Dynamic Curriculum Learning for Low-Resource Neural Machine Translation

Chen Xu[1], Bojie Hu[2], Yufan Jiang[3], Kai Feng[1], Zeyang Wang[1],
Shen Huang[2], Qi Ju[2]*, Tong Xiao[1,4], Jingbo Zhu[1,4]*

[1]NLP Lab, School of Computer Science and Engineering,
Northeastern University, Shenyang, China
[2]Tencent Minority-Mandarin Translation
[3]Smart Platform Product Department of Tencent Inc., China
[4]NiuTrans Research, Shenyang, China
`xuchenneu@163.com, {xiaotong,zhujingbo}@mail.neu.edu.cn`
`{bojiehu, garyyfjiang, springhuang, damonju}@tencent.com`

Abstract

Large amounts of data has made neural machine translation (NMT) a big success in recent years. But it is still a challenge if we train these models on small-scale corpora. In this case, the way of using data appears to be more important. Here, we investigate the effective use of training data for low-resource NMT. In particular, we propose a dynamic curriculum learning (DCL) method to reorder training samples in training. Unlike previous work, we do not use a static scoring function for reordering. Instead, the order of training samples is dynamically determined in two ways - loss decline and model competence. This eases training by highlighting easy samples that the current model has enough competence to learn. We test our DCL method in a Transformer-based system. Experimental results show that DCL outperforms several strong baselines on three low-resource machine translation benchmarks and different sized data of WMT'16 En-De.

1 Introduction

Recently, neural machine translation (NMT) has demonstrated impressive performance improvements and became the de-facto standard (Sutskever et al., 2014; Bahdanau et al., 2015; Wu et al., 2016; Vaswani et al., 2017). However, like other neural methods, NMT is data-hungry. This makes it challenging when we train such a model in low-resource scenarios (Koehn and Knowles, 2017). Researchers have developed promising approaches to low-resource NMT. Among these are data augmentation (Sennrich et al., 2016a; Cheng et al., 2016; Fadaee et al., 2017), transfer learning (Zoph et al., 2016; Kocmi and Bojar, 2018), and pre-trained models (Peters et al., 2018; Devlin et al., 2019). But these approaches rely on external data other than bi-text. To date, it is rare to see work on the effective use of bilingual data for low-resource NMT.

In general, the way of feeding samples plays an important role in training neural models. A good instance is that it is popular to shuffle the input data for robust training in state-of-the-art systems. More systematic studies on this issue can be found in recent papers (Bengio et al., 2009; Kumar et al., 2010; Shrivastava et al., 2016). For example, Arpit et al. (2017) have pointed out that deep neural networks tend to prioritize learning "easy" samples first. This agrees with the idea of curriculum learning (Bengio et al., 2009) in that an easy-to-hard learning strategy can yield better convergence for training.

In NMT, curriculum learning is not new. Several research groups have applied it to large-scale translation tasks although few of them discuss the issue in a low-resource setup (Zhang et al., 2018; Platanios et al., 2019; Liu et al., 2020). The first question here is how to define the "difficulty" of a training sample. Previous work resorts to functions that produce a difficulty score for each training sample. This score is then used to reorder samples before training. But the methods of this type enforce a static scoring strategy and somehow disagrees with the fact that the sample difficulty might be changing when the model is

---

updated during training. Another assumption behind curriculum learning is that the difficulty of a sample should fit the competence of the model we are training. Researchers have implicitly modeled this issue by hand-crafted curriculum schedules (Zhang et al., 2018) or simple functions (Platanios et al., 2019), whereas there has no in-depth discussion on it yet.

In this paper, we continue the line of research on curriculum learning in low-resource NMT. We propose a dynamic curriculum learning (DCL) method to address the problems discussed above. The novelty of DCL is two-fold. First, we define the difficulty of a sample to be the decline of loss (i.e., negative log-likelihood). In this way, we can measure how hard a sentence can be translated via the real objective used in training. Apart from this, the DCL method explicitly estimates the model competence once the model is updated, so that one can select samples that the newly-updated model has enough competence to learn.

DCL is general and applicable to any NMT system. In this work, we test it in a Transformer-based system on three low-resource MT benchmarks and different sized data selected from the WMT'16 En-De task. Experimental results show that our system outperforms the strong baselines and several curriculum learning-based counterparts.

2 Related work

2.1 Low-Resource NMT

Koehn and Knowles (2017) show that NMT systems result in worse translation performance in low-resource scenarios. Researchers have developed promising approaches to this problem which mainly focus on introducing external knowledge to improve low-resource NMT performance. Data augmentation (Sennrich et al., 2016a; Cheng et al., 2016; Fadaee et al., 2017) alleviates this problem by generating pseudo parallel data. A large amount of auxiliary parallel corpus from other related language pairs can be used to pre-train model parameters and transfer to target language pair (Zoph et al., 2016; Chen et al., 2017; Gu et al., 2018a; Gu et al., 2018b; Kocmi and Bojar, 2018). Pre-trained language models trained with a large amount of monolingual data (Peters et al., 2018; Devlin et al., 2019) improve the quality of NMT model significantly (Clinchant et al., 2019; Yang et al., 2020; Zhu et al., 2020).

