

Competence-based Curriculum Learning for Multilingual Machine Translation

Mingliang Zhang^{1*}, Fandong Meng², Yunhai Tong¹, Jie Zhou²

¹Key Laboratory of Machine Perception, School of EECS, Peking University

²Pattern Recognition Center, WeChat AI, Tencent Inc, China

{zml24, yhtong}@pku.edu.cn

{fandongmeng, withtomzhou}@tencent.com

Abstract

Currently, multilingual machine translation is receiving more and more attention since it brings better performance for low resource languages (LRLs) and saves more space. However, existing multilingual machine translation models face a severe challenge: imbalance. As a result, the translation performance of different languages in multilingual translation models are quite different. We argue that this imbalance problem stems from the different learning competencies of different languages. Therefore, we focus on balancing the learning competencies of different languages and propose *Competence-based Curriculum Learning for Multilingual Machine Translation*, named CCL-M. Specifically, we firstly define two competencies to help schedule the high resource languages (HRLs) and the low resource languages: 1) *Self-evaluated Competence*, evaluating how well the language itself has been learned; and 2) *HRLs-evaluated Competence*, evaluating whether an LRL is ready to be learned according to HRLs' *Self-evaluated Competence*. Based on the above competencies, we utilize the proposed CCL-M algorithm to gradually add new languages into the training set in a curriculum learning manner. Furthermore, we propose a novel competence-aware dynamic balancing sampling strategy for better selecting training samples in multilingual training. Experimental results show that our approach has achieved a steady and significant performance gain compared to the previous state-of-the-art approach on the TED talks dataset.

1 Introduction

With the development of natural language processing and deep learning, multilingual machine translation has gradually attracted the interest of researchers (Dabre et al., 2020). Moreover, the

multilingual machine translation model demands less space than multiple bilingual unidirectional machine translation models, making it more popular among developers (Liu et al., 2020; Zhang et al., 2020; Fan et al., 2020).

However, existing multilingual machine translation models face imbalance problems. On the one hand, various sizes of training corpora for different language pairs cause imbalance. Typically, the training corpora size of some high resource languages (HRLs) is hundreds or thousands of times that of some low resource languages (LRLs) (Schwenk et al., 2019), resulting in lower competence of LRL learning. On the other hand, translation between different languages has different difficulty, which also leads to imbalance. In general, translation between closely related language pairs is more effortless than that between distant language pairs, even if the training corpora is of the same size (Barrault et al., 2020). This would lead to low learning competencies for distant languages compared to closely related languages. Therefore, multilingual machine translation is inherently imbalanced, and dealing with this imbalance is critical to advancing multilingual machine translation (Dabre et al., 2020).

To address the above problem, existing balancing methods can be divided into two categories, i.e., static and dynamic. 1) Among static balancing methods, temperature-based sampling (Arivazhagan et al., 2019) is the most common one, compensating for the gap between different training corpora sizes by oversampling the LRLs and undersampling the HRLs. 2) Researchers have also proposed some dynamic balancing methods (Jean et al., 2019; Wang et al., 2020). Jean et al. (2019) introduce an adaptive scheduling, oversampling the languages with poorer results than their respective baselines. In addition, MultiDDS-S (Wang et al., 2020) focus on learning an optimal strategy to automatically balance the usage of training corpora

*Work was done when Mingliang Zhang was interning at Pattern Recognition Center, WeChat AI, Tencent Inc, China.

for different languages at multilingual training.

Nevertheless, the above methods focus too much on balancing LRLs, resulting in lower competencies for HRLs compared to that trained only on bitext corpora. Consequently, the performance on the HRLs by the multilingual translation model is inevitably worse than that of bitext models by a large margin (Lin et al., 2020). Besides, knowledge learned by related HRLs is also beneficial for LRLs (Neubig and Hu, 2018), while is neglected by previous approaches, limiting the performance on LRLs.

Therefore, in this paper, we try to balance the learning competencies of languages and propose a *Competence-based Curriculum Learning Approach for Multilingual Machine Translation*, named CCL-M. Specifically, we firstly define two competence-based evaluation metrics to help schedule languages, which are 1) *Self-evaluated Competence*, for evaluating how well the language itself has been learned; and 2) *HRLs-evaluated Competence*, for evaluating whether an LRL is ready to be learned by the LRL-specific HRLs’ *Self-evaluated Competence*. Based on the above two competence-based evaluation metrics, we design the CCL-M algorithm to gradually add new languages into the training set. Furthermore, we propose a novel competence-aware dynamic balancing sampling method for better selecting training samples at multilingual training.

We evaluate our approach on the multilingual Transformer (Vaswani et al., 2017) and conduct experiments on the TED talks¹ to validate the performance in two multilingual machine translation scenarios, i.e., *many-to-one* and *one-to-many* (“one” refers to English). Experimental results show that our approach brings in consistent and significant improvements compared to the previous state-of-the-art approach (Wang et al., 2020) on multiple translation directions in the two scenarios.

Our contributions² are summarized as follows:

- We propose a novel competence-based curriculum learning method for multilingual machine translation. To the best of our knowledge, we are the first that integrate curriculum learning into multilingual machine translation.
- We propose two effective competence-based

¹<https://www.ted.com/participate/translate>

²We release our code on <https://github.com/zml24/ccl-m>.

evaluation metrics to dynamically schedule which languages to learn, and a competence-aware dynamic balancing sampling method for better selecting training samples at multilingual training.

- Comprehensive experiments on the TED talks dataset in two multilingual machine translation scenarios, i.e., *many-to-one* and *one-to-many*, demonstrating the effectiveness and superiority of our approach, which significantly outperforms the previous state-of-the-art approach.

2 Background

2.1 Multilingual Machine Translation

Bilingual machine translation model translates a sentence of source language S into a sentence of target language T (Sutskever et al., 2014; Cho et al., 2014; Bahdanau et al., 2014; Luong et al., 2015; Vaswani et al., 2017), which is trained as

$$\theta^* = \operatorname{argmin}_{\theta} \mathcal{L}(\theta; S, T), \quad (1)$$

where \mathcal{L} is the loss function, θ^* is the model parameters.

Multilingual machine translation system aims to train multiple language pairs in a single model, including *many-to-one* (translation from multiple languages into one language), *one-to-many* (translation from one language to multiple languages), and *many-to-many* (translation from several languages into multiple languages) (Dabre et al., 2020). Specifically, we denote the training corpora of n language pairs in multilingual machine translation as $\{S_1, T_1\}, \{S_2, T_2\}, \dots, \{S_n, T_n\}$ and multilingual machine translation aims to train a model θ^* as

$$\theta^* = \operatorname{argmin}_{\theta} \frac{1}{n} \sum_{i=1}^n \mathcal{L}(\theta; S_i, T_i). \quad (2)$$

2.2 Sampling Methods

Generally, the size of the training corpora for different language pairs in multilingual machine translation varies greatly. Researchers hence developed two kinds of sampling methods, i.e., static and dynamic, to sample the language pairs at training (Dabre et al., 2020).

There are three mainstream static sampling methods, i.e., uniform sampling, proportional sampling, and temperature-based sampling (Arivazhagan et al., 2019). These methods sample the

language pairs by the predefined fixed sampling weights ψ .

Uniform Sampling. Uniform sampling is the most straightforward solution (Johnson et al., 2017). The sampling weight ψ_i for each language pair i of this method is calculated as follows

$$\psi_i = \frac{1}{|\mathcal{S}_{\text{lang}}|}, \quad (3)$$

where $\mathcal{S}_{\text{lang}}$ is the language sets for training.

Proportional Sampling. Another method is sampling by proportion (Neubig and Hu, 2018). This method improves the model’s performance on high resource languages and reduces the performance of the model on low resource languages. Specifically, we calculate its sampling weight ψ_i for each language pair i as

$$\psi_i = \frac{|\mathcal{D}_{\text{Train}}^i|}{\sum_{k \in \mathcal{S}_{\text{lang}}} |\mathcal{D}_{\text{Train}}^k|}, \quad (4)$$

where $\mathcal{D}_{\text{Train}}$ is the training corpora of language i .

Temperature-based Sampling. It samples the language pairs according to the corpora size exponentiated by a temperature term τ (Arivazhagan et al., 2019; Conneau et al., 2020) as

$$\psi_i = \frac{p_k^{1/\tau}}{\sum_{k \in \mathcal{S}_{\text{lang}}} p_k^{1/\tau}} \text{ where } p_i = \frac{|\mathcal{D}_{\text{Train}}^i|}{\sum_{k \in \mathcal{S}_{\text{lang}}} |\mathcal{D}_{\text{Train}}^k|}. \quad (5)$$

Obviously, $\tau = \infty$ is the uniform sampling and $\tau = 1$ is the proportional sampling. Both of them are a bit extreme from the perspective of τ . In practice, we usually select a proper τ to achieve a balanced result.

On the contrary, dynamic sampling methods (e.g., MultiDDS-S (Wang et al., 2020)) aim to automatically adjust the sampling weights by some predefined rules.

MultiDDS-S. MultiDDS-S (Wang et al., 2020) is a dynamic sampling method performing differentiable data sampling. It takes turns to optimize the sampling weights of different languages and the multilingual machine translation model, showing more significant potential than static sampling methods. This method optimizes the sample weight ψ to minimize the development loss as follows

$$\psi^* = \underset{\psi}{\operatorname{argmin}} \mathcal{L}(\theta^*; \mathcal{D}_{\text{Dev}}), \quad (6)$$

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \sum_{i=1}^n \psi_i \mathcal{L}(\theta; \mathcal{D}_{\text{Train}}), \quad (7)$$

where \mathcal{D}_{Dev} and $\mathcal{D}_{\text{Train}}$ denote the development corpora and the training corpora, respectively.

3 Methodology

In this section, we first define a directed bipartite language graph, on which we deploy the languages to train. Then, we define two competence-based evaluation metrics, i.e., the *Self-evaluated Competence* c and the *HRLs-evaluated Competence* \hat{c} , to help decide which languages to learn. Finally, we elaborate the entire CCL-M algorithm.

3.1 Directed Bipartite Language Graph

Formally, we define a directed bipartite language graph $G(V, E)$, in which one side is full of HRLs and the other side of LRLs. Each vertex v_i on the graph represents a language, and the weight of each directed edge (from HRLs to LRLs) e_{ij} indicates the similarity between a HRL i and an LRL j :

$$e_{ij} = \operatorname{sim}(i, j). \quad (8)$$

Inspired by TCS (Wang and Neubig, 2019), we measure it using vocabulary overlap and define the language similarity between language i and language j as

$$\operatorname{sim}(i, j) = \frac{|\operatorname{vocab}_k(i) \cap \operatorname{vocab}_k(j)|}{k}, \quad (9)$$

where $\operatorname{vocab}_k(\cdot)$ represents the top k most frequent subwords in the training corpus of a specific language.

3.2 Competence-based Evaluation Metrics

Self-evaluated Competence. We define how well a language itself has been learned as the *Self-evaluated Competence* c . In the following paragraphs, we first introduce the concept of *Likelihood Score* and then give a formula for calculating the *Self-evaluated Competence* in multilingual training based on the relationship between current *Likelihood Score* and the *Likelihood Score* of model trained on bitext corpus.

For machine translation, we usually use the label smoothed (Szegedy et al., 2016) cross-entropy loss \mathcal{L} to measure how well the model is trained, and calculate it as

$$\mathcal{L} = - \sum_i p_i \log_2 q_i, \quad (10)$$

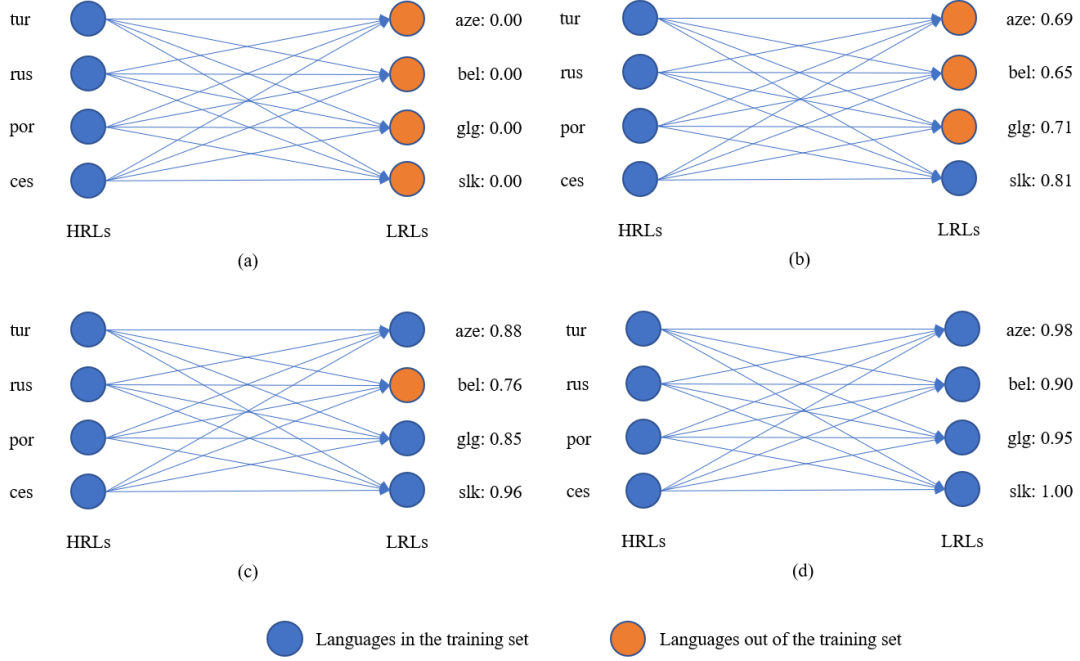


Figure 1: Diagram of the CCL-M Algorithm. This graph shows how to gradually add the LRLs to the training set $\mathcal{S}_{\text{selected}}$ using graph coloring. "aze" stands for Azerbaijani, "bel" stands for Belarusian, etc. The number after the colon indicates current *HRLs-evaluated Competence*, and suppose the corresponding threshold t is set to 0.8. Subfigure (a) represents the state before training. Subfigure (b) indicates that "slk" (Slovak) is added to the training set because the *HRLs-evaluated Competence* is higher than the threshold. Subfigure (c) indicates that "aze" (Azerbaijani) and "glg" (Glacian) are added to the training set, and Subfigure (d) indicates that all the LRLs are added to the training set. Notice we use the abbreviation of language (xxx) to indicate language pairs (xxx-eng or eng-xxx), which is more general.

where p is the label smoothed actual probability distribution, and q is the model output probability distribution³.

We find that the exponential of negative label smoothed cross-entropy loss is a likelihood to some extent, which is negatively correlated to the loss. Since neural network usually optimizes the model by minimizing the loss, we use the likelihood as a positive correlation indicator to measure competence. Therefore, we define a *Likelihood Score* s to estimate how well the model is trained as follows

$$s = 2^{-\mathcal{L}} = \prod_i q_i^{p_i}. \quad (11)$$

Inspired by Jean et al. (2019), we estimate the *Self-evaluated Competence* c of a specific language by calculating the quotient of its current *Likelihood Score* and baseline's *Likelihood Score*. Finally, we obtain the formula as follows

$$c = \frac{s}{s^*} = 2^{\mathcal{L}^* - \mathcal{L}}, \quad (12)$$

³We select 2 as the base number for all relevant formulas and experiments in this paper.

where \mathcal{L} is the current loss on the development set, \mathcal{L}^* is the *benchmark* loss of the converged bi-text model on the development set, and s and s^* are their corresponding *Likelihood Scores*, respectively.

HRLs-evaluated Competence. Furthermore, we define how well an LRL is ready to be learned as its *HRLs-evaluated Competence* \hat{c} . We believe that each LRL can learn adequate knowledge from its similar HRLs before training. Therefore, we estimate each LRL's *HRLs-evaluated Competence* by the LRL-specific HRLs' *Self-evaluated Competence*.

Specifically, we propose two methods for calculating the *HRLs-evaluated Competence*, i.e., *maximal* (CCL-M_{max}) and *weighted average* (CCL-M_{avg}). The CCL-M_{max} only migrates the knowledge from the HRL that is most similar to the LRL, so we calculate *maximal HRLs-evaluated Competence* \hat{c}_{max} for each LRL j as

$$\hat{c}_{\text{max}}(j) = c_{\text{argmax}_{i \in \mathcal{S}_{\text{HRLs}}} e_{ij}}, \quad (13)$$

where $\mathcal{S}_{\text{HRLs}}$ is the set of the HRLs.

On the other side, the CCL- M_{avg} method pays attention to all HRLs. In general, the higher the language similarity, the more knowledge an LRL can migrate from HRLs. Therefore, we calculate *weighted average HRLs-evaluated Competence* \hat{c}_{avg} for each LRL j as

$$\hat{c}_{\text{avg}}(j) = \sum_{i \in \mathcal{S}_{\text{HRLs}}} \left(\frac{e_{ij}}{\sum_{k \in \mathcal{S}_{\text{HRLs}}} e_{kj}} \cdot c_i \right). \quad (14)$$

3.3 The CCL-M Algorithm

Now we detailly describe the *Competence-based Curriculum Learning for Multilingual Machine Translation*, namely the CCL-M algorithm. The algorithm is divided into two parts: 1) curriculum learning scheduling framework, guiding when to add a language to the training set; 2) competence-aware dynamic balancing sampling, guiding how to sample languages in the training set.

First, we present how to schedule which languages on the directed bipartite language graph should be added to the training set according to the two competence-based evaluation metrics as shown in Figure 1 and Algorithm 1, where $\mathcal{S}_{\text{LRLs}}$ is the set of LRLs, and $f(\cdot)$ is the function calculating the *HRLs-evaluated Competence* \hat{c} for LRLs. Initialized as Line 1, we add all languages on the HRLs side to the training set $\mathcal{S}_{\text{selected}}$ at the beginning of training, leaving all languages on the LRLs side in the candidate set $\mathcal{S}_{\text{candidate}}$. Then, we regularly sample the development corpora of different languages and calculate current *HRLs-evaluated Competence* of the languages in the candidate set $\mathcal{S}_{\text{candidate}}$ as shown in Line 8 and 9. Further, the "if" condition in Line 13 illustrates that we would add the LRL to the training set $\mathcal{S}_{\text{selected}}$ when its *HRLs-evaluated Competence* is greater than a pre-defined threshold t . However, as the calculation of Equation 12, the upper bound of the *Self-evaluated Competence* for a specific language may not always be 1 at multilingual training. This may cause that some LRLs remain out of the training set $\mathcal{S}_{\text{selected}}$ for some thresholds. To ensure the completeness of our algorithm, we will directly add the languages still in the candidate set $\mathcal{S}_{\text{candidate}}$ to the training set $\mathcal{S}_{\text{selected}}$ after a long enough number of steps, which is described between Line 22 and Line 32.

