PAPER

Inference Discrepancy Based Curriculum Learning for Neural Machine Translation

Lei ZHOU[†a)], **Ryohei SASANO**[†], *Nonmembers*, *and* **Koichi TAKEDA**[†], *Member*

SUMMARY In practice, even a well-trained neural machine translation (NMT) model can still make biased inferences on the training set due to distribution shifts. For the human learning process, if we can not reproduce something correctly after learning it multiple times, we consider it to be more difficult. Likewise, a training example causing a large discrepancy between inference and reference implies higher learning difficulty for the MT model. Therefore, we propose to adopt the inference discrepancy of each training example as the difficulty criterion, and according to which rank training examples from easy to hard. In this way, a trained model can guide the curriculum learning process of an initial model identical to itself. We put forward an analogy to this training scheme as guiding the learning process of a curriculum NMT model by a pretrained vanilla model. In this paper, we assess the effectiveness of the proposed training scheme and take an insight into the influence of translation direction, evaluation metrics and different curriculum schedules. Experimental results on translation benchmarks WMT14 English ⇒ German, WMT17 Chinese ⇒ English and Multitarget TED Talks Task (MTTT) English ⇔ German, English ⇔ Chinese, English ⇔ Russian demonstrate that our proposed method consistently improves the translation performance against the advanced Transformer baseline.

key words: *curriculum learning, machine translation, inference discrepancy, self-paced learning*

1. Introduction

Human learning is a continuous process in which individuals use prior knowledge to build up new understandings and modify their behaviors to adapt to the environment. It does not necessarily mean improvements or developments in the right direction. Goal-directed learning activities often need a meticulous curriculum to direct the gradual acquisition of knowledge and skills. Analogous to the nature of human learning, Elman [1] found it beneficial for neural networks to undergo a gradual and phased training process from easy to hard. This is a very early approach that arranges training examples by their learning difficulty in accordance with the maturity of neural networks.

Curriculum learning method consists of two important components, a criterion to assess the learning difficulty of each training examples and a schedule to arrange training steps for subsets of training examples with different level of training difficulty. We describe them as *difficulty criterion* and *curriculum schedule*. In the context neural machine translation (NMT), recent curriculum learning approaches structure their curriculum based on various difficulty criteria. Kocmi and Bojar [2] built a difficulty criterion on the basis of linguistic features, i.e. sentence length. Platanios *et al.* [3] developed this idea by computing the cumulative distribution function (CDF) value over the distribution of sentence length and word rarity of all training data. Other approaches derive difficulty criterion from distribution learned by a language model [4]–[6] or a word embedding model [7]. While great progress has been made, commonly used difficulty criteria do not consider the learning difficulty from the "perspective" of an NMT model. Guiding the learning process with such difficulty criteria may not fit close enough to the nature of NMT model training. The idea of defining the difficulty by an NMT model is adopted in two approaches, where Zhang *et al.* [8] use an auxiliary translation model much simpler configuration than the baseline model to get the probability of the one-best translation and Xu *et al.* [9] evaluate the decline of loss of each training samples during the NMT model training. There are also drawbacks. First, both the loss function and the function for the probability of one-best translation are at the word level, while the performance of an NMT model is usually evaluated at a sentence level with discrete metrics or sentence similarity. The evaluation of model performance on one training example is not consistent with that on the test set during inference. Second, due to the exposure bias [10], the sentences with lower training loss may indicate higher inference accuracy.

In this paper, we continue the line of research intending to propose a difficulty criterion that reflects the "perspective" of an NMT model itself and guide the learning process accordingly. In human learning, if some knowledge is difficult to master, we are likely to make more mistakes when putting it into use. Likewise, a trained NMT model tends to make inferior translation on difficult examples. For a well-trained NMT model, the discrepancy between translation and reference during inference, which we refer to as *inference discrepancy*, reflects the gap between training and inference. The gap is primarily attributable to three factors. First, during training the teacher forcing approach use words from data distribution as context while during inference the auto-regressive approach use previously generated words from model distribution as context. The distribution shift between data distribution and and model distribution, addressed as exposure bias, is considered to be one causal factor [10], [11]. This is why inference discrepancy also exists when enforcing inference on the training set, which has been seen during training, see Table 1. Second, the distribution shift between test data distribution and training data

Manuscript received March 10, 2023.
Manuscript revised August 22, 2023.
Manuscript publicized October 18, 2023.

[†]The authors are with Graduate School of Informatics, Nagoya University, Nagoya-shi, 464–8601 Japan.
a) E-mail: leizhou.acad@outlook.com

DOI: 10.1587/transinf.2023EDP7048

Table 1 An example of inference discrepancy. The sentences are from the WMT14 English-German training set. We choose German⇒English translation direction for it is easy to read. SRC is the source sentence, REF is the reference sentence and TR is the translation generated by a vanilla NMT model trained on WMT14 English-German training set. The sentence-level BLEU (4-gram) score of this translation is 7.29. The discrepant parts of the reference and translation sentences are marked in blue and red correspondingly.

SRC	Und wie Herr Simpson sehr richtig sagte, darf der Prozeß niemals als abgeschlossen, als gewonnen oder als vollendet betrachtet werden.
REF	As Mr Simpson has said very correctly, this is a process which we can never take for granted or regard as having come to an end.
TR	And, as Mr Simpson quite rightly said, the process must never be considered closed, won or completed.