However, these approaches rely on a large number of external resources or conditions, e.g., the auxiliary parallel corpus related to the source or target language, or a large amount of monolingual data. Sennrich and Zhang (2019) demonstrate the competitive NMT model can be trained with the appropriate hyperparameters in low-resource scenarios without any external resources. This is consistent with our motivation. The difference lies in that they focus on the model settings, and we explore the training strategy which utilizes bilingual data effectively for low-resource NMT.

2.2 Curriculum Learning

Curriculum learning (Bengio et al., 2009) is motivated by the learning strategy of biological organisms which orders the training samples in an easy-to-hard manner (Elman, 1993). Benefited from organized training, the neural network explores harder samples effectively utilizing the previous knowledge learned from easier samples. Weinshall et al. (2018) demonstrate curriculum learning speeds up the learning process, especially at the beginning of training. Curriculum learning has been applied to several tasks, including language modeling (Bengio et al., 2009), image classification (Weinshall et al., 2018), and human attribute analysis (Wang et al., 2019).

Curriculum learning has recently shown to train large-scale translation tasks efficiently and effectively by controlling the way of feeding samples. Kocmi and Bojar (2017) construct mini-batch contains sentences similar in length and linguistic phenomena, then organize the order by increased complexity in one epoch. Zhang et al. (2018) group the training samples into shards based on model-based and linguistic difficulty criteria, then train with hand-crafted curriculum schedules. Platanios et al. (2019) propose competence-based curriculum learning that select training samples based on sample difficulty and model competence. Kumar et al. (2019) use reinforcement learning to learn the curriculum automatically. Liu et al. (2020) propose a norm-based curriculum learning method based on the norm of word embedding to improve the efficiency of training an NMT system. Zhou et al. (2020) propose uncertainty-aware cur-

riculum learning. To the best of our knowledge, this is the first comprehensive discussion of curriculum learning in a low-resource setup.

On the other hand, curriculum learning is similar to data selection and data sampling methods. More similar work is that Wang et al. (2018) propose a dynamic sampling method that calculates the decline of loss during training to improve the NMT training efficiency. They start training from the full training set and then gradually decrease. This is contrary to the idea of curriculum learning.

3 Problem Definition

Let D_{train} be the training corpus and $|D_{train}|$ be the corpus size. $\mathbf{s} = (\mathbf{x}, \mathbf{y})$ is a training sample in D_{train}, where $\mathbf{x}$ is the source sentence and $\mathbf{y}$ is the target sentence. NMT systems learn a conditional probability $P(\mathbf{y}|\mathbf{x})$:

$$P(\mathbf{y}|\mathbf{x}; \boldsymbol{\theta}) = \prod_{i=1}^{|\mathbf{y}|} P(\mathbf{y}_i|\mathbf{x}, \mathbf{y}_{<i}; \boldsymbol{\theta}) \tag{1}$$

where $|\mathbf{y}|$ is the length of $\mathbf{y}$, $\boldsymbol{\theta}$ is a set of model parameters. The training objective is to seek the optimal parameters $\hat{\boldsymbol{\theta}}$ by minimizing the negative log-likelihood (NLL) of the training set:

$$
\begin{aligned}
\hat{\boldsymbol{\theta}} &= \arg\min_{\boldsymbol{\theta}} \sum_{\mathbf{s} \in D_{train}} -\log P(\mathbf{y}|\mathbf{x}; \boldsymbol{\theta}) \\
&= \arg\min_{\boldsymbol{\theta}} \sum_{\mathbf{s} \in D_{train}} \sum_{i=1}^{|\mathbf{y}|} -\log P(\mathbf{y}_i|\mathbf{x}, \mathbf{y}_{<i}; \boldsymbol{\theta})
\end{aligned}
\tag{2}
$$

Our objective is to learn better model parameters by curriculum learning in low-resource NMT. We decompose the whole training process into multiple phases $T = (t^0, t^1, t^2, \dots, t^k)$[1]. For every phase t, the sub-optimal process can be viewed as:

$$\boldsymbol{\theta}^{t+1} = \arg\min_{\boldsymbol{\theta}^t} \sum_{\mathbf{s} \in D_{train}^t} -\log P(\mathbf{y}|\mathbf{x}; \boldsymbol{\theta}^t) \tag{3}$$

where $\boldsymbol{\theta}^t$ is the model parameter at phase t. Two sub-questions (Platanios et al., 2019) in curriculum learning are separated to determine the training data D_{train}^t:

- **Sample Difficulty**. How to measure the difficulty of a training sample with a quantified value?

- **Model Competence**. How to estimate the competence of the model to arrange training data that model can learn effectively?

Previous work enforces a static scoring strategy to measure sample difficulty and encourage simple functions to estimate the model competence. In this way, the training data at each phase is pre-determined before training. But in fact, sample difficulty and model competence are not independent with the current model parameters.

A natural idea is to re-arrange the curriculum once the model is updated, so that we can select training data with appropriate difficulty for current training. We discuss these two questions in-depth in the following section.

4 Dynamic Curriculum Learning

We propose a dynamic curriculum learning method to reorder training samples in training. We determine the order of training samples dynamically, rather than using a static scoring for reordering. Besides, we propose a batching method to reduce gradient noise.

---

[1]In the standard NMT, every epoch is a phase.

4.1 Sample Difficulty

Equation 2 shows that the training objective of NMT is to minimize the loss (i.e., negative log-likelihood) of the training set. For a training sample $\mathbf{s} = (\mathbf{x}, \mathbf{y})$ at phase t, the loss is calculated as:

$$l(\mathbf{s}; \boldsymbol{\theta}^t) = -\log P(\mathbf{y}|\mathbf{x}; \boldsymbol{\theta}^t) \qquad (4)$$

The NMT model generally translates sentences with lower loss better. Based on this idea, Zhang et al. (2018) define the difficulty as the probability of the top-1 translation candidates generated by a pre-trained NMT model. While translation probability related to the training objective represents the sample difficulty accurately compared with the heuristic metrics, it still suffers from the problem of static scoring. Therefore, a natural idea is calculating the loss of training samples dynamically to measure its difficulty. To this end, we evaluate the loss of all the training samples with the fixed model parameters before each training phase:

$$d(\mathbf{s}; \boldsymbol{\theta}^t) = l(\mathbf{s}; \boldsymbol{\theta}^t) \qquad (5)$$

where $d(\mathbf{s}; \boldsymbol{\theta}^t)$ is the difficulty of the sample $\mathbf{s}$ at phase t.

While the loss shows the level that the current model can handle this sample, there also suffers from two drawbacks (Wang et al., 2018). First, the loss of a sample may be large at the initial phase but easy to decrease rapidly after a few phases. Second, one sentence with small loss may have no space to further decrease and training model with these sentences iteratively may lead to overfitting.

Therefore, we define the difficulty of a sample to be the decline of loss. In this way, we take into account the model change between the previous phase and the current phase. The decline-based sample difficulty is measured as:

$$d(\mathbf{s}; \boldsymbol{\theta}^t, \boldsymbol{\theta}^{t-a}) = \begin{cases} l(\mathbf{s}; \boldsymbol{\theta}^t) & t < a \\ \\ -\dfrac{l(\mathbf{s}; \boldsymbol{\theta}^{t-a}) - l(\mathbf{s}; \boldsymbol{\theta}^t)}{l(\mathbf{s}; \boldsymbol{\theta}^{t-a})} & t \geq a \end{cases} \qquad (6)$$

where $a \geq 1$ represents we compare the loss decline at phase $t - a$ and phase t.

Based on this difficulty metric, the sentence with low difficulty indicates the NMT model improves the predicted accuracy of its translation result significantly. Therefore, it is more likely to learn better in the next phase. On the contrary, the sentence with high difficulty indicates the current NMT model does not have enough competence to handle it and may wait to be learned at a later phase.

4.2 Model Competence

Platanios et al. (2019) propose a competence-based curriculum learning framework which defines the model competence $c(t) \in (0, 1]$ at training step t by simple functional forms:

$$c(t) = \min(1, \sqrt[p]{\frac{t}{T}(1 - c_0^p) + c_0^p}) \qquad (7)$$

where $c_0 \geq 0$ is the initial competence, p is the coefficient to control the curriculum schedule. Competence is seen as a linear function when $p = 1$ and the harder samples increase by a constant amount, square root function when $p = 2$ and the harder samples increase faster in the early phases and slower in the later phases.

However, these intuitive functions might not be universal for model competence. There is a gap between the high-resource and low-resource tasks. Limited by a small amount of data, the performance of the NMT model improves slowly at the early phases (see Section 7.1).

In this paper, we propose a dynamic estimation method, which measures model competence at every phase based on the performance of the development set. While the loss of development set is an optional

---

Algorithm 1 NMT with Dynamic curriculum learning

Input: The training set D_{train}, the development set D_{dev},
   the best BLEU value of a baseline model BLEU_T.

Output: A NMT model with dynamic curriculum learning.

1: $t = 0$; Randomly initialize the model parameters $\boldsymbol{\theta}^0$;
2: **while** $t < T$ **do**
3:    Evaluate the D_{train} and get loss $l(\mathbf{s}; \boldsymbol{\theta}^t)$ of every training sample $\mathbf{s}$;
4:    **for all** $\mathbf{s}$ in D_{train} **do**
5:       Measure the difficulty of $\mathbf{s}$ by Equation 6;
6:    **end for**
7:    Sort D_{train} based on the difficulty of every training sample;
8:    Evaluate the D_{dev} and get BLEU_t;
9:    Estimate the model competence $c(t)$ by Equation 8;
10:    Train the NMT model with the $|D_{train}| * c(t)$ easiest training samples;
11:    $t \leftarrow t + 1$;
12: **end while**

---

method, the sentence-level evaluation metric BLEU (Papineni et al., 2002) presents more superiorities (Shen et al., 2016). In this way, the model competence avoids the prior hypothesis of the training process and is related to the real performance on the unseen dataset.