Then, we introduce our competence-aware dynamic balancing sampling method, which is based on the *Self-evaluated Competence*. For languages in the training set $\mathcal{S}_{\text{selected}}$, we randomly select samples from the development corpora and calculate

Algorithm 1: The CCL-M Algorithm

Input: Randomly initialized model θ ;
language graph G ; *benchmark*
losses \mathcal{L}_i^* ; training corpora $\mathcal{D}_{\text{Train}}$;
development corpora \mathcal{D}_{Dev} ;
Output: The converged model θ^* ;

```

1  $\mathcal{S}_{\text{selected}} \leftarrow \mathcal{S}_{\text{HRLs}}, \mathcal{S}_{\text{candidate}} \leftarrow \mathcal{S}_{\text{LRLs}},$   

    $\psi \leftarrow 0;$ 
2 for  $i \in \mathcal{S}_{\text{selected}}$  do
3    $\psi_i \leftarrow \frac{1}{|\mathcal{S}_{\text{selected}}|};$ 
4 end
5 while  $\theta$  not converge do
6   train the model on  $\mathcal{D}_{\text{Train}}$  for some steps  

   with sampling weight  $\psi$ ;
7   for  $i \in \mathcal{S}_{\text{selected}} \cup \mathcal{S}_{\text{candidate}}$  do
8     sample  $\mathcal{D}_{\text{Dev}}$  and calculate  $\mathcal{L}_i$ ;
9      $c_i \leftarrow 2^{\mathcal{L}_i^* - \mathcal{L}_i};$ 
10  end
11  for  $i \in \mathcal{S}_{\text{candidate}}$  do
12     $\hat{c}_i \leftarrow f(G, i, c_{\mathcal{S}_{\text{HRLs}}});$ 
13    if  $\hat{c}_i \geq t$  then
14       $\mathcal{S}_{\text{selected}} \leftarrow \mathcal{S}_{\text{selected}} \cup \{i\};$ 
15       $\mathcal{S}_{\text{candidate}} \leftarrow \mathcal{S}_{\text{candidate}} \setminus \{i\};$ 
16    end
17  end
18  for  $i \in \mathcal{S}_{\text{selected}}$  do
19     $\psi_i \leftarrow \frac{1}{c_i};$ 
20  end
21 end
22 if  $\mathcal{S}_{\text{candidate}} \neq \emptyset$  then
23    $\mathcal{S}_{\text{selected}} \leftarrow \mathcal{S}_{\text{selected}} \cup \mathcal{S}_{\text{candidate}};$ 
24   while  $\theta$  not converge do
25     train the model on  $\mathcal{D}_{\text{Train}}$  for some  

     steps with sampling weight  $\psi$ ;
26     for  $i \in \mathcal{S}_{\text{selected}}$  do
27       sample  $\mathcal{D}_{\text{Dev}}$  and calculate  $\mathcal{L}_i$ ;
28        $c_i \leftarrow 2^{\mathcal{L}_i^* - \mathcal{L}_i};$ 
29        $\psi_i \leftarrow \frac{1}{c_i};$ 
30     end
31   end
32 end

```

their *Self-evaluated Competence*. Those languages with low *Self-evaluated Competence* should get more attention, therefore we simply assign the sampling weight ψ_i to each language i in the training set to the reciprocal of its *Self-evaluated Compe-*

tence, as follows

$$\psi_i \propto \frac{1}{c_i} = 2^{\mathcal{L} - \mathcal{L}^*}. \quad (15)$$

Notice that the uniform sampling is used for the training set $\mathcal{S}_{\text{selected}}$ at the beginning of training as a balancing cold-start strategy. The corresponding pseudo code can be found in Line 3.

4 Experiments

4.1 Dataset Setup

Following Wang et al. (2020), we use the 58-languages-to-English TED talks parallel data (Qi et al., 2018) to conduct experiments. Two sets of language pairs with different levels of language diversity are selected: *related* (language pairs with high similarity) and *diverse* (language pairs with low similarity). Both of them consist of 4 high resource languages (HRLs) and 4 low resource languages (LRLs).

For the *related* language set, we select 4 HRLs (Turkish: "tur", Russian: "rus", Portuguese: "por", Czech, "ces") and its related LRLs (Azerbaijani: "aze", Belarusian: "bel", Glacian: "glg", Slovak: "slk"). For the *diverse* language set, we select 4 HRLs (Greek: "ell", Bulgarian: "bul", French: "fra", Korean: "kor") and 4 LRLs (Bosnian: "bos", Marathi: "mar", Hindi: "hin", Macedonian: "mkd") as (Wang et al., 2020). Please refer to Appendix for a more detailed description.

We test two kinds of multilingual machine translation scenarios for each set: 1) *many-to-one* (M2O): translating 8 languages to English; 2) *one-to-many* (O2M): translating English to 8 languages. The data is preprocessed by SentencePiece⁴ (Kudo and Richardson, 2018) with a vocabulary size of 8k for each language. Moreover, we add a target language tag before the source and target sentences in O2M as (Johnson et al., 2017).

4.2 Implementation Details

Baseline. We select three static heuristic strategies: uniform sampling, proportional sampling, and temperature-based sampling ($\tau = 5$), and the bitext models for the baseline. In addition, we compare our approach with the previous state-of-the-art sampling method, MultiDDS-S (Wang et al., 2020). All baseline methods use the same model and the same set of hyper-parameters as our approach.

⁴<https://github.com/google/sentencepiece>

Model. We validate our approach upon the multilingual Transformer (Vaswani et al., 2017) implemented by fairseq⁵ (Ott et al., 2019). The number of layers is 6 and the number of attention heads is 4, with the embedding dimension d_{model} of 512 and the feed-forward dimension d_{ff} of 1024 as (Wang et al., 2020). For training stability, we adopt Pre-LN (Xiong et al., 2020) for the layer-norm (Ba et al., 2016) module. For M2O tasks, we use a shared encoder with a vocabulary of 64k. Similarly, for O2M tasks, we use a shared decoder with a vocabulary of 64k.