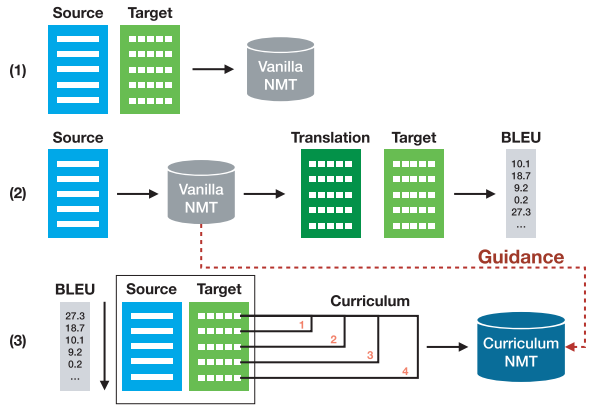


Fig. 1 The basic training scheme of inference discrepancy based curriculum learning strategy. Given a training set (1) train a vanilla; (2) translate the source sentences with the trained vanilla model and measure the discrepancy between translation and inference with BLEU; (3) sorted the training data from easy to hard by the BLEU score and train the curriculum NMT model. The analogy is that the vanilla model offer guidance to curriculum learning process of the curriculum NMT model.

distribution is another causal factor, if the test data during inference is unseen and different from the distribution of training data. Therefore inference discrepancy is a pervasive issue which can be observed on both seen and unseen data. To summarize, small inference discrepancy indicates minor distribution shift and examples causing small inference discrepancy fit better into the model distribution of a trained NMT model. Third, the NMT model is trained with loss function at the word level while evaluated with metrics at sentence level. In that light, we propose an inference-discrepancy difficulty criterion, according to which, examples with large inference discrepancy are more difficult for an NMT model to master while those with small inference discrepancy are easier. And the inference discrepancy is measured with commonly used metrics, which allows the metric signals to exert influence on model training process in an indirect way.

On the basis of the inference-discrepancy difficulty criterion, we propose a novel inference discrepancy based curriculum learning strategy, letting a regularly trained NMT model "guide" the curriculum learning process of other NMT model. We refer to the regularly trained one as the vanilla NMT model and the one undergoes curriculum learning as the curriculum NMT model. The basic scheme of proposed strategy consists of three steps: (1) train a vanilla NMT model on the training set; (2) enforce inference on the training set and measure inference discrepancy with evaluation metrics, BLEU as a default; (3) implement curriculum learning with another NMT model on the same training set. Figure 1 shows the workflow of the basic training scheme. The vanilla model and the curriculum learning model have completely identical architecture. It is like the hindsight that after skimming over a textbook, a man can roughly make an initial plan to learn things from easy to hard. In this work, we also explored several different training schemes. In the basic scheme, the vanilla NMT model and the curriculum NMT model have the same translation direction, and the inference discrepancy is measured on the forward-translation results. For comparison, we explored the training scheme where the vanilla model and the curriculum model have the opposite translation directions, and the inference discrepancy is computed on the back-translation results. Furthermore, in another training scheme, we divided the training set to train more than one vanilla models, and perform cross-validation to get inference discrepancy on unseen data. To investigate our proposed strategy from a broader perspective, we also made diversified attempts within the framework of the basic training scheme, such as experiment with different curriculum schedules or different evaluation metrics. Extensive experiments on WMT14 English ⇒ German, WMT17 Chinese ⇒ English and The Multitarget TED Talks Task (MTTT) English ⇔ German, English ⇔ Chinese, English ⇔ Russian demonstrate that our proposed method can constantly boost the performance.

This inference discrepancy based curriculum learning strategy contributes to the curriculum learning research in NMT by taking the gap between training and inference into consideration when designing the difficulty criterion and training scheme. It also has several advantageous features: (1) it is model agnostic and easy to implement since only minor modifications on the training pipeline are needed; (2) it can be easily transferred to curriculum learning research in different tasks or even in other domain by changing the evaluation metrics for inference discrepancy for different preference.

2. Preliminary

2.1 NMT Training

Let X and Y denote source and target languages and let $\mathbb{D} = \{(x^n, y^n)\}_{n=1}^{N}$ represent the training data. Let $\hat{P}_{\mathbb{D}}$ denote the training data distribution as opposed to P for model distribution. NMT model training is to learn the conditional distribution with a probabilistic model $P(y|x;\theta)$, where θ is estimated by minimizing the loss function J:

$$J(\theta) = -\mathbb{E}_{x,y\sim\hat{P}_{\mathbb{D}}} \log P(y|x;\theta), \tag{1}$$

where $\mathbb{E}_{x,y\sim\hat{P}_{\mathbb{D}}}$ is the expectation of source and target examples following the training data distribution. When both translation directions are involved, we use $\mathrm{MT}_{X\to Y}$ and $\mathrm{MT}_{Y\to X}$ for distinction.

2.2 Curriculum Learning

In essence, curriculum learning and continuation method are in the same line [12]. The basic idea of the continuation method is to construct a set of objective functions in sequence of smoothing level, e.g. $J_1(\theta),\ldots,J_K(\theta)$, and then optimize the objectives one by one, from easy to hard. Curriculum learning implements a very similar training strategy that $\theta^1,\ldots,\theta^K$ are learned with the same objective function but on different collections of training examples, from easy to hard.

We can formalize curriculum learning for NMT as follows. Given a training set $\mathbb{D}$, we use z to represent a pair of parallel sentences for simplicity, namely $z^n = (x^n, y^n), x^n, y^n \in \mathbb{D}$. Let $d(\cdot)$ denote the difficulty criterion, that the difficulty score of each training example is $d(z^n)$. An ordered set $\mathbb{D}^*$ is then obtained by ranking all N elements in $\mathbb{D}$ in ascending order of difficulty, so that:

$$\mathbb{D}^* : i < j \to d(z^i) \le d(z^j), \forall z^i, z^j \in \mathbb{D}^*. \tag{2}$$

Then, a sequence of collections $\mathbb{C}_1,\ldots,\mathbb{C}_K$ is constructed from $\mathbb{D}^*$ by collecting a sequence of training examples with similar difficulty degrees, i.e.$\mathbb{C} := \{z^i,\ldots,z^j\}, 1 \le i \le j \le N$. At training time, these collections are loaded one by one as the training set.

We would like to elaborate on some schedule details of existing curriculum learning techniques.

(1) One-pass or baby-steps: In one-pass schedule, $\mathbb{C}_1,\ldots,\mathbb{C}_K$ are mutually exclusive and $\bigcup_{k\in K} \mathbb{C}_k = \mathbb{D}^*$. Model training switches to harder examples when a new collection is taken into use. In baby-steps schedule, every collection starts from z^1, such that $\mathbb{C}_1 \subset \mathbb{C}_2 \subset \cdots \subset \mathbb{C}_K$ and $\mathbb{C}_K = \mathbb{D}^*$. When a new collection is taken into use, harder examples are merged into the current training set while easier examples are not cast aside. As baby-steps curriculum outperforms one-pass [13], it becomes the basis of more curriculum strategies.

(2) Multiple batches or a single batch: As collections are loaded one by one as the current training data, one or more than one batches are created out of it. In some works [7], [13]–[15], multiple batches are created out of one collection and K is set to a small value. By loading data collections one by one, it naturally divides the training process into K phases. In other approaches [3], [4], [16], data collection is made at every step and then a batch is created out of it.

(3) Preset or dynamic: We identify a curriculum strategy as preset if both the scope of the collections and the training steps spent on each collection can be set before training. Otherwise, if either can only be set during training, we identify it as dynamic. One type of dynamic method is sometimes addressed as self-paced learning [5], [15], [17]–[20], in which the training steps for each collection are determined based on the model training state. Training continues on one data collection until meeting certain conditions, then moves forward. Another type constructs its difficulty criterion based on model training states [9].

In Sect. 3, we will use these preliminaries to categorize and explain our proposed curriculum learning strategy and the training schemes.

3. Inference Discrepancy Based Curriculum Learning

In this section, we propose a inference-discrepancy difficulty criterion and curriculum learning strategy based on inference discrepancy. Inference discrepancy, namely the discrepancy between translation and reference is a reflection of the learning difficulty from the perspective of the trained vanilla NMT model. Such inference discrepancy can be measured by commonly used metrics. Implementing the inference-discrepancy difficulty criterion differently, we put forward different training schemes and corporate the basic scheme with diversified attempts to investigate our proposed curriculum learning strategy from a broader perspective.

3.1 Training Schemes

3.1.1 Basic Scheme: Forward-Translation Inference

We denote VMT as the vanilla NMT model and CLMT as the curriculum NMT model. For a pair of languages X, Y, we first learn a vanilla NMT model $\mathrm{VMT}_{X\to Y}$ with parameters φ by minimizing the objective function:

$$J(\varphi) = \mathbb{E}_{x,y\sim\hat{P}_{\mathbb{D}}} L(f(x;\varphi), y), \tag{3}$$

where $f(\varphi)$ represents NMT model with a function, capable of mapping sentences from the source side to the target side, and $L(\cdot)$ is the loss function. Letting $d(\cdot)$ denote the difficulty criterion, we measure the degree of inference discrepancy with sentence-level BLEU score, i.e. for $x, y \in \mathbb{D}$:

$$d(x, y) = -\mathrm{BLEU}(f(x;\varphi), y). \tag{4}$$

As mentioned in Sect. 2, we then sort all training examples by ascending difficulty to obtain an ordered set $\mathbb{D}^*$. Afterward, construct a sequence of data collections $\mathbb{C}_1,\ldots,\mathbb{C}_K$, from easy to hard. Eventually, we can train $\mathrm{CLMT}_{X\to Y}$ model with a curriculum as:

$$\begin{aligned}
\theta^1 &= \arg\max_{\theta} \mathbb{E}_{x,y\sim\hat{P}_{\mathbb{C}_1}} \log P(y|x;\theta),\\
\theta^2 &= \arg\max_{\theta} \mathbb{E}_{x,y\sim\hat{P}_{\mathbb{C}_2}} \log P(y|x;\theta),\\
&\vdots\\
\theta^1 &\to \theta^2 \to \cdots \to \theta^K.
\end{aligned} \tag{5}$$

In this basic scheme, the vanilla NMT model and the curriculum NMT model have the same translation direction. The inference discrepancy is computed on the forward-translation

results, namely translation is generated from the source to the target language. So, it is also referred to as forward-translation inference in the following section when comparing with other training schemes.

3.1.2 Back-Translation Inference

For comparison, we also devise a scheme in which the vanilla model is trained on the same training set $\mathbb{D}$ but in the opposite translation direction. We represent this reversed vanilla model as $\mathrm{VMT}_{Y\to X}$ with parameters γ. The inference discrepancy is then computed on the back-translation results:

$$J(\gamma) = \mathbb{E}_{y,x\sim\hat{P}_{\mathbb{D}}} L(f(y;\gamma),x), \tag{6}$$

$$d(x,y) = -\mathrm{BLEU}(f(y,\gamma),x). \tag{7}$$

For distinguishing, we use d_{FT} to represent inference discrepancy on the forward-translation results and d_{BT} to represent inference discrepancy on the back-translation results.

As an variant of the forward and the back translation inference, we also take the average of d_{FT} and d_{BT} to measure the inference discrepancy on both forward and back-translation results:

$$d_{\mathrm{avg}}(x,y) = \mathrm{AVG}(d_{\mathrm{FT}}(x,y), d_{\mathrm{BT}}(x,y)). \tag{8}$$

3.1.3 Cross-Validation Inference

In both the forward-translation and the back-translation inference, the vanilla NMT model is trained and enforced to inference on the same training set, that means all data for inference has been seen during training. As described in Sect. 1, inference on unseen data is also a causal factor for the gap between training and inference. To address this issue, we divide the training set into subsets to train multiple vanilla NMT models separately and adopt cross-validation method to enforce inference with each model on its unseen subset. To make the computational cost low and affordable to us, we carry on this scheme with 2-fold cross-validation. The advantages of it is that all examples in the training set are used for inference once and only once, and it can avoid overlaps between the subsets for the training of vanilla models. In this scheme, we only look at the inference discrepancy computed on the forward-translation results.