Specifically, we pre-train a vanilla NMT model and record the best BLEU value on the development set as curriculum length BLEU_T. The model competence is estimated as:

$$c(t) = \min(1, \frac{\mathrm{BLEU}_t}{\mathrm{BLEU}_T * \beta}(1 - c_0) + c_0) \tag{8}$$

where BLEU_t is the BLEU at phase t, $\beta \in (0, 1]$ is a coefficient to control the curriculum speed. With a smaller β, the progress of curriculum learning is faster and the model can be trained on the entire training set earlier.

We suppose that the model has weak competence to only learn well from the easiest training samples at the initial phase and gradually has enough competence to handle the entire training set D_{train}. We measure the sample difficulty and model competence before every phase, then the $|D_{train}| * c(t)$ easiest training samples are selected to train the NMT model. Benefited from dynamic measurement, the newly-updated model has enough competence to learn samples with the appropriate difficulties.

4.3 Batching

Goyal et al. (2017) address the optimization difficulty when training a neural network with large batches and exhibit good generalization. Large batches have demonstrated better performance in high-resource NMT tasks due to the lower gradient noise scale (Ott et al., 2018; Popel and Bojar, 2018). However, this method degrades the performance in low-resource tasks due to poorer generalization (Keskar et al., 2017).

The dominant NMT batches the samples with similar lengths to speed training up (Khomenko et al., 2017). To reduce gradient noise, we propose a batching method which batches the samples based on similar difficulty in our curriculum learning method. Samples with similar difficulty indicate their losses fall at a similar rate. It might have a stabilized gradient direction and leads to better performance.

4.4 Training Strategy

Zhang et al. (2018) define two general types of curriculum learning strategy. The deterministic curriculum (Kocmi and Bojar, 2017) arranges the training samples with fixed order and performs worse due to lacking randomization. The probabilistic curriculum (Platanios et al., 2019) generates a batch uniformly sampled from the training set based on sample difficulty and model competence. The latter generally works well in the previous curriculum learning methods. However, in our preliminary experiments, we

find that the vanilla model trained by sampling performs worse or converges slower slightly than the training strategy which trains the model with the whole training set in an epoch. A possible reason is sampling might lead to unbalanced training because some samples are not fully trained due to sampling omission.

Our method dynamically measures the sample difficulty and model competence at each phase, then selects a certain proportion of easier samples to train based on model competence. It ensures that training samples are not missed due to sampling and also retains randomization to avoid overfitting. Algorithm 1 shows the overall training procedure of our method.

5 Experiment Setup

5.1 Datasets

We consider three different low-resource machine translation benchmarks from IWSLT TED talk, running experiments in IWSLT'15 Chinese-English (**Zh-En**), IWSLT'15 Thai-English (**Th-En**), and IWSLT'17 English-Japanese (**En-Ja**). We concatenate dev2010, tst2010, tst2011, tst2012, and tst2013 as the development set. We use tst2015 as the test set for Zh-En and En-Th, tst2017 for En-Ja. To simulate different amounts of training resources, we randomly subsample 50K/100K/300K sentence pairs from WMT'16 English-German dataset and denote them as **50K/100K/300K**. Furthermore, we also verify the effect of our method in the high-resource scenarios with all WMT'16 English-German training set (**4.5M**). We concatenate newstest2012 and newstest2013 as the development set and newstest2016 as the test set. Data statistics are shown in Table 1.

We tokenize the Chinese sentences using NiuTrans (Xiao et al., 2012) word segmentation tookit and Japanese sentences using MeCab[2]. For other language pairs, we apply the same tokenization using Moses scripts (Koehn et al., 2007). We learn Byte-Pair Encoding (Sennrich et al., 2016b) subword segmentation with 10,000 merge operations for IWSLT datasets and 32,000 merge operations for WMT dataset. Especially, we learn BPE with a shared vocabulary for WMT dataset.

Dataset	# Train	# Dev	# Test
IWSLT'15 En-Th	85019	4904	756
IWSLT'15 En-Zh	213377	6360	1080
IWSLT'17 En-Ja	226834	6861	1452
WMT'16 En-De	50K/100K/300K/4.5M	6003	2999

Table 1: Number of sentences in each dataset.

5.2 Model Settings

In all experiments, we use the *fairseq* (Ott et al., 2019)[3] implementation of the Transformer. Inspired by (Sennrich and Zhang, 2019), we select different hyperparameters for each dataset to build a strong baseline in low-resource NMT[4]. The model hyperparameters are shown in Table 2. We use the Adam optimizer (Kingma and Ba, 2015) with $\beta_1 = 0.9$, $\beta_2 = 0.98$, and $\epsilon = 10^{-9}$. We increase the learning rate from 10^{-7} to 0.0005 for IWSLT datasets and 0.0007 for WMT dataset with linear warmup over the first 4000 steps and decay the learning rate by inverse square root way. For all experiments, we use the dropout of 0.1 and label smoothing $\epsilon_{ls} = 0.1$ for regularization. We share the embedding and the output layers of the decoder for IWSLT experiments, share all embedding for WMT experiments.

---

[2]https://taku910.github.io/mecab/

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

[4]Universal hyperparameters lead to a weaker baseline on some low-resource data sets, even not converge. We report the results of the stronger baseline trained with tuned hyperparameters on the development set to demonstrate the effectiveness of our method.

Dataset	Embedding Size	Feed-Forward Size	Depth	Head
Th-En	512	1024	6	4
Zh-En	512	1024	6	4
En-Ja	512	1024	4	4
50K/100K/300K	512	2048	4	8
4.5M	512	2048	6	8

Table 2: Model hyperparameters for each task.

5.3 Training and Inference

We train the model with batch size consisted of 4096 tokens except for 4.5M which uses $4096 * 8$ tokens to compare with a standard setting. We evaluate the development set at every epoch during training by decoding using greedy search and calculate the BLEU[5] (Papineni et al., 2002) score. We early stop training when there is no improvement of the BLEU for 10 consecutive checkpoints. During inference, we use beam search with the beam size of 5 and length penalty of 1 for IWSLT datasets, beam size of 4 and length penalty of 0.6 for WMT datasets.

5.4 Curriculum Learning Setup

We implement our method with *fairseq* by simply modifying. Our DCL method measures the sample difficulty and model competence before every phase dynamically. While it results in extra time consumption (about 30%), it is acceptable for low-resource tasks.

In all experiments, we set $a = 1$ uniformly in Equation 6 to measure the difficulty, which means the sample difficulty takes into account two adjacent phases. For the model competence described in Equation 8, we record the best BLEU of the baseline model on the development set as the BLEU_T. Although hyperparameter with careful selection can bring improvement, we set $c_0 = 0.2$ and $\beta = 0.9$ universally for all experiments. It means we start training with the 20% easiest sentences and train the model with the whole training set when the performance achieves 90% of BLEU_T.

We use the following notations to represent different curriculum learning strategies. For the sample difficulty, we compare our method with previous heuristic metrics and two other dynamic metrics:

- Heuristic metrics: Source sentence length (**Length**) and source word rarity (**Rarity**) (Platanios et al., 2019).
- Dynamic metrics: Random difficulty value (**Random**) and the loss at the current phase (**Loss**).
- Our method: Loss decline between the previous phase and the current phase (**Decline**).

For the model competence, we experiment with the following methods:

- Functional forms: Linear (**Linear**) and square root (**Sqrt**) model competence (Platanios et al., 2019).
- Our method: Dynamic model competence (**DMC**) based on the performance.

6 Results

Table 3 summaries the experimental results with different curriculum learning methods. The existing curriculum learning methods (row 2 and row 3) can not improve performance stably or even degraded, which demonstrates the heuristic metrics are not helpful in low-resource scenarios.

With the dynamic model competence, we observe the BLEU scores of Random (row 4) fluctuates on the baseline model. Although the difficulty is measured dynamically during training, the meaningless value is not favorable for curriculum learning. We also measure the difficulty with loss dynamically (row 5), which leads to a slight improvement in the larger datasets. However, it degrades performance in the smaller datasets (Th-En, 50K, and 100K). It agrees with analysis in Section 4.1 that it is easy to fall into

---

[5]https://github.com/moses-smt/mosesdecoder/tree/master/scripts/generic/multi-bleu.perl

Method	Th-En	Zh-En	En-Ja	50K	100K	300K	4.5M
Baseline	18.71	20.64	12.33	9.21	17.56	23.84	33.49
Length + Sqrt	18.89	20.51	11.90	8.16	16.67	23.86	32.96
Rarity + Sqrt	18.92	20.65	12.78	8.83	16.58	24.23	33.17
Random + DMC	18.00	20.93	13.03	7.80	17.35	24.10	33.47
Loss + DMC	18.34	20.76	12.59	8.46	17.06	24.35	33.30
Decline + Linear	18.95	20.75	12.68	10.87	18.03	24.02	**33.61**
Decline + Sqrt	19.16	21.05	12.80	10.49	17.86	**24.63**	33.52
Decline + DMC	19.13	21.14	13.24	10.65	17.89	24.35	33.52
+Batching	**19.32**	**21.20**	**13.47**	**12.90**	**18.44**	24.45	/

Table 3: BLEU scores (%) of different methods in each tasks. +Batching represents we batch the sentences with similar difficulty.

overfitting due to repeated training on some samples with low loss, especially in the extremely scarce datasets.

Then, we test our proposed difficulty metric of Decline. With the simple competence functions (row 6 and row 7), they outperform significantly the strong baselines and previous methods. It demonstrates our proposed metric is of high relevance with sample difficulty for NMT than the other four metrics. However, the Sqrt competence function does not perform better than the Linear function in all datasets, this is not consistent with the previous conclusion (Platanios et al., 2019). The possible reason is that the model competence of some low-resource datasets improves slowly in the early phases. DMC (row 8) avoids the prior hypothesis of the performance change and achieves better or similar performance compared with the above methods.