Training Setup. We use the Adam optimizer (Kingma and Ba, 2014) with $\beta_1 = 0.9$, $\beta_2 = 0.98$ to optimize the model. Further, the same learning rate schedule as Vaswani et al. (2017) is used, i.e., linearly increase the learning rate for 4000 steps to $2e-4$ and decay proportionally to the inverse square root of the step number. We accumulate the batch size to 9,600 and adopt half-precision training implemented by apex⁶ for faster convergence (Ott et al., 2018). For regularization, we also use a dropout (Srivastava et al., 2014) $p = 0.3$ and a label smoothing (Szegedy et al., 2016) $\epsilon_{ls} = 0.1$. As for our approach, we sample 256 candidates from each languages' development corpora every 100 steps to calculate the *Self-evaluated Competence* c for each language and *HRLs-evaluated Competence* \hat{c} for each LRL.

Evaluation. In practice, we perform a grid search for the best threshold t in $\{0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$, and select the checkpoints with the lowest weighted loss⁷ on the development sets to conduct the evaluation. The corresponding early stopping patience is set to 10. For target sentence generation, we set the beam size to 5 and a length penalty of 1.0. Following Wang et al. (2020), we use the SacreBLEU (Post, 2018) to evaluate the model performance. In the end, we compare our result with MultiDDS-S using paired bootstrap resampling (Koehn, 2004) for significant test.

4.3 Results

Main Results. The main results are listed in Table 1. As we can see, both methods significantly

⁵<https://github.com/pytorch/fairseq>

⁶<https://github.com/NVIDIA/apex>

⁷This loss is calculated by averaging the loss of each samples in development corpora of all languages, which is equivalent to taking the proportional weighted average of the loss for each language.

Method	M2O		O2M	
	Related	Diverse	Related	Diverse
Bitext Models	20.37	22.38	15.73	17.83
Uniform Sampling ($\tau = \infty$)	22.63	24.81	15.54	16.86
Temperature-Based Sampling ($\tau = 5$)	24.00	26.01	16.61	17.94
Proportional Sampling ($\tau = 1$)	24.88	26.68	15.49	16.79
MultiDDS (Wang et al., 2020)	25.26	26.65	17.17	18.40
MultiDDS-S (Wang et al., 2020)	25.52	27.00	17.32	18.24
CCL-M _{max} (Ours)	26.59**	28.29**	18.89**	19.53**
CCL-M _{avg} (Ours)	26.73**	28.34**	18.74**	19.53**

Table 1: Average BLEU scores (%) on test sets of the baselines and our methods. CCL-M_{max} is the CCL-M algorithm using *maximal HRLs-evaluated Competence*, CCL-M_{avg} is the CCL-M algorithm using *weighted average HRLs-evaluated Competence*. Bold indicates the highest value. "**" indicates significantly (Koehn, 2004) better than MultiDDS-S with t-test $p < 0.01$.

Method	Related M2O		Diverse M2O	
	LRLs	HRLs	LRLs	HRLs
Bi.	10.45	30.29	11.18	33.58
MultiDDS-S	22.51	28.54	22.72	31.29
CCL-M _{max}	23.14*	30.04**	23.31*	33.26**
CCL-M _{avg}	23.30*	30.15**	23.55*	33.13**

Table 2: Average BLEU scores (%) on test sets of the HRLs and the LRLs for the best baselines and our methods in M2O tasks. Bitext models ("Bi." for short) and MultiDDS-S are selected from the baselines since "Bi." performs better on the HRLs and MultiDDS-S performs better on the LRLs. Bold indicates the highest value. "*" and "**" indicates significantly better than MultiDDS-S with t-test $p < 0.05$ and $p < 0.01$, respectively.

outperform the baselines and MultiDDS with averaged BLEU scores of over +1.07 and +1.13, respectively, indicating the superiority of our approach. Additionally, the CCL-M_{avg} is slightly better than the CCL-M_{max} in more cases. This is because the CCL-M_{avg} can get more information provided by the HRLs, and can more accurately estimate when to add an LRL into the training. Moreover, we find that O2M tasks are much more complicated than M2O tasks since decoders shared by multiple languages might generate tokens in wrong languages. Consequently, the BLEU scores of O2M tasks are more inferior than M2O tasks by a large margin.

Results on HRLs and LRLs in M2O. We further study the performance of our approach on LRLs and the HRLs in M2O tasks and list the results in Table 2. As widely known, the bitext model

Method	Related O2M		Diverse O2M	
	LRLs	HRLs	LRLs	HRLs
Bi.	8.25	23.22	7.82	27.83
MultiDDS-S	15.31	19.34	13.98	22.52
CCL-M _{max}	16.54**	21.24**	14.36*	24.71**
CCL-M _{avg}	16.33**	21.14**	13.82	25.42**

Table 3: Average BLEU scores (%) on test sets of the HRLs and the LRLs for the best baselines and our methods in O2M tasks. Bitext models ("Bi." for short) and MultiDDS-S are selected from the baselines since "Bi." performs better on the HRLs and MultiDDS-S performs better on the LRLs. Bold indicates the highest value. "*" and "**" indicates significantly better than MultiDDS-S with t-test $p < 0.05$ and $p < 0.01$, respectively.

performs poorly on LRLs while performs well on HRLs. Also, we find our method performs much better than MultiDDS-S, both on LRLs and HRLs. Although our method does not strictly match the performance of the bitext model on HRLs, the gap between them is much smaller than that of MultiDDS-S and bitext models. All of the above proves the importance of balancing learning competencies of different languages.

Results on HRLs and LRLs in O2M. As shown in Table 3, our approach also performs well on the more difficult scenario, i.e., the O2M. Apparently, our approach almost doubles the performance of the LRLs from bitext models. Consistently, there is a roughly -2 and -3 BLEU decay for the HRLs in the *related* and *diverse* language sets. Compared to MultiDDS-S, both our approach in the LRLs and

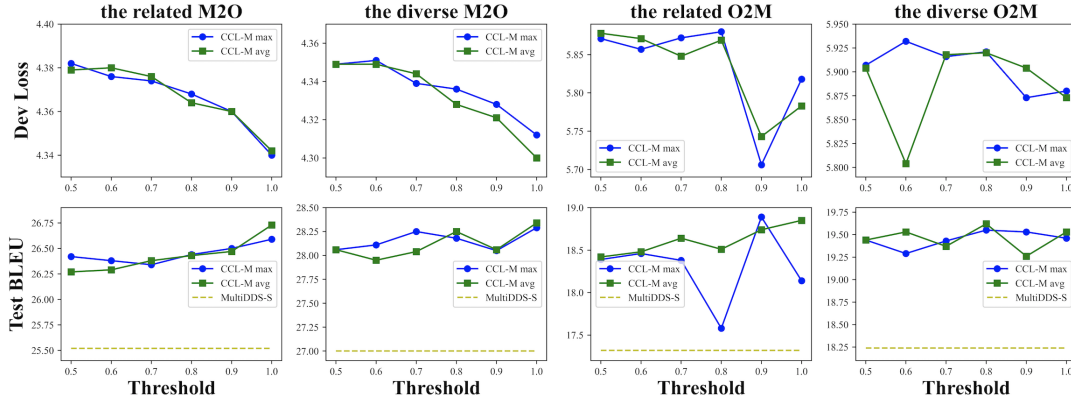


Figure 2: Weighted losses on development sets and average BLEU scores (%) on test sets for different thresholds (the abscissa) in four scenarios. The blue line represents CCL-M_{\max} , the green line represents $\text{CCL-M}_{\text{avg}}$. The yellow dotted line represents MultiDDS-S (Wang et al., 2020).

the HRLs are significantly better. This again proves the importance of balancing the competencies of different languages. Additionally, the performance on HRLs in O2M task has a larger drop from the bitext model than that in M2O task. This is because the decoder shares a 64k vocabulary for all languages in O2M tasks, but each language has only 8k vocabulary. Thus, it is easier for the model to output misleading tokens that do not belong to the target language during inference.

5 Analysis

5.1 Effects of Different Threshold t

We firstly conduct a grid search for the best *HRLs-evaluated Competence* threshold t . As we can see from Figure 2, the more HRLs are trained (the larger the threshold t is), the better the model’s performance is in M2O tasks. This phenomenon again suggests that M2O tasks are easier than O2M tasks. The curriculum learning framework performs better in the *related* set than that in the *diverse* set in M2O tasks, because languages in the *related* set are more similar. Still, our method is better than MultiDDS-S, as shown in Figure 2. This again demonstrates the positive effect of our curriculum learning framework.

Experimental results also reveal that the optimal threshold t for O2M tasks may not be 1 because more training on HRLs would not produce optimal overall performance. Furthermore, the optimal threshold for the *diverse* language set is lower than that for the *related* language set as the task in the *diverse* language set is more complicated.

Method	M2O		O2M	
	Related	Diverse	Related	Diverse
$\text{CCL-M}_{\text{avg}}$	26.73	28.34	18.74	19.53
+ Uni.	24.59	27.13	18.29	18.21
+ Temp.	25.28	27.50	18.65	19.28
+ Prop.	27.21	28.72	18.20	18.80

Table 4: Average BLEU scores (%) on test sets by the $\text{CCL-M}_{\text{avg}}$ algorithm using our dynamic sampling method and three static sampling methods. "Uni." refers to the uniform sampling, "Temp." refers to the temperature-based sampling ($\tau = 5$), and "Prop." refers to the proportional sampling. Bold indicates the highest value.

5.2 Effects of Different Sampling Methods

We also analyze the effects of different sampling methods. Substituting our competence-aware dynamic sampling method in the $\text{CCL-M}_{\text{avg}}$ with three static sampling methods, we get the results in Table 4. Consistently, our method performs best among the sampling methods in O2M tasks, which shows the superiority of sampling by language-specific competence.

Surprisingly, we find that proportional sampling surpasses our proposed dynamic method in M2O tasks. This also indicates that more training on HRLs has a positive effect in M2O tasks, since proportional sampling would train more on the HRLs than the dynamic sampling we proposed. In addition, all three static sampling methods outperform their respective baselines in Table 1. Some of them are even better than the previous state-of-the-art sampling method, i.e., MultiDDS-S. This shows that our curriculum learning approach has a strong generability.

6 Related Work

Curriculum learning was first proposed by Bengio et al. (2009) with the idea of learning samples from easy to hard to get a better optimized model. As a general method for model improvement, curriculum learning has been widely used in a variety of machine learning fields (Gong et al., 2016; Kocmi and Bojar, 2017; Hacohen and Weinshall, 2019; Platanios et al., 2019; Narvekar et al., 2020)

There are also some previous curriculum learning researches for machine translation. For example, (Kocmi and Bojar, 2017) divide the training corpus into smaller buckets using some features such as sentence length or word frequency and then train the buckets from easy to hard according to the predefined difficulty. Platanios et al. (2019) propose competence-based curriculum learning for machine translation, which treats the model competence as a variable in training and samples the training corpus in line with the competence. In detail, they believe that competence is positively related to the training steps, and uses linear or square root functions for experiments. We bring the concept of competence and redefine it in this paper with a multilingual context. Further, we define *Self-evaluated Competence* and *HRLs-evaluated Competence* as the competence of each language pair to capture the model’s multilingual competence more accurately.

7 Conclusion

In this paper, we focus on balancing the learning competencies of different languages in multilingual machine translation and propose a competence-based curriculum learning framework for this task. The experimental results show that our approach brings significant improvements over baselines and the previous state-of-the-art balancing sampling method, MultiDDS-S. Furthermore, the ablation study on sampling methods verifies the great generalizability of our curriculum learning framework.

Acknowledgements

We would like to thank anonymous reviewers for their suggestions and comments. This work was supported by the National Key Research and Development Program of China (No. 2020YFB2103402).

References

- Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George Foster, Colin Cherry, et al. 2019. Massively multilingual neural machine translation in the wild: Findings and challenges. *arXiv preprint arXiv:1907.05019*.
- Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.
- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.
- Loïc Barrault, Magdalena Biesialska, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Matthias Huck, Eric Joanis, Tom Kocmi, Philipp Koehn, Chi-kiu Lo, Nikola Ljubešić, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Santanu Pal, Matt Post, and Marcos Zampieri. 2020. [Findings of the 2020 conference on machine translation \(WMT20\)](#). In *Proceedings of the Fifth Conference on Machine Translation*, pages 1–55, Online. Association for Computational Linguistics.
- Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48.
- Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. [Learning phrase representations using RNN encoder–decoder for statistical machine translation](#). In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.
- Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. [Unsupervised cross-lingual representation learning at scale](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.
- Raj Dabre, Chenhui Chu, and Anoop Kunchukuttan. 2020. [Multilingual neural machine translation](#). In *Proceedings of the 28th International Conference on Computational Linguistics: Tutorial Abstracts*, pages 16–21, Barcelona, Spain (Online). International Committee for Computational Linguistics.
- Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav

- Chaudhary, et al. 2020. Beyond english-centric multilingual machine translation. *arXiv preprint arXiv:2010.11125*.
- Chen Gong, Dacheng Tao, Stephen J Maybank, Wei Liu, Guoliang Kang, and Jie Yang. 2016. Multi-modal curriculum learning for semi-supervised image classification. *IEEE Transactions on Image Processing*, 25(7):3249–3260.
- Guy Hacohen and Daphna Weinshall. 2019. On the power of curriculum learning in training deep networks. In *International Conference on Machine Learning*, pages 2535–2544. PMLR.
- Sébastien Jean, Orhan Firat, and Melvin Johnson. 2019. Adaptive scheduling for multi-task learning. *arXiv preprint arXiv:1909.06434*.
- Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google’s multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.
- Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Tom Kocmi and Ondřej Bojar. 2017. Curriculum learning and minibatch bucketing in neural machine translation. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 379–386, Varna, Bulgaria. INCOMA Ltd.
- Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388–395, Barcelona, Spain. Association for Computational Linguistics.
- Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.
- Zehui Lin, Xiao Pan, Mingxuan Wang, Xipeng Qiu, Jiangtao Feng, Hao Zhou, and Lei Li. 2020. Pre-training multilingual neural machine translation by leveraging alignment information. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2649–2663, Online. Association for Computational Linguistics.
- Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.
- Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.
- Sanmit Narvekar, Bei Peng, Matteo Leonetti, Jivko Sinapov, Matthew E Taylor, and Peter Stone. 2020. Curriculum learning for reinforcement learning domains: A framework and survey. *arXiv preprint arXiv:2003.04960*.
- Graham Neubig and Junjie Hu. 2018. Rapid adaptation of neural machine translation to new languages. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 875–880, Brussels, Belgium. 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 *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.
- Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. 2018. Scaling neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 1–9, Brussels, Belgium. Association for Computational Linguistics.
- Emmanouil Antonios Platanios, Otilia Stretcu, Graham Neubig, Barnabas Poczos, and Tom Mitchell. 2019. Competence-based curriculum learning for neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1162–1172, Minneapolis, Minnesota. Association for Computational Linguistics.
- Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.
- Ye Qi, Devendra Sachan, Matthieu Felix, Sarguna Padmanabhan, and Graham Neubig. 2018. When and why are pre-trained word embeddings useful for neural machine translation? In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 529–535, New Orleans, Louisiana. Association for Computational Linguistics.

Holger Schwenk, Guillaume Wenzek, Sergey Edunov, Edouard Grave, and Armand Joulin. 2019. Cc-matrix: Mining billions of high-quality parallel sentences on the web. *arXiv preprint arXiv:1911.04944*.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. *arXiv preprint arXiv:1409.3215*.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762*.

Xinyi Wang and Graham Neubig. 2019. **Target conditioned sampling: Optimizing data selection for multilingual neural machine translation**. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5823–5828, Florence, Italy. Association for Computational Linguistics.

Xinyi Wang, Yulia Tsvetkov, and Graham Neubig. 2020. **Balancing training for multilingual neural machine translation**. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8526–8537, Online. Association for Computational Linguistics.

Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tieyan Liu. 2020. On layer normalization in the transformer architecture. In *International Conference on Machine Learning*, pages 10524–10533. PMLR.

Biao Zhang, Philip Williams, Ivan Titov, and Rico Senrich. 2020. **Improving massively multilingual neural machine translation and zero-shot translation**. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1628–1639, Online. Association for Computational Linguistics.

A Dataset

A.1 Dataset Statistics

As we can see in Table 5 and Table 6, there are 4 low resource languages (LRLs) and 4 high resource languages (HRLs) in both language sets.

Language	Train	Dev	Test
aze	5.94k	671	903
bel	4.51k	248	664
glg	10.0k	682	1007
slk	61.5k	2271	2445
tur	182k	4045	5029
rus	208k	4814	5483
por	185k	4035	4855
ces	103k	3462	3831

Table 5: Dataset statistics of the *related* language set.

Language	Train	Dev	Test
bos	5.64k	474	463
mar	9.84k	767	1090
hin	18.7k	854	1243
mkd	25.3k	640	438
ell	134k	3344	4433
bul	174k	4082	5060
fra	192k	4320	4866
kor	205k	4441	5637

Table 6: Dataset statistics of the *diverse* language set.

Language	xxx-eng	eng-xxx
aze	7.87	9.703
bel	7.843	9.051
glg	6.891	7.688
slk	5.205	5.84
tur	4.344	5.225
rus	4.577	5.011
por	3.687	4.067
ces	4.495	5.083

Table 7: Losses on development sets of bitext models in the *related* language set.

A.2 Development Losses

We use the same model and hyper-parameters as we used in subsection 4.2 and get the results in Table 7 and Table 8. We then use them to calculate the *Self-evaluated Competence*. Obviously, the losses of HRLs are lower than the losses of LRLs. At the same time, we find that the *xxx-eng* tasks is easier than the *eng-xxx* tasks. Because in *eng-xxx* tasks, the decoder shares a 64k vocabulary and would output misleading tokens.

Language	xxx-eng	eng-xxx
bos	7.499	8.687
mar	7.472	9.184
hin	6.956	7.961
mkd	5.581	6.221
ell	4.164	4.522
bul	4.004	4.278
fra	3.883	3.968
kor	4.725	5.843

Table 8: Losses on development sets of bitext models in the *diverse* language set.

Language	aze	bel	glg	slk
tur	0.50	0.12	0.24	0.30
rus	0.09	0.34	0.07	0.08
por	0.22	0.12	0.59	0.26
ces	0.24	0.11	0.27	0.68

Table 9: Language similarity of the *related* language set. Bold indicates significant similarity.

Language	bos	mar	hin	mkd
ell	0.09	0.09	0.09	0.07
bul	0.12	0.11	0.11	0.60
fra	0.18	0.08	0.09	0.07
kor	0.10	0.10	0.09	0.07

Table 10: Language similarity of the *diverse* language set. Bold indicates significant similarity.

A.3 Language Similarity

Using Equation 9, we obtain the language similarities shown in Table 9 and Table 10. As we can see, each HRL has a high-similarity LRL corresponding to it in the *related* language set. Meanwhile, languages are generally not similar in *diverse* language set, only a pair of HRL and LRL ("bul" and "mkd") have high similarity.

B Individual BLEU Scores

Here we also list the individual BLEU scores as the supplement to Table 1.

Method	SacreBLEU	aze	bel	glg	slk	tur	rus	por	ces
Bitext Models	22.38	7.07	3.77	10.79	23.09	37.33	38.26	39.75	18.96
Uniform Sampling ($\tau = \infty$)	24.81	21.52	9.48	19.99	30.46	33.22	33.70	35.15	15.03
Temperature-Based Sampling ($\tau = 5$)	26.01	23.47	10.19	21.26	31.13	34.69	34.94	36.44	16.00
Proportional Sampling ($\tau = 1$)	26.68	23.43	10.10	22.01	31.06	35.62	36.41	37.91	16.91
MultiDDS (Wang et al., 2020)	26.65	25.00	10.79	22.40	31.62	34.80	35.22	37.02	16.36
MultiDDS-S (Wang et al., 2020)	27.00	25.34	10.57	22.93	32.05	35.27	35.77	37.30	16.81
CCL-M _{max} (Ours)	28.29	25.20	11.50	23.23	33.31	37.55	38.03	39.26	18.20
CCL-M _{avg} (Ours)	28.34	25.61	11.60	23.52	33.48	37.26	38.10	39.19	17.98

Table 11: Individual BLEU scores (%) on test sets of the baselines and our methods in M2O diverse language sets. Bold indicates the highest value.

Method	SacreBLEU	aze	bel	glg	slk	tur	rus	por	ces
Bitext Models	20.37	2.59	2.69	11.62	24.88	26.34	24.12	44.53	26.18
Uniform Sampling ($\tau = \infty$)	22.63	8.81	14.80	25.22	27.32	20.16	20.95	38.69	25.11
Temperature-Based Sampling ($\tau = 5$)	24.00	10.42	15.85	27.63	28.38	21.53	21.82	40.18	26.26
Proportional Sampling ($\tau = 1$)	24.88	11.20	17.17	27.51	28.85	23.09	22.89	41.60	26.80
MultiDDS (Wang et al., 2020)	25.26	12.20	18.60	28.83	29.21	22.24	22.50	41.40	27.22
MultiDDS-S (Wang et al., 2020)	25.52	12.20	19.11	29.37	29.35	22.81	22.78	41.55	27.03
CCL-M _{max} (Ours)	26.59	12.61	19.43	29.96	30.55	24.63	23.93	43.05	28.55
CCL-M _{avg} (Ours)	26.73	12.59	19.54	30.20	30.86	24.78	24.09	43.13	28.61

Table 12: Individual BLEU scores (%) on test sets of the baselines and our methods in M2O related language sets. Bold indicates the highest value.

Method	SacreBLEU	bos	mar	hin	mkd	ell	bul	fra	kor
Bitext Models	15.73	2.22	2.54	9.82	18.40	15.02	19.57	39.42	18.86
Uniform Sampling ($\tau = \infty$)	15.54	5.76	10.51	21.08	17.83	9.94	13.59	30.33	15.35
Temperature-Based Sampling ($\tau = 5$)	16.61	6.66	11.29	21.81	18.60	11.27	14.92	32.10	16.26
Proportional Sampling ($\tau = 1$)	15.49	4.42	5.99	14.92	17.37	12.86	16.98	34.90	16.53
MultiDDS (Wang et al., 2020)	17.17	6.24	11.75	21.46	20.67	11.51	15.42	33.41	16.94
MultiDDS-S (Wang et al., 2020)	17.32	6.59	12.39	21.65	20.61	11.58	15.26	33.52	16.98
CCL-M _{max} (Ours)	18.89	7.59	13.01	23.83	21.71	13.38	16.85	35.43	19.31
CCL-M _{avg} (Ours)	18.85	7.38	13.17	23.89	21.67	13.37	16.92	35.34	19.04

Table 13: Individual BLEU scores (%) on test sets of the baselines and our methods in O2M related language sets. Bold indicates the highest value.

Method	SacreBLEU	bos	mar	hin	mkd	ell	bul	fra	kor
Bitext Models	17.83	5.00	2.68	8.17	15.44	31.35	33.88	38.02	8.06
Uniform Sampling ($\tau = \infty$)	16.86	14.12	4.69	14.52	20.10	22.87	25.02	27.64	5.95
Temperature-Based Sampling ($\tau = 5$)	17.94	14.73	4.93	15.49	20.59	24.82	26.60	29.74	6.62
Proportional Sampling ($\tau = 1$)	16.79	6.93	3.69	10.70	15.77	26.69	29.59	33.51	7.49
MultiDDS (Wang et al., 2020)	18.40	14.91	4.83	14.96	22.25	24.80	27.99	30.77	6.76
MultiDDS-S (Wang et al., 2020)	18.24	14.02	4.76	15.68	21.44	25.69	27.78	29.60	7.01
CCL-M _{max} (Ours)	19.53	14.87	4.81	15.33	22.43	28.10	29.97	33.31	7.44
CCL-M _{avg} (Ours)	19.53	14.87	4.81	15.33	22.43	28.10	29.97	33.31	7.44

Table 14: Individual BLEU scores (%) on test sets of the baselines and our methods in O2M diverse language sets. Bold indicates the highest value. The scores of CCL-M_{max} and CCL-M_{avg} are the same because they add different LRLs to the training set at the same time (the training set $\mathcal{S}_{\text{selected}}$ is adjusted every 100 steps), even with different thresholds.