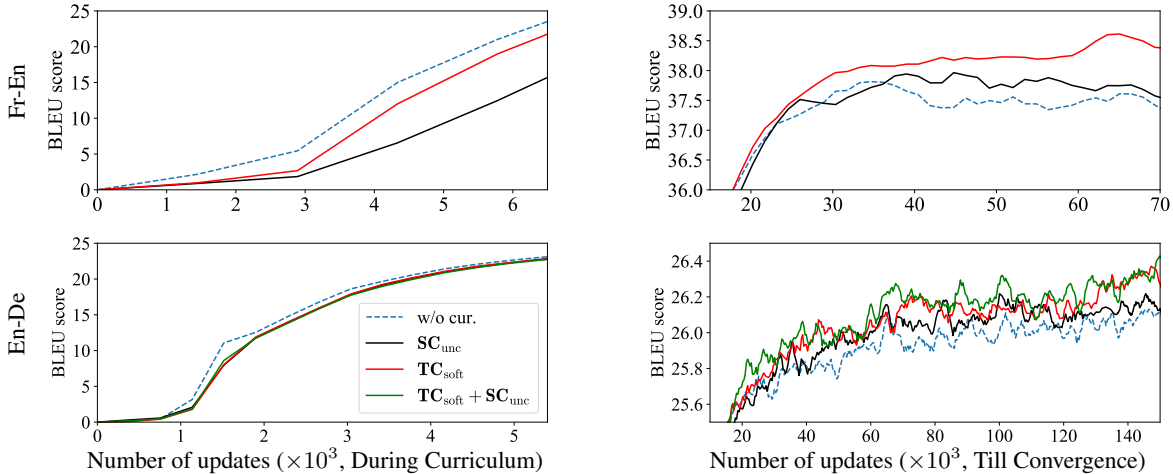
Figure 6: BLEU scores (dev) in low-resource dataset (Fr-En) and high-resource dataset (En-De).

is expanding bidirectionally. Table 8 shows that by choosing initial sub-sequence other than beginning of the sentence, the performance drops even below the baseline in some cases. This implies that the *left-to-right* design is essential as it aligns with the decoding order.

	De-En	Fr-En	Ro-En
w/o Curriculum	34.33	37.21	32.10
$TC_{hard}, 30-40\%$	34.08	37.48	32.25
$TC_{hard}, 60-70\%$	34.17	37.12	32.03
$TC_{hard}, 90-100\%$	34.18	37.45	32.43
$TC_{hard}, 0-10\%$	34.88	38.22	32.45

Table 8: BLEU score (test) comparison to selecting initial sub-sequence from different relative positions.

5 Language Modeling Experiments

To demonstrate our token-wise curriculum can be applied to other sequence generation task, we presents experimental results on language modeling.

5.1 Data Preparation & Processing

We conduct experiments on two popular word-level datasets: a preprocessed version of the Penn Treebank (PTB) (Mikolov et al., 2010) and the WikiText-2 (WT2) (Merity et al., 2016). PTB contains about 929K training words, 73K validation words, and 82K test words. All capitalization, numbers and punctuation are removed as part of the pre-processing step. WT2 consists of around 2M words extracted from Wikipedia articles. The dataset is lightly processed with capitalization, punctuation,

and numbers retained. It is tokenized and preprocessed using the Moses (Koehn et al., 2007) with over 30K vocabulary size.

5.2 Model & Training

We use AWD-LSTM (Merity et al., 2017), a 3-layer standard LSTM equipped with the drop-connection (Wan et al., 2013) on recurrent weights. The model is trained with non-monotonically triggered averaged stochastic gradient descent (NT-ASGD), a variant of ASGD (Polyak and Juditsky, 1992). We follow the training settings from Merity et al. (2017)[10] and report performance in perplexity under static evaluation.

We fix λ_0, γ_0 and α_0 the same as in Section 4.3. The curriculum length I is set to be 2100 and 4200 for PTB and WT2. See hyperparameters selection details in A.3.

5.3 Main Results

Table 9 shows the language modeling performance on PTB and WT2. As can be seen, both TC_{hard} and TC_{soft} outperform the baseline performance by over 0.5 points of perplexity. Furthermore, TC_{soft} slightly outperforms TC_{hard} in both datasets.

	PTB	WT2
w/o Curriculum	58.96	65.54
TC_{hard}	58.41	65.14
TC_{soft}	**58.23**	**65.09**

Table 9: The perplexity (test) in language modeling.