Finally, we batch the samples with similar difficulties[6] (row 9). While the model is trained with more training steps due to padding, it achieves further improvement over our curriculum learning method in all tasks. This verifies our hypothesis that the gradient of samples with similar difficulty is more stabilized. This is an interesting result and we will explore it in future work.

Overall, the experimental results show our proposed method achieves better performance compared with strong baselines and several curriculum learning-based counterparts in the low-resource NMT tasks.

7 Analysis

We take En-De 50K dataset which achieves the most performance improvement to analyze our method. Although it is sampled from WMT dataset, we think it can demonstrate the advantages of our method obviously.

7.1 Learning Curve

We visual the learning curve for comparing the convergence of our method on WMT'16 En-De datasets of different sizes in Figure 1. One obvious difference in the learning curve is that the performance changes of high-resource datasets (300K/4.5M) during training are more similar to square root function, and low-resource datasets (50K/100K) are more similar to the linear function. This phenomenon is consistent with the above experimental results and shows the necessity of calculating the model competence self-adaptively.

We also observe that our proposed method converges faster and better than the baseline model significantly in all datasets, especially in early phases. On the extremely scarce dataset (50K), DCL improves the performance significantly by learning the bilingual data effectively.

On the other hand, our method only slightly works for high-resource tasks of 4.5M parallel sentence pairs. Although DCL improves the convergence speed in early training, the baseline model catches up

---

[6]We do not do this experiment on the En-De 4.5M dataset because it is trained with large batches.
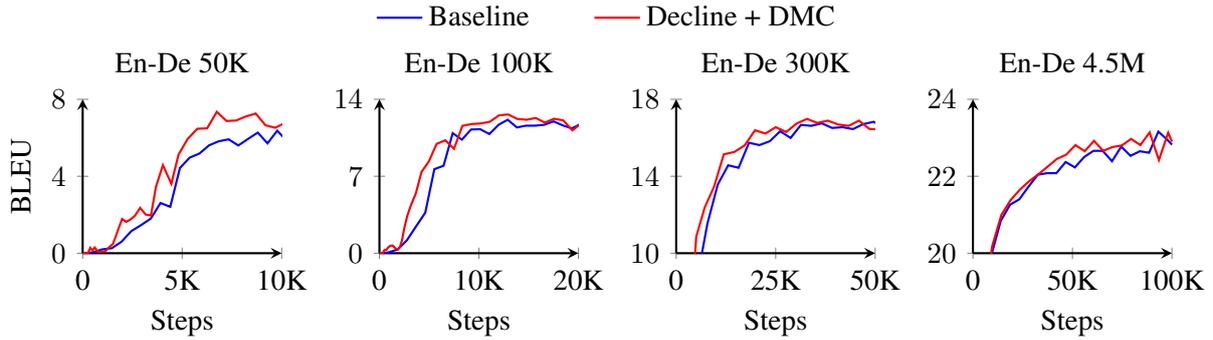
Figure 1: BLEU scores (%) of the development set on WMT'16 En-De dataset of different sizes.
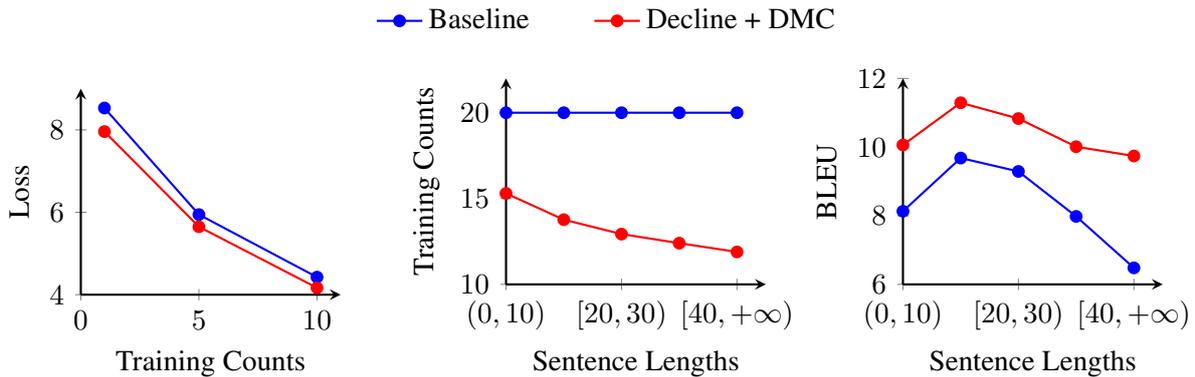

Figure 2: Average loss of all the training samples during training.

Figure 3: Training counts of samples with different sentence lengths in the training set.

Figure 4: BLEU scores (%) of different sentence lengths in the test set.

with enough training steps. A possible reason is that a large-scale parallel corpus includes sufficient knowledge and reduces the benefits of the method. The weak model trained with small-scale corpus is easy to underfit or overfit, and DCL method improves the performance by highlighting easy samples. However, the strong model trained in high-resource scenarios should pay attention to learning difficult samples. We will explore it in the future work.