For the 2-fold cross-validation, We first divide the training set $\mathbb{D}$ into two subsets $\{\mathbb{D}_1, \mathbb{D}_2\}$, then train one vanilla NMT model $\mathrm{VMT}^{(1)}_{X\to Y}$ with $\mathbb{D}_1$ and train the other $\mathrm{VMT}^{(2)}_{X\to Y}$ with $\mathbb{D}_2$. After training, we perform cross validation with both vanilla models, letting $\mathrm{VMT}^{(1)}_{X\to Y}$ do inference on $\mathbb{D}_2$ and $\mathrm{VMT}^{(2)}_{X\to Y}$ do the same on $\mathbb{D}_1$. Then the two sets of inference sentences are combined for inference discrepancy measurements. The rest process is the same as the basic scheme. With a simple setting, we can generally assure that the inference discrepancy is computed on the translation results of unseen data.

3.2 Curriculum Schedule

We train the vanilla model VMT and the curriculum learning model CLMT for the same number of time steps T. Scheduling is to allocate time steps for each data collection properly. With the categories described in Sect. 2, we identify our curriculum schedule as baby-steps and multiple batches. We now review data collection with notations. Given the ordered set $\mathbb{D}^* = \{z^1, \ldots, z^i, \ldots, z^N\}$, N being the total number of training examples, we construct K data collections $\mathbb{C}_1, \ldots, \mathbb{C}_K$. In our setting, the size of these collections increases with equal proportions:

$$\mathbb{C}_k = \{z^1, \ldots, z^{Nk/K}\}, k = 1, \ldots, K. \tag{9}$$

With the data collections, we employ different curriculum schedules for model training. The first idea is a **preset schedule**. To better understand the nature of the optimization process, we propose three calculations for presetting training steps: equal, exponential, and staged. The number of training steps t_k for the k-th training phase is computed as:

$$\begin{aligned}
&\text{Equal}: \ t_k = \frac{1}{K}T; \\
&\text{Exponential}: \ t_k = \frac{2^{k-1}}{2^K - 1}T; \\
&\text{Staged}: \ t_k = \begin{cases} kt_0 & k < K \\ T - \sum_1^{K-1} t_k & \text{else} \end{cases}.
\end{aligned} \tag{10}$$

We use equal-preset, exp-preset, and staged-preset to represent these three settings. In the following sections, if not specified, we use the exp-preset schedule by default.

Algorithm 1: Curriculum Schedule

Data: Parallel corpus $\mathbb{D} = \{(z^n)\}_{n=1}^N$
Result: Curriculum Learning NMT Model CLMT

```
Train one or more vanilla models VMT with Eq. (3)(6) and
  obtain difficulty criterion d(·) via Eq. (4)(7)(8);
Rank examples in 𝔻 from easy to hard via Eq. (2) to get an
  ordered set 𝔻*, from which construct data collections
  ℂ_1, ..., ℂ_K according to Eq. (9);
Choose preset schedule or skip-dynamic schedule:
if preset schedule then
    Allocate training steps t_1, ..., t_K, Σ_1^K t_k = T according to
      Eq. (10).
    for k = 1, ..., K; do
        Load ℂ_k as the current training set.
        for training steps t = 1, ..., t_k do
            Train CLMT with ℂ_k
else
    for training steps t = 1, ..., T do
        for k = 1, ..., K; do
            Load ℂ_k as the current training set.
            while not reaching patience do
                Train CLMT with ℂ_k
```

We also adopt a **dynamic** method to further approximate the idea, in which the loading of data collections is depending on the training state. When model training has converged on the current data collection, the training will progress toward the next phase and the next data collection will be loaded. This process repeats until finishing with all data collections and all training steps. Based on our observations of the preset schedule, skipping to the next training phase before full convergence on the current data collection is quite important at the initial stage. Therefore, we implement early stopping at earlier training phases based on the valid performance, which is controlled by a hyperparameter, patience, in practice. We address it as skip-dynamic in this work. The curriculum learning algorithm of the preset and skip-dynamic schedule can be seen from Algorithm 1, in which preset schedule follows line 4-9 while skip-dynamic schedule follows line 10-15.

3.3 Evaluation Metrics

Contextual embedding based evaluation metrics BLEURT [21] and COMET [22] are often used for MT evaluation aside from the BLEU score. As these metrics are learned on the basis of a large-scale pre-trained language model, it can better capture semantic information such as synonyms for evaluation. Sometimes, they are reported to have higher correlation with human evaluation result. We can replace BLEU metrics in the difficulty criterion with BLEURT or COMET in the basic training scheme described earlier:

$$d(x, y) = -\text{BLEURT}(f(y, \gamma), x), \tag{11}$$

or:

$$d(x, y) = -\text{COMET}(f(y, \gamma), x). \tag{12}$$

4. Experiments

4.1 Datasets

We validate the proposed curriculum learning strategy on two large-scale benchmarks WMT14 EN ⇒ DE and WMT17 ZH ⇒ EN and three small-scale datasets from MTTT[†], including EN ⇔ DE, EN ⇔ ZH, EN ⇔ RU, on both directions.

For WMT14 EN ⇒ DE, the training set consists of 4.5m parallel sentences. We adopt newstest2012 and newstest2013 as the validation set, newstest2014 as the test set. Following the common practice on this benchmark, we also use shared vocabulary. From WMT17 ZH ⇒ EN, we extract 20m sentence pairs as the training set [23]. We adopt newsdev2017 as the validation set and newstest2017 as the test set.

For the MTTT dataset, three language pairs EN ⇔ DE, EN ⇔ ZH, EN ⇔ RU each consists of 152k, 169k, 180k training data. We simply use the validation set and test set provided by MTTT.

We preprocess all datasets via BPE [24] with 32k merge operations.

4.2 Model Settings

We validate our curriculum learning strategy on Transformer [25] with Fairseq[††] [26]. We choose Transformer-BASE as the baseline because recent curriculum learning approaches report their results on the basis of it.

Following the common practice in NMT using Transformer-BASE as the baseline, we use Adam optimizer [27] and the label smoothing is set to 0.1 for both vanilla model and curriculum learning model training. For all experiments, we check for early stopping while training.