---

[10] https://github.com/salesforce/awd-lstm-lm

6 Conclusion

In this paper, we introduce a novel token-wise curriculum learning method for NMT. We show its superiority in low-resource setting, and is beneficial in high-resource setting. Different from existing works, we only consider a vanilla curriculum schedule, where the created sub-sequences expand linearly, as our focus is to validate the idea of token-wise design. We leave other potential scheduler design, e.g., training adaptive scheduler (Liu et al., 2020; Xu et al., 2020), as future discussion.

Broader Impact

This paper proposes a new curriculum learning method for training neural language models in sequence-to-sequence prediction tasks. Our designed curriculum neither introduces any social/ethical bias to the model nor amplify any bias in the data. We do not foresee any direct social consequences or ethical issues.

References

Roee Aharoni, Melvin Johnson, and Orhan Firat. 2019. Massively multilingual neural machine translation. *arXiv preprint arXiv:1903.00089*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

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

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, et al. 2016. Findings of the 2016 conference on machine translation. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 131–198.

Mauro Cettolo, Jan Niehues, Sebastian Stüker, Luisa Bentivogli, Roldano Cattoni, and Marcello Federico. 2015. The iwslt 2015 evaluation campaign. In *IWSLT 2015, International Workshop on Spoken Language Translation*.

Yun Chen, Yang Liu, Yong Cheng, and Victor OK Li. 2017. A teacher-student framework for zero-resource neural machine translation. *arXiv preprint arXiv:1705.00753*.

Yong Cheng. 2019. Semi-supervised learning for neural machine translation. In *Joint Training for Neural Machine Translation*, pages 25–40. Springer.

Anna Currey, Antonio Valerio Miceli-Barone, and Kenneth Heafield. 2017. Copied monolingual data improves low-resource neural machine translation. In *Proceedings of the Second Conference on Machine Translation*, pages 148–156.

Jiatao Gu, Hany Hassan, Jacob Devlin, and Victor OK Li. 2018a. Universal neural machine translation for extremely low resource languages. *arXiv preprint arXiv:1802.05368*.

Jiatao Gu, Yong Wang, Yun Chen, Kyunghyun Cho, and Victor OK Li. 2018b. Meta-learning for low-resource neural machine translation. *arXiv preprint arXiv:1808.08437*.

Kenneth Heafield. 2011. Kenlm: Faster and smaller language model queries. In *Proceedings of the sixth workshop on statistical machine translation*, pages 187–197.

Haoming Jiang, Chen Liang, Chong Wang, and Tuo Zhao. 2019. Multi-domain neural machine translation with word-level adaptive layer-wise domain mixing. *arXiv preprint arXiv:1911.02692*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Tom Kocmi and Ondrej Bojar. 2017. Curriculum learning and minibatch bucketing in neural machine translation. *arXiv preprint arXiv:1707.09533*.

Tom Kocmi and Ondřej Bojar. 2018. Trivial transfer learning for low-resource neural machine translation. *arXiv preprint arXiv:1809.00357*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*, pages 177–180. Association for Computational Linguistics.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. *arXiv preprint arXiv:1706.03872*.

Gaurav Kumar, George Foster, Colin Cherry, and Maxim Krikun. 2019. Reinforcement learning based curriculum optimization for neural machine translation. *arXiv preprint arXiv:1903.00041*.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *arXiv preprint arXiv:1901.07291*.

Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Phrase-based & neural unsupervised machine translation. *arXiv preprint arXiv:1804.07755*.

Xuebo Liu, Houtim Lai, Derek F Wong, and Lidia S Chao. 2020. Norm-based curriculum learning for neural machine translation. *arXiv preprint arXiv:2006.02014*.

Stephen Merity, Nitish Shirish Keskar, and Richard Socher. 2017. Regularizing and Optimizing LSTM Language Models. *arXiv preprint arXiv:1708.02182*.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*.

Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Eleventh Annual Conference of the International Speech Communication Association*.

Graham Neubig and Junjie Hu. 2018. Rapid adaptation of neural machine translation to new languages. *arXiv preprint arXiv:1808.04189*.

Toan Q Nguyen and David Chiang. 2017. Transfer learning across low-resource, related languages for neural machine translation. *arXiv preprint arXiv:1708.09803*.

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 NAACL-HLT 2019: Demonstrations*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Emmanouil Antonios Platanios, Otilia Stretcu, Graham Neubig, Barnabas Poczos, and Tom M Mitchell. 2019. Competence-based curriculum learning for neural machine translation. *arXiv preprint arXiv:1903.09848*.