7.2 Average Loss

As described in section 4, the loss of a training sample indicates whether the NMT model can predict it well. Figure 2 shows the average loss of all training samples when they are trained different counts on the En-De 50K dataset. Our method selects the samples with the fastest loss decline for learning at each training phase, which achieves the lower loss than baseline when training the same counts. It demonstrates learning with more training data is not always beneficial and the better strategy is to select dynamically according to the current model state.

7.3 BLEU with Different Sentence Lengths

Curriculum learning over-samples the easy samples and one underlying drawback is that it may reduce performance on hard samples due to less training. Although sentence length can not represent the difficulty of the training samples accurately, a widely accepted conclusion is that it is more difficult to translate longer sentences. We visual the training counts of samples with different sentence lengths in Figure 3. The training counts of samples in our method is significantly less than baseline, especially for long sentences. This also shows that long sentences are also more difficult in our measurement method.

We divide the test set into different groups according to sentence length and show the BLEU scores of baseline and our method in Figure 4. We observe that our method outperforms the baseline model

3985

significantly in translating different length sentences, especially in translating the shorter and longer sentences. This demonstrates the organized training process can achieve better translation performance for hard samples even with fewer training times.

8 Conclusion

In this paper, we propose a dynamic curriculum learning (DCL) method to explore the effective use of bilingual data for low-resource NMT. We define the difficulty of a training sample by the decline of loss and estimate the model competence self-adaptively based on the performance of the development set. Different from previous work, we re-arrange the curriculum once the model is updated, so that the training data with appropriate difficulty is learned by the current model effectively. Experimental results show that our method outperforms the strong baselines and several curriculum learning-based counterparts on several low-resource translation tasks.

DCL only modifies the training strategy without any external data, which has great practical significance for real low-resource scenarios. With a strong baseline model, we can improve the effectiveness of other semi-supervised methods, such as generating the high quality back-translation data. In the future, we will rethinking the existing training startegies and explore the application of DCL methods to more difficult tasks, such as unsupervised learning and model training with pseudo data.

Acknowledgments

References

Devansh Arpit, Stanislaw Jastrzebski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S. Kanwal, Tegan Maharaj, Asja Fischer, Aaron C. Courville, Yoshua Bengio, and Simon Lacoste-Julien. 2017. A closer look at memorization in deep networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 233–242. PMLR.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, June 14-18, 2009*, pages 41–48.

Yun Chen, Yang Liu, Yong Cheng, and Victor O. K. Li. 2017. A teacher-student framework for zero-resource neural machine translation. In Regina Barzilay and Min-Yen Kan, editors, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1925–1935. Association for Computational Linguistics.

Yong Cheng, Wei Xu, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. 2016. Semi-supervised learning for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

Stéphane Clinchant, Kweon Woo Jung, and Vassilina Nikoulina. 2019. On the use of BERT for neural machine translation. In Alexandra Birch, Andrew M. Finch, Hiroaki Hayashi, Ioannis Konstas, Thang Luong, Graham Neubig, Yusuke Oda, and Katsuhito Sudoh, editors, *Proceedings of the 3rd Workshop on Neural Generation and Translation@EMNLP-IJCNLP 2019, Hong Kong, November 4, 2019*, pages 108–117. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors,

Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers), pages 4171–4186. Association for Computational Linguistics.

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

Marzieh Fadaee, Arianna Bisazza, and Christof Monz. 2017. Data augmentation for low-resource neural machine translation. In Regina Barzilay and Min-Yen Kan, editors, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 2: Short Papers*, pages 567–573. Association for Computational Linguistics.

Priya Goyal, Piotr Dollár, Ross B. Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. 2017. Accurate, large minibatch SGD: training imagenet in 1 hour. *CoRR*, abs/1706.02677.

Jiatao Gu, Hany Hassan, Jacob Devlin, and Victor O. K. Li. 2018a. Universal neural machine translation for extremely low resource languages. In Marilyn A. Walker, Heng Ji, and Amanda Stent, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 344–354. Association for Computational Linguistics.

Jiatao Gu, Yong Wang, Yun Chen, Victor O. K. Li, and Kyunghyun Cho. 2018b. Meta-learning for low-resource neural machine translation. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 3622–3631. Association for Computational Linguistics.

Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. 2017. On large-batch training for deep learning: Generalization gap and sharp minima. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Viacheslav Khomenko, Oleg Shyshkov, Olga Radyvonenko, and Kostiantyn Bokhan. 2017. Accelerating recurrent neural network training using sequence bucketing and multi-gpu data parallelization. *CoRR*, abs/1708.05604.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Tom Kocmi and Ondrej Bojar. 2017. Curriculum learning and minibatch bucketing in neural machine translation. In Ruslan Mitkov and Galia Angelova, editors, *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017, Varna, Bulgaria, September 2 - 8, 2017*, pages 379–386. INCOMA Ltd.