For large-scale benchmarks WMT14 EN ⇒ DE and WMT17 ZH ⇒ EN, we empirically set 128k tokens for a batch for both vanilla model and curriculum learning model training. The total training steps T is 300k, and the learning rate warms up to 5×10^{-4} for 160k steps and then decays. The beam size is set to 5. Tuned on the validation set, the length penalty is 0.6 for WMT14 EN ⇒ DE while 1.0 for WMT17 ZH ⇒ EN and the dropout rate is 0.3 for WMT14 EN ⇒ DE while 0.2 for WMT17 ZH ⇒ EN.

For small-scale datasets MTTT EN ⇔ DE, EN ⇔ ZH, EN ⇔ RU, we set 16k tokens for a batch. The beam size is set to 5 and length penalty to 0.6 for WMT14 EN ⇒ DE while for the rest datasets the length penalty is set to 1. The max training steps is 40k and the learning rate warms up for 4k steps and then decays. The beam size is 5 and length penalty is 1. The dropout rate is simply set to 0.3 for all language pairs and directions.

We evaluate on an ensemble of 5 checkpoints to avoid stochasticity and report lowercased, tokenized BLEU [28] for comparison with previous work. When comparing with the performance implementing existing approaches, we report average and standard deviation of 3 independent training runs with different initial values for the vanilla model and the curriculum model, while the initial values for the vanilla NMT model and curriculum model are the same for each training run. Statistical significance test is made according to Collins *et al.* [29]. Otherwise, only the results of training with predefined initial values are reported.

All our experiments are conducted with 4 NVIDIA Quadro GV100 GPUs.

4.3 Curriculum Settings

The vanilla NMT model and the curriculum NMT model share the same Transformer-BASE setting. To measure the inference discrepancy, we use `fairseq-score` to compute sentence-level BLEU score, which implements the NIST smoothing method [30] by default. And for evaluation metrics comparison experiments, we use BLEURT[†††] and

[†]The Multitarget TED Talks Task (MTTT), is a collection of multitarget bitexts based on TED Talks extracted from WIT[3] (https://www.cs.jhu.edu/~kevinduh/a/multitarget-tedtalks/).

[††]https://github.com/pytorch/fairseq

[†††]https://github.com/google-research/bleurt

Table 2 BLEU evaluation results on large-scale bench marks WMT14 EN ⇒ DE and WMT17 ZH ⇒ EN. We adopt different difficulty criteria from existing curriculum learning (CL) methods, including sentence rarity from Plantanios et al. [3], sentence perplexity with BERT from Zhou et al. [7], and sentence norm with fastText from Liu et al. [4]. For approaches using multiple difficulty criteria, we only adopt the best performed one as reported. For the results of our proposed methods, "⇑/↑" indicates significant difference ($p < 0.01/0.05$) from Transformer BASE.

#	Model	WMT14 EN ⇒ DE		WMT17 ZH ⇒ EN	
		BLEU	Δ	BLEU	Δ
	Baseline and Related CL Methods				
1	Transformer-BASE	27.63	-	23.78	-
2	+ Sentence Rarity (Competence-based CL)	28.10	0.47	24.12	0.34
3	+ Sentence Perplexity with BERT (Uncertanty-aware CL)	28.08	0.45	24.10	0.32
4	+ Sentence Norm with fastText (Norm-based CL)	27.83	0.20	24.53	0.75
	Proposed Method				
5	+ FT-BLEU-ExpPreset	28.48(±0.39)$^{\uparrow}$	0.85	24.63(±0.25)$^{\uparrow}$	0.85
6	+ FT-BLEU-SkipDynamic	**28.53 (±0.45)**$^{\Uparrow}$	0.90	**24.87(±0.23)**$^{\Uparrow}$	1.09

COMET† for inference discrepancy measurements. Following the recent curriculum approach using baby-steps, we also choose $K = 4$ for training and data collection [7], [14]. For the stage-preset schedule, t_0 is set to 20k, and for skip-dynamic schedule, we tune the patience value from 2 to 5 and choose 2 for related experiments.

Implementing the proposed curriculum learning strategy, we investigate different training schemes together with various evaluation metrics and different curriculum schedules. We name different experiment settings by "Training Scheme-Metrics-Schedule". For different training schemes:

- **FT-** represents the basic training scheme, i.e. forward-translation (FT) inference;
- **BT-** represents back-translation (BT) inference;
- **FT/BT-** represents an average of forward-translation (FT) and back-translation (BT) inference;
- **XVAL-** represents cross-validation inference.

Basic training scheme with different evaluation metrics for inference discrepancy:

- **FT-BLEU-**;
- **FT-BLEURT-**;
- **FT-COMET-**.

Basic training scheme with BLEU by default following different curriculum schedule:

- **FT-BLEU-EqualPrest** using equal-preset schedule;
- **FT-BLEU-ExpPreset** using exp-preset schedule;
- **FT-BLEU-StagedPreset** using staged-preset schedule;
- **FT-BLEU-SkipDynamic** using skip dynamic schedule.

†https://github.com/Unbabel/COMET

4.4 Results and Analysis

4.4.1 Results

To investigate the effectiveness of using sentence-level BLEU as the difficulty criterion, we compare sentence level BLEU with other difficulty criteria used in existing curriculum learning methods, including sentence rarity [3], sentence perplexity [7] with BERT [31], and sentence norm [4] derived from fastText [32]. Transformer-BASE is used as the backbone and the vanilla model for learning difficulty measurements. Results are as shown in Table 2. All these difficulty criteria outperform the strong baseline of Transformer-BASE on the WMT14 EN ⇒ DE and WMT17 ZH ⇒ EN large-scale benchmarks. We use the basic forward-translation training scheme together with the exp-preset schedule and skip-dynamic schedule for a fair comparison across different difficulty criteria. Among these results, sentence rarity and sentence perplexity have similar performances and our proposed sentence-level BLEU shows an even better results. According to the experimental results, for both two language pairs, curriculum learning can boost performance over the strong baseline. And if we compare the two curriculum schedule, the skip-dynamic schedule outperforms the exp-preset on both benchmarks.