Boris T Polyak and Anatoli B Juditsky. 1992. Acceleration of stochastic approximation by averaging. *SIAM journal on control and optimization*, 30(4):838–855.

Dana Ruiter, Cristina España-Bonet, and Josef van Genabith. 2020. Self-induced curriculum learning in neural machine translation. *arXiv preprint arXiv:2004.03151*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Edinburgh neural machine translation systems for wmt 16. *arXiv preprint arXiv:1606.02891*.

Aditya Siddhant, Ankur Bapna, Yuan Cao, Orhan Firat, Mia Chen, Sneha Kudugunta, Naveen Arivazhagan, and Yonghui Wu. 2020. Leveraging monolingual data with self-supervision for multilingual neural machine translation. *arXiv preprint arXiv:2005.04816*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

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

Li Wan, Matthew Zeiler, Sixin Zhang, Yann Le Cun, and Rob Fergus. 2013. Regularization of neural networks using dropconnect. In *International conference on machine learning*, pages 1058–1066.

Wei Wang, Isaac Caswell, and Ciprian Chelba. 2019a. Dynamically composing domain-data selection with clean-data selection by" co-curricular learning" for neural machine translation. *arXiv preprint arXiv:1906.01130*.

Wei Wang, Ye Tian, Jiquan Ngiam, Yinfei Yang, Isaac Caswell, and Zarana Parekh. 2019b. Learning a multitask curriculum for neural machine translation. *arXiv preprint arXiv:1908.10940*.

Wei Wang, Taro Watanabe, Macduff Hughes, Tetsuji Nakagawa, and Ciprian Chelba. 2018. Denoising neural machine translation training with trusted data and online data selection. *arXiv preprint arXiv:1809.00068*.

Benfeng Xu, Licheng Zhang, Zhendong Mao, Quan Wang, Hongtao Xie, and Yongdong Zhang. 2020. Curriculum learning for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6095–6104.

Wen Zhang, Yang Feng, Fandong Meng, Di You, and Qun Liu. 2019a. Bridging the gap between training and inference for neural machine translation. *arXiv preprint arXiv:1906.02448*.

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*.

Xuan Zhang, Pamela Shapiro, Gaurav Kumar, Paul McNamee, Marine Carpuat, and Kevin Duh.

2019b. Curriculum learning for domain adaptation in neural machine translation. *arXiv preprint arXiv:1905.05816*.

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

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer learning for low-resource neural machine translation. *arXiv preprint arXiv:1604.02201*.

A Appendix

A.1 Datasets

• **IWSLT14 De-En** We follow Ott et al. (2019) [11] to split the dev/test sets.

• **IWSLT15 En-Vi** We use the standard TED dev2010 and tst2013 as dev and test set following Platanios et al. (2019).

• **IWSLT16 Fr-En** We use the standard TED tst2015 and tst2016 as dev and test set following Platanios et al. (2019).

• **WMT16 Ro-En** We use the standard newsdev-2016 and newstest-2016 are used as dev and test set.

• **WMT16 En-De** We use the standard newstest-2013 and newstest-2014 as dev and test set.

A.2 TC Methods Implementation Details

• **NMT Standard Training Experiments.** For all language pairs, we use a inverse square root schedule with weight decay rate of 1×10^{-4}, label smoothing ratio of 0.1, and dropout rate of 0.3.

For low resource setting, we share the decoder and encoder output embeddings. We use dynamic batching with maximum tokens of 4096 per GPU and train on 1 GPU for 60 epochs.

For high resource setting, we share all the embeddings. We use dynamic batching with 14336 tokens per GPU, accumulate gradient for 7 steps, and train for 150K updates.

For extremely low-resource setting, we follow the same hyperparameter setting for each language pair.

• **NMT Transfer Learning Experiments.** We use 2 NVIDIA V100 GPUs for each experiment. We choose finetuning learning rate from $\{1 \times 10^{-5}, 5 \times 10^{-5}, 5 \times 10^{-4}\}$. We use dynamic batch size, which is limited by GPU memory (16G per GPU). We report the evaluation results by conducting beam search with beam size of 5 and length penalty of 0.6 for datasets in WMT, and beam size of 5 and length penalty of 2 for datasets in IWSLT.