Tom Kocmi and Ondrej Bojar. 2018. Trivial transfer learning for low-resource neural machine translation. In Ondrej Bojar, Rajen Chatterjee, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno-Yepes, Philipp Koehn, Christof Monz, Matteo Negri, Aurélie Névéol, Mariana L. Neves, Matt Post, Lucia Specia, Marco Turchi, and Karin Verspoor, editors, *Proceedings of the Third Conference on Machine Translation: Research Papers, WMT 2018, Belgium, Brussels, October 31 - November 1, 2018*, pages 244–252. Association for Computational Linguistics.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation, NMT@ACL 2017, Vancouver, Canada, August 4, 2017*, pages 28–39.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In John A. Carroll, Antal van den Bosch, and Annie Zaenen, editors, *ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, June 23-30, 2007, Prague, Czech Republic*. The Association for Computational Linguistics.

M. Pawan Kumar, Benjamin Packer, and Daphne Koller. 2010. Self-paced learning for latent variable models. In John D. Lafferty, Christopher K. I. Williams, John Shawe-Taylor, Richard S. Zemel, and Aron Culotta, editors, *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010. Proceedings of a meeting held 6-9 December 2010, Vancouver, British Columbia, Canada*, pages 1189–1197. Curran Associates, Inc.

Gaurav Kumar, George F. Foster, Colin Cherry, and Maxim Krikun. 2019. Reinforcement learning based curriculum optimization for neural machine translation. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 2054–2061. Association for Computational Linguistics.

Xuebo Liu, Houtim Lai, Derek F. Wong, and Lidia S. Chao. 2020. Norm-based curriculum learning for neural machine translation. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 427–436. Association for Computational Linguistics.

Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. 2018. Scaling neural machine translation. In Ondrej Bojar, Rajen Chatterjee, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno-Yepes, Philipp Koehn, Christof Monz, Matteo Negri, Aurélie Névéol, Mariana L. Neves, Matt Post, Lucia Specia, Marco Turchi, and Karin Verspoor, editors, *Proceedings of the Third Conference on Machine Translation: Research Papers, WMT 2018, Belgium, Brussels, October 31 - November 1, 2018*, pages 1–9. Association for Computational Linguistics.

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 Waleed Ammar, Annie Louis, and Nasrin Mostafazadeh, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Demonstrations*, pages 48–53. Association for Computational Linguistics.

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 of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA*, pages 311–318. ACL.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In Marilyn A. Walker, Heng Ji, and Amanda Stent, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 2227–2237. Association for Computational Linguistics.

Emmanouil Antonios Platanios, Otilia Stretcu, Graham Neubig, Barnabás Póczos, and Tom M. Mitchell. 2019. Competence-based curriculum learning for neural machine translation. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 1162–1172. Association for Computational Linguistics.

Martin Popel and Ondrej Bojar. 2018. Training tips for the transformer model. *Prague Bull. Math. Linguistics*, 110:43–70.

Rico Sennrich and Biao Zhang. 2019. Revisiting low-resource neural machine translation: A case study. In Anna Korhonen, David R. Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 211–221. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

Shiqi Shen, Yong Cheng, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. 2016. Minimum risk training for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

Abhinav Shrivastava, Abhinav Gupta, and Ross B. Girshick. 2016. Training region-based object detectors with online hard example mining. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 761–769. IEEE Computer Society.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3104–3112.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 5998–6008.

Rui Wang, Masao Utiyama, and Eiichiro Sumita. 2018. Dynamic sentence sampling for efficient training of neural machine translation. In Iryna Gurevych and Yusuke Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 2: Short Papers*, pages 298–304. Association for Computational Linguistics.

Yiru Wang, Weihao Gan, Jie Yang, Wei Wu, and Junjie Yan. 2019. Dynamic curriculum learning for imbalanced data classification. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 5016–5025. IEEE.

Daphna Weinshall, Gad Cohen, and Dan Amir. 2018. Curriculum learning by transfer learning: Theory and experiments with deep networks. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 5235–5243. PMLR.

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

Tong Xiao, Jingbo Zhu, Hao Zhang, and Qiang Li. 2012. Niutrans: An open source toolkit for phrase-based and syntax-based machine translation. In *The 50th Annual Meeting of the Association for Computational Linguistics, Proceedings of the System Demonstrations, July 10, 2012, Jeju Island, Korea*, pages 19–24. The Association for Computer Linguistics.

Jiacheng Yang, Mingxuan Wang, Hao Zhou, Chengqi Zhao, Weinan Zhang, Yong Yu, and Lei Li. 2020. Towards making the most of BERT in neural machine translation. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 9378–9385. AAAI Press.

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

Yikai Zhou, Baosong Yang, Derek F. Wong, Yu Wan, and Lidia S. Chao. 2020. Uncertainty-aware curriculum learning for neural machine translation. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 6934–6944. Association for Computational Linguistics.

Jinhua Zhu, Yingce Xia, Lijun Wu, Di He, Tao Qin, Wengang Zhou, Houqiang Li, and Tie-Yan Liu. 2020. Incorporating BERT into neural machine translation. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer learning for low-resource neural machine translation. In Jian Su, Xavier Carreras, and Kevin Duh, editors, *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 1568–1575. The Association for Computational Linguistics.