We also conduct experiments on a small-scale MTTT datasets with three language pairs for both directions [33]. As shown in Table 3, curriculum learning with an exp-preset schedule can boost the performance on all the translation directions. But different from the results of the large-scale datasets, curriculum learning with a skip-dynamic schedule can barely bring improvements. We attribute it to the quick overfitting on the small-scale datasets. It impairs the effect of smoothing the optimization curve with the curriculum learning method as the model can get overfit on one data collection very quickly before moving to the next data collection.

Table 3 BLEU evaluation results on small-scale dataset MTTT. For the results of our proposed methods, "⇑/↑" indicates significant difference ($p < 0.01/0.05$) from Transformer BASE. The model name Row 3-4 is the same as in Table 2, we use a short description due to the length of this table.

#	Model	MTTT ENDE		MTTT ENZH		MTTT ENRU	
		EN⇒DE	DE⇒EN	EN⇒ZH	ZH⇒EN	EN⇒RU	RU⇒EN
	Baseline and Related CL Methods						
1	Transformer-BASE	28.27	34.28	11.12	17.31	19.24	24.25
2	+ Sentence Rarity	28.57	34.28	11.10	17.37	19.39	24.60
3	+ Sentence Perplexity	28.51	34.24	11.23	17.30	19.27	24.47
4	+ Sentence Norm	28.30	34.35	11.21	18.00	19.78	24.80
	Proposed Method						
5	+ FT-BLEU-ExpPreset	**28.66(±0.26)**$^{\uparrow}$	**34.73(±0.45)**$^{\uparrow}$	**11.36(±0.08)**	**18.10(±0.09)**$^{\uparrow}$	**19.81(±0.16)**$^{\uparrow}$	**24.83(±0.18)**$^{\uparrow}$
6	+ FT-BLEU-SkipDynamic	28.39(±0.37)	34.20(±0.40)	11.19(±0.02)	17.37(±0.09)	19.30(±0.21)	24.76(±0.17)

Table 4 Results of different inference method, namely forward-translation, back-translation, an average of forward and back translation and cross-validation.

#	Model	WMT14 EN⇒DE	WMT17 ZH⇒EN
1	FT-BLEU-ExpPreset	28.47	24.60
2	BT-BLEU-ExpPreset	**28.51**	**24.83**
3	FT/BT-BLEU-ExpPreset	28.31	24.31
4	XVAL-BLEU-ExpPreset	28.45	24.66

Table 5 Results of curriculum learning with different settings of preset schedule.

#	Model	WMT14 EN⇒DE	WMT17 ZH⇒EN
1	FT-BLEU-EqualPreset	28.04	24.59
2	FT-BLEU-ExpPreset	**28.47**	24.60
3	FT-BLEU-StagedPreset	28.24	**24.70**

4.4.2 Analysis

We conduct a number of experiments for factor analysis on the WMT14 EN ⇒ DE and WMT17 ZH ⇒ EN benchmarks.

Different Inference Method In the basic scheme, the vanilla model and the curriculum learning model have the same translation direction, in which the BLEU score is computed one the forward-translation results. In the back-translation inference, the vanilla model and the curriculum learning model have opposite translation directions, in which the BLEU score computation is on the back-translation results. To assess whether and how inference discrepancy generated by forward or back translation may affect the curriculum learning performance, we perform a set of experiments, together with an average of the BLEU score on forward and back translation. We also experiment with the cross-validation inference method. For better comparison, we only use the exp-preset schedule. Table 4 compares the performances. Row 2-4 shows that all three criteria can outperform the baseline and the back-translation inference reports better results than the forward-translation inference for both language pairs. The results of average inference scores are not better than any of the single ones. The preference of the back-translation inference is consistent with the preference of using the source side linguistic features or perplexity as a difficulty criterion in related approaches. This could be attributed to the assumption that source-side difficulty is more representative of the learning difficulty of training examples. But we would like to put forward another interpretation that reversed vanilla model may compensate for the blind point of exposure bias by spotting difficult examples based on a different model distribution. But additional uncertainty also arises from the difficulty of different translation directions. As an average of forward and back translation has not exhibited any superiority, we would suggest the hypothesis that a simple combination of two scores can harm the internal logic of the two difficulty criteria and make the ranking less significant. Cross-validation inference, which is only on the forward-translation inference, shows very similar results as the forward one. We assume that as the inference discrepancy is computed on the training set, that the number of training data is fixed, whether a training example is seen or unseen to the vanilla model doesn't influence the final performance of the curriculum model after all. But we would like to try more cross-validation settings in the future to test our assumption.

Preset Schedule with Different Settings To better understand how schedules may influence curriculum learning performance, we also make a comparison between three settings of the preset schedule, i.e. equal-preset, exp-preset, and staged-preset, as listed in Eq. (10). For staged-preset schedule, t_0 is set to 20k steps. To control the factors, we only use the basic training schemes, namely forward-translation inference, on this set of experiments. As seen in Table 5, exp-preset and staged-preset schedule have better performance than equal-preset schedule. It is important to allocate more steps in the later training phases as the size of data collection increases.

Contextual Embedding Based Evaluation We also conduct a set of experiments with BLEURT and COMET on WMT14 EN⇒DE due to the computational time for sentence-level score. From Table 6 we can see, although curriculum learning with COMET or BLEURT can outperform the strong baseline, these two metrics have not shown superior performance over the BLEU score, but better performance on the same metrics when evaluating the model performance. We show the original BLEURT and COMET scores here. But as the computation of BLEURT and COMET is time-consuming when computing inference

Table 6 Results of curriculum learning with BLEU, COMET, and BLEURT as the difficulty criterion.