A.3 TC Methods Hyperparameter Selection

• **Selection of $\lambda_0, \alpha_0, \gamma_0$.** We choose λ_0 in $\{0.1, 0.2, 0.3\}$, γ_0 in $\{0.5, 0.6, 0.7, 0.8, 0.9\}$, and α_0 in $\{12, 25, 37, 50\}$. We find that setting λ_0 in $\{0.1, 0.3\}$, γ_0 in $\{0.7, 0.9\}$ and α_0 in $\{12, 25\}$

leads to less than 0.05 variance in validation performance, suggesting **TC** methods are insensitivity to hyperparameters.

• **Selection of I.** In NMT experiments, we determine I in a similar manner as Platanios et al. (2019): we train the baseline model and compute the number of training steps it takes to reach approximately 70% of its final BLEU score. We then set I to this value. In language modeling experiments, I is determined similarly: we train the baseline model and set I to be the number of training steps it takes to reach approximately 30% initial perplexity + 70% final perplexity.

A.4 SC Methods Implementation Details

• **SC$_{\text{r-sqrt}}$.** We adopt the SR curriculum and c_{sqrt} competence function setting in Platanios et al. (2019). We set initial competence c_0 to 0.01 for all language pairs and set curriculum length T in the same manner following Platanios et al. (2019). In addition, we adopt the special learning rate schedule as proposed in Equation (9) in the original paper, where we set $T_{\text{warmup}} = 8000$.

• **SC$_{\text{norm}}$.** Following Liu et al. (2020), we extract a word2vec embedding E^{w2v} from a pre-trained Transformer-base model and measure sample difficulty on the source sentences embedding mapped through E^{w2v}. The initial competence c_0 is set to 0.01 for all language pairs. For En-De, λ_m and λ_w are set to 2.5 and 0.5 following Liu et al. (2020). For low-resource datasets, we tune and choose λ_m and λ_w as 0.25 and 0.05, respectively.

• **SC$_{\text{unc}}$.** We follow Zhou et al. (2020) to use 4 baby steps. We measure sample difficulty using the "joint" source and target uncertainty. It is obtained by evaluating the perplexity measured by a pretrained 4-gram KENLM model (Heafield, 2011)).

A.5 Validation Performance

• **NMT Standard Training Experiments.** Table 10 shows the validation performance on low-resource datasets. We report the BLEU score on the best checkpoint. and Table 11 shows the validation performance on a high-resource dataset En-De. We report the BLEU score on the averaged last 10 checkpoints. We use the same beam search setting as in Section 4.3.

• **Language Modeling Experiments.** Table 12 shows the validation performance of the language modeling experiments.

---

[11] https://github.com/pytorch/fairseq/blob/master/examples/translation/prepare-iswlt14.sh

	En-Vi	De-En	Fr-En	Ro-En
w/o Cur.	29.77	35.62	37.99	32.91
$SC_{r\text{-}sqrt}$	29.75	35.44	38.14	32.95
SC_{norm}	29.70	35.83	38.21	33.02
SC_{unc}	29.32	35.80	38.26	32.75
TC_{hard}	30.39	36.11	38.68	33.21
TC_{soft}	**30.42**	**36.14**	**38.77**	**33.24**

Table 10: The BLEU scores (dev) on low-resource datasets.

	En-De
w/o Cur. (Vaswani et al., 2017)	26.10
$SC_{r\text{-}sqrt}$ (Platanios et al., 2019)	26.15
SC_{norm} (Liu et al., 2020)	26.32
SC_{unc} (Zhou et al., 2020)	26.26
TC_{hard}	26.44
TC_{soft}	26.46
$TC_{soft} + SC_{norm}$	**26.57**
$TC_{soft} + SC_{unc}$	26.48

Table 11: BLEU scores (dev) on a high-resource dataset.

	PTB	WT2
AWD-LSTM	61.09	68.40
TC_{hard}	60.85	68.32
TC_{soft}	60.63	67.82

Table 12: The perplexity (dev) in language modeling.

	All Lengths	Lengths > 100
w/o Curriculum	76.8%	99.3%
SC_{unc}	77.0%	97.1%
TC_{hard}	75.7%	87.7%

Table 13: Averaged prediction error rate over the end 20% tokens for sentences in De-En dataset.

A.6   Additional Analysis

We further verify that the token-wise curriculum can effectively alleviate the error accumulation. We conduct beam search with beam size of 5 on Transformer-base model trained on De-En, and compute the averaged prediction error rate over the end 20% tokens of the sentences. As shown in the Table 13, the model trained with TC_{hard} suffers less from error accumulation than the model trained with SC_{unc}. In addition, TC_{hard} particularly alleviates error accumulation in long sentences (i.e.,

sentences with length larger than 100).