Improving Unsupervised Neural Machine Translation via Training Data Self-Correction

Jinliang Lu[1,2], **Jiajun Zhang**[1,2,3]*

[1]Institute of Automation, Chinese Academy of Sciences, Beijing, China
[2]School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China
[3]Wuhan AI Research, Wuhan, China
lujinliang2019@ia.ac.cn, jjzhang@nlpr.ia.ac.cn

Abstract

Unsupervised neural machine translation (UNMT) models are trained with pseudo-parallel sentences constructed by *on-the-fly* back-translation using monolingual corpora. However, the quality of pseudo-parallel sentences cannot be guaranteed, which hinders the final performance of UNMT. This paper demonstrates that although UNMT usually generates mistakes during pseudo-parallel data construction, some of them can be corrected by the token-level translations that exist in the embedding table. Therefore, we propose a self-correction method to automatically improve the quality of pseudo-parallel sentences during training, thereby enhancing translation performance. Specifically, for a pseudo sentence pair, our self-correction method first estimates the alignment relations between tokens by treating and solving it as an optimal transport problem. Then, we measure the translation reliability for each token and detect the mis-translated ones. Finally, the mis-translated tokens are corrected with real-time computed token-by-token translations based on the embedding table, yielding a better training example. Considering that the modified examples are semantically equivalent to the original ones when UNMT converges, we introduce second-phase training to strengthen the output consistency between them, further improving the generalization capability and translation performance. Empirical results on widely used UNMT datasets demonstrate the effectiveness of our method and it significantly outperforms several strong baselines.

Keywords: unsupervised machine translation, data quality, self-correction, back-translation

1. Introduction

Unsupervised neural machine translation (UNMT) (Lample et al., 2018a; Artetxe et al., 2018b) aims to train machine translation models using monolingual corpora only. In recent years, UNMT has achieved significant progress and attracted plenty of attention. Generally speaking, UNMT utilizes cross-lingual pre-trained language models (*c*PLMs) (Conneau and Lample, 2019; Song et al., 2019) for parameter initialization to ensure the basic cross-lingual processing capabilities. Subsequently, UNMT *on-the-fly* translates the sentence from the target monolingual corpora into the source language and then translates the synthetic source sentence into the original target, basically enabling the translation capability.

Therefore, the quality of training data used in UNMT is worse than the human-labeled parallel sentences, especially at the early training stage when translation proficiency is not yet achieved. The low-quality data leads to the performance disparity between UNMT and supervised neural machine translation (SNMT) (Bahdanau et al., 2014). We use the example in Table 1 to illustrate the problem. During training, UNMT first samples a sentence $\mathbf{y}$ from monolingual corpora and translates $\mathbf{y}$ to $\hat{\mathbf{x}}$. Then, it uses $(\hat{\mathbf{x}}, \mathbf{y})$ as the pair to optimize the parameters. We find that "acord" is inaccurately translated to "happy", whose original meaning is "agree". When using $(\hat{\mathbf{x}}, \mathbf{y})$ as the pseudo sentence pair to train UNMT, such mistakes can result in the model learning incorrect alignments, consequently impairing translation performance.

src - $\mathbf{y}$	Suntem de acord cu asta .
hyp - $\hat{\mathbf{x}}$	I 'm very happy with that .
ref - $\mathbf{x}$	We do agree with this .

Table 1: An example of the pseudo parallel sentence (En-Ro) constructed by UNMT. For clarity, we provide the standard reference $\mathbf{x}$, which is not available during UNMT training.

Although UNMT usually makes token-level mistakes when constructing pseudo-parallel sentences, our investigation highlights that *c*PLMs have already acquired token-level translation proficiency (*i.e.* "acord" is already aligned with "agree" in the embedding table) after pre-training, offering a potential remedy for such mistakes. Therefore, we conduct preliminary experiments and demonstrate that carefully detecting translation mistakes and using these exported token translations to correct them in synthetic source sentences can improve the quality of constructed pseudo-parallel sentences, thereby benefiting UNMT training.

Based on the analysis, we propose a novel self-correction approach to automatically improve the

* Corresponding author

LREC-COLING 2024, pages 8942–8954

data quality during UNMT training.[1] Our method consists of three main components: *alignment matrix estimation*, *translation-mistake detection* and *continuous semantic modification*. At each training step, for the sentence pair $(\hat{\mathbf{x}}, \mathbf{y})$ constructed by UNMT, we first view the estimation of the alignment between tokens in the two sentences as an optimal transport problem (Levina and Bickel, 2001; Kusner et al., 2015) and calculate the solution (*alignment matrix estimation*). Taking the tokens of $\mathbf{y}$ as the references, we compare the aligned tokens in $\hat{\mathbf{x}}$ and the token-by-token translations in the embedding table, thus determining which token in $\mathbf{y}$ has not been accurately translated (*translation-mistake detection*). Finally, we measure the degree of mis-translated tokens in $\hat{\mathbf{x}}$ and make semantic modifications using real-time computed token-by-token translations and the alignment matrix (*continuous semantic modification*), obtaining the modified training example $(\mathbf{z}, \mathbf{y})$.

During training, our self-correction method fulfills two distinct roles. Before the model converges, self-correction improves the quality of training data. When the model achieves convergence, self-correction produces semantically equivalent yet distinct examples. Thus, we introduce the second-phase training to encourage UNMT models to produce consistent outputs (Xie et al., 2020) in line with the original and modified training examples. This phase enhances the robustness of UNMT models in dealing with semantic-equivalent yet diverse inputs. Notably, the second phase requires only approximately a hundred training steps to yield further improvements in translation quality.