#	Model	WMT14 EN⇒DE		
		BLEU	BLEURT	COMET
1	& BLEURT	28.03	0.6603	0.7215
2	& COMET	27.91	0.6466	0.7284
3	& BLEU	**28.47**	0.5697	0.6448

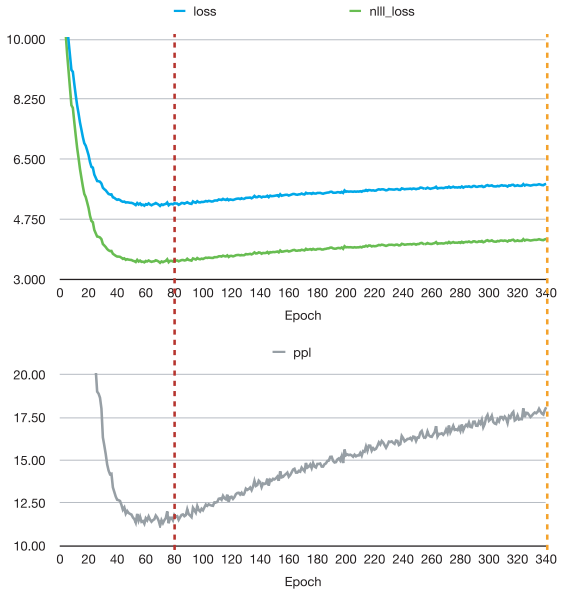


Fig. 2 Validation results of Equal schedule during the first training phase, including loss, negative log-likelihood loss (nll_loss) and perplexity (ppl).

discrepancy on the training set, we would recommend BLEU more for basic set-ups.

Step forward before over-fitting In this paper, we choose to determine the curriculum schedule through mathematical derivation, so that we can conduct interpretative analysis to the core question: when to load a more difficult data collection. Figure 2 illustrates the validation results of the equal-preset schedule at the first training phase. The orange vertical dashline shows its turning point towards the next training phase, which is 75k in our set-up, and it is when a new data collection is loaded. The red vertical dash line is the turning point of exp-preset and staged-preset schedules, both 20k. The curves of validation loss, negative log-likelihood loss (nll_loss), and perplexity (ppl) clearly show that exp-preset and staged-preset schedules step forward before over-fitting on the current data collection, while the equal-preset schedule moves forward steps later. We believe this is the reason why exp-preset and staged-preset schedules outperform the equal-preset schedule. This observation is also the basis of our proposed skip-dynamic schedule, which determines stepping forward according to training patience. However, when conducting curriculum learning on very small-scale datasets, the size of each data collection is even smaller. Training patience on such small data set is not reliable as that on large-scale datasets. We believe this is the cause of the inferior performance of the skip-dynamic schedule on the small-scale datasets.

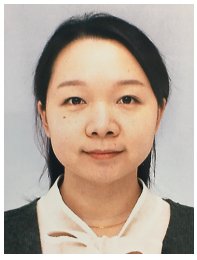
5. Conclusion

In this work, we propose this inference discrepancy based curriculum learning strategy for neural machine translation. Through a self-reflexive process, the NMT model naturally learns how to estimate learning difficulty and allocate time steps properly. We assess the effect of difficulty criteria on different translation directions, evaluation metrics, and curriculum schedules. Empirical results show that the proposed curriculum scheme under various set-ups constantly achieves performance boosts over the strong baseline. In the future, we are interested in applying this inference discrepancy based curriculum learning strategy to other scenarios, e.g., non-autoregressive generation [34]–[36].

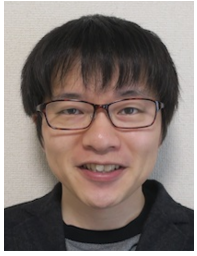
References

[1] J.L. Elman, "Learning and development in neural networks: The importance of starting small," Cognition, vol.48, no.1, pp.71–99, 1993.
[2] T. Kocmi and O. Bojar, "Curriculum learning and minibatch bucketing in neural machine translation," RANLP, Varna, Bulgaria, pp.379–386, INCOMA Ltd., 2017.
[3] E.A. Platanios, O. Stretcu, G. Neubig, B. Poczos, and T. Mitchell, "Competence-based curriculum learning for neural machine translation," NAACL, Minneapolis, Minnesota, pp.1162–1172, Association for Computational Linguistics, 2019.
[4] X. Liu, H. Lai, D.F. Wong, and L.S. Chao, "Norm-based curriculum learning for neural machine translation," ACL, Online, pp.427–436, Association for Computational Linguistics, 2020.
[5] Z.-Y. Dou, A. Anastasopoulos, and G. Neubig, "Dynamic data selection and weighting for iterative back-translation," EMNLP, Online, pp.5894–5904, Association for Computational Linguistics, 2020.
[6] T. Mohiuddin, P. Koehn, V. Chaudhary, J. Cross, S. Bhosale, and S. Joty, "Data selection curriculum for neural machine translation," ArXiv, vol.abs/2203.13867, 2022.
[7] Y. Zhou, B. Yang, D.F. Wong, Y. Wan, and L.S. Chao, "Uncertainty-aware curriculum learning for neural machine translation," ACL, Online, pp.6934–6944, Association for Computational Linguistics, 2020.
[8] X. Zhang, G. Kumar, H. Khayrallah, K. Murray, J. Gwinnup, M.J. Martindale, P. McNamee, K. Duh, and M. Carpuat, "An empirical exploration of curriculum learning for neural machine translation," ArXiv, vol.abs/1811.00739, 2018.
[9] C. Xu, B. Hu, Y. Jiang, K. Feng, Z. Wang, S. Huang, Q. Ju, T. Xiao, and J. Zhu, "Dynamic curriculum learning for low-resource neural machine translation," COLING, Barcelona, Spain (Online), pp.3977–3989, International Committee on Computational Linguistics, 2020.
[10] M. Ranzato, S. Chopra, M. Auli, and W. Zaremba, "Sequence level training with recurrent neural networks," ICLR (Poster), 2016.
[11] W. Zhang, Y. Feng, F. Meng, D. You, and Q. Liu, "Bridging the gap between training and inference for neural machine translation," ACL, Florence, Italy, pp.4334–4343, Association for Computational Linguistics, July 2019.
[12] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," Proc. 26th Annual International Conference on Machine Learning, New York, NY, USA, pp.41–48, Association for Computing Machinery, 2009.
[13] V. Cirik, E. Hovy, and L.P. Morency, "Visualizing and understanding curriculum learning for long short-term memory networks," ArXiv, vol.abs/1611.06204, 2016.

[14] X. Zhang, P. Shapiro, G. Kumar, P. McNamee, M. Carpuat, and K. Duh, "Curriculum learning for domain adaptation in neural machine translation," NAACL, Minneapolis, Minnesota, pp.1903–1915, Association for Computational Linguistics, 2019.
[15] L. Zhou, L. Ding, K. Duh, S. Watanabe, R. Sasano, and K. Takeda, "Self-guided curriculum learning for neural machine translation," IWSLT, Bangkok, Thailand (online), Association for Computational Linguistics, Aug. 2021.
[16] M. Zhang, F. Meng, Y. Tong, and J. Zhou, "Competence-based curriculum learning for multilingual machine translation," EMNLP, 2021.
[17] M.P. Kumar, B. Packer, and D. Koller, "Self-paced learning for latent variable models," NIPS, p.1189–1197, Curran Associates Inc., 2010.
[18] L. Jiang, D. Meng, Q. Zhao, S. Shan, and A. Hauptmann, "Self-paced curriculum learning," AAAI, Austin, Texas, USA, pp.2694–1700, AAAI Press, 2015.
[19] Y. Wan, B. Yang, D.F. Wong, Y. Zhou, L.S. Chao, H. Zhang, and B. Chen, "Self-paced learning for neural machine translation," EMNLP, Online, pp.1074–1080, Association for Computational Linguistics, 2020.
[20] D. Ruiter, J. van Genabith, and C. España-Bonet, "Self-induced curriculum learning in self-supervised neural machine translation," EMNLP, Online, pp.2560–2571, Association for Computational Linguistics, 2020.
[21] T. Sellam, D. Das, and A. Parikh, "BLEURT: Learning robust metrics for text generation," ACL, Online, pp.7881–7892, Association for Computational Linguistics, July 2020.
[22] R. Rei, C. Stewart, A.C. Farinha, and A. Lavie, "COMET: A neural framework for MT evaluation," EMNLP, Online, pp.2685–2702, Association for Computational Linguistics, Nov. 2020.
[23] H. Hassan, A. Aue, C. Chen, V. Chowdhary, J. Clark, C. Federmann, X. Huang, M. Junczys-Dowmunt, W. Lewis, M. Li, et al., "Achieving human parity on automatic chinese to english news translation," arXiv, vol.abs/1803.05567, 2018.
[24] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," ACL, Berlin, German, pp.1715–1725, The Association for Computer Linguistics, 2016.
[25] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," NIPS, Red Hook, NY, USA, pp.6000–6010, Curran Associates Inc., 2017.
[26] M. Ott, S. Edunov, A. Baevski, A. Fan, S. Gross, N. Ng, D. Grangier, and M. Auli, "fairseq: A fast, extensible toolkit for sequence modeling," NAACL-HLT, Minneapolis, Minnesota, pp.48–53, Association for Computational Linguistics, 2019.
[27] D.P. Kingma and J. Ba, "Adam: A method for stochastic optimization," ICLR (Poster), San Diego, CA, USA, OpenReview.net, 2015.
[28] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," ACL, Philadelphia, Pennsylvania, USA, pp.311–318, Association for Computational Linguistics, 2002.
[29] M. Collins, P. Koehn, and I. Kučerová, "Clause restructuring for statistical machine translation," ACL, Ann Arbor, Michigan, pp.531–540, The Association for Computer Linguistics, 2005.
[30] B. Chen and C. Cherry, "A systematic comparison of smoothing techniques for sentence-level bleu," WMT, Baltimore, Maryland, USA, pp.362–367, Association for Computational Linguistics, 2014.
[31] J. Devlin, M.W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," NAACL, pp.4171–4186, Association for Computational Linguistics, June 2019.
[32] A. Joulin, E. Grave, P. Bojanowski, M. Douze, H. Jégou, and T. Mikolov, "Fasttext.zip: Compressing text classification models," arXiv preprint arXiv:1612.03651, 2016.
[33] K. Duh, "The multitarget ted talks task." http://www.cs.jhu.edu/~kevinduh/a/multitarget-tedtalks/, 2018.
[34] J. Gu, J. Bradbury, C. Xiong, V.O.K. Li, and R. Socher, "Non-autoregressive neural machine translation," ICLR, Vancouver, BC, Canada, OpenReview.net, 2018.
[35] D. Wu, L. Ding, F. Lu, and J. Xie, "Slotrefine: A fast non-autoregressive model for joint intent detection and slot filling," EMNLP, Online, pp.313–319, Association for Computational Linguistics, 2020.
[36] L. Ding, L. Wang, D. Wu, D. Tao, and Z. Tu, "Context-aware cross-attention for non-autoregressive translation," COLING, Barcelona, Spain (Online), pp.4396–4402, International Committee on Computational Linguistics, 2020.



Lei Zhou received the B.S. degree from Beijing University of Posts and Telecommunications, Beijing, China, in 2010 and the M.S. degree from Institute of Scientific and Technical Information of China, Beijing, China, in 2016. She is currently pursuing a Ph.D. degree in Informatics at Nagoya University, Nagoya, Japan. Her research interest includes machine translation, machine translation evaluation, and domain adaptation.



Ryohei Sasano is currently an Associate Professor at Nagoya University, Japan. He obtained his Ph.D. in Information Science and Technology from the University of Tokyo, Japan. In addition to his academic career, Prof. Sasano was a Project Researcher at Kyoto University and an Assistant Professor at the Tokyo Institute of Technology. His area of interest is natural language processing, in particular frame semantics, predicate-argument structure analysis, and anaphora resolution.

Koichi Takeda is currently the Director and Professor of the Future Value Creation Center at Nagoya University, Nagoya, Japan. He joined IBM Research-Tokyo in 1983 and contributed to the productization of machine translation, insight discovery from electronic medical records, and text analytics. He was a member of the core team for developing the question-answering system Watson during 2007-2011. His area of interest is natural language processing and explainable artificial intelligence. He obtained his Ph.D. in Informatics from the Graduate School of Informatics, Kyoto University, Japan.