We evaluate our method on widely used WMT14 En↔Fr, WMT16 En↔De, and WMT16 En↔Ro testsets using XLM and MASS models as the *c*PLMs. Experimental results demonstrate that our self-correction method significantly improves the translation performance compared with several strong baselines. Further experiments show that our method is orthogonal to other UNMT methods, allowing for their integration to yield greater improvements. Finally, we adapt our method to large language models (LLMs) (Brown et al., 2020; Touvron et al., 2023) and unveil its effectiveness with LLMs as well.

Our contributions can be summarized as follows:

- We find that UNMT models produce synthetic source sentences with mistakes and demonstrate that many mistakes can be alleviated by token-by-token translations derived from the embedding table.
- We propose a self-correction method to modify the synthetic source sentences on-the-fly, reducing the semantic mistakes and thus improving the quality of UNMT training data.
- Empirical results demonstrate the effectiveness of our method, which improves the data quality and thus enhances the translation performance on ordinary UNMT models as well as LLMs.

[1]Our code is available in https://github.com/JinliangLu96/Self-Correction-UNMT

2. Background

2.1. Unsupervised Machine Translation

The architecture of the current state-of-the-art UNMT is the same as the SNMT model. The training procedure comprises two main components: the initialization of cross-lingual PLM and *on-the-fly* back-translation.[2]

Cross-lingual PLMs are typically pre-trained with monolingual corpora collected for specific languages, which aims to encode the source sentences and target sentences into a shared space. The parameters are used to initialize the encoder and decoder in the UNMT model before training.

***On-the-Fly* Back-Translation (BT)** is the essential component of UNMT, which explicitly guarantees the model to have translation capability. First, each batch of monolingual sentences is translated into the other language by the UNMT model $\mathcal{M}$. Then, $\mathcal{M}$ applies the pseudo parallel sentences $(\mathcal{M}_{l_1 \to l_2}(\mathbf{x}), \mathbf{x})$ and $(\mathcal{M}_{l_2 \to l_1}(\mathbf{y}), \mathbf{y})$ into training. The objective function is:

$$\begin{aligned}\mathcal{L}_{bt} = &\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_{l_1}}\left[-\log P_{l_2\to l_1}(\mathbf{x}|\mathcal{M}_{l_1\to l_2}(\mathbf{x}))\right] \\ &+ \mathbb{E}_{\mathbf{y}\sim\mathcal{D}_{l_2}}\left[-\log P_{l_1\to l_2}(\mathbf{y}|\mathcal{M}_{l_2\to l_1}(\mathbf{y}))\right]\end{aligned} \tag{1}$$

Although strong UNMT models have been proposed in recent years, the uneven quality of training data, especially at the early training stage, is still a key factor that influences the final performance.

3. How to Improve Data Quality?

As we described above, the main difference between SNMT and UNMT lies in the quality of training data. Therefore, we first conduct some preliminary analyses to explore the feasible method to improve the data quality.

3.1. Token-Level Translation

Early studies about UNMT are built upon cross-lingual word embeddings, which are aligned in the same space and can form token translations

[2]Denoising Auto-Encoder (DAE) is another important component in UNMT, which can improve the model learning ability through reconstructing the original sentences from the sentences with artificial noise.

(Lample et al., 2018b; Artetxe et al., 2018a). However, current UNMT models are initialized by cross-lingual pre-trained language models, which do not involve the cross-lingual projection like previous unsupervised word translation studies. Therefore, we first demonstrate whether token-level translations still exist in the embedding table of *c*PLMs.

Suppose that language l_1 and l_2 have corresponding corpora $\mathcal{C}_{l_1}$ and $\mathcal{C}_{l_2}$. We first record the tokens that occur in $\mathcal{C}_l$ and $\mathcal{C}_{l_2}$. Then, we remove shared tokens to avoid the impact of overlapping, obtaining the language-specific vocabularies independent of each other, $\mathcal{V}_{l_1}$ and $\mathcal{V}_{l_2}$.

Next, we adopt the cross-domain similarity local scaling (CSLS) (Lample et al., 2018b) to compute the token similarity from $\mathcal{V}_{l_1}$ to $\mathcal{V}_{l_2}$. For token embeddings x and y in two languages, the CSLS score is computed as:

$$\text{CSLS}(x, y) = 2\cos(x, y) - r_K(x) - r_K(y) \quad (2)$$

where $r_K(x)$ is the average score from x to the K-nearest target neighbourhoods $\mathcal{N}(x)$.

$$r_K(x) = \frac{1}{K}\sum\nolimits_{\hat{y}_t \in \mathcal{N}(x)} \cos(x, \hat{y}_t) \quad (3)$$

For a specific token in $\mathcal{V}_{l_1}$, we find the most similar token (according to CSLS scores) in $\mathcal{V}_{l_2}$ as its translation. To measure the quality of exported token translations, we collect golden dictionaries from MUSE (Lample et al., 2018b) and compute the Hit@1 accuracy[3], which is shown in Table 2. We can find that the embedding of the XLM models has high accuracy scores of the token translation.

	De-En	Fr-En	Ro-En
nums. of pairs	4250	8347	3976
accuracy	74.87%	73.36%	75.33%

Table 2: The Hit@1 accuracy of token translations derived from XLM models.

3.2. The Feasibility of Self-Correction

As we described above, UNMT usually constructs pseudo-parallel sentences with mistakes, while accurate token translations exist in the embedding table. For example, "acord" is aligned with "agree" in the embedding table. The direct question arises: can we improve the data quality by correcting the mistakes using the exported dictionary?

To achieve this, we first choose 128 sentence pairs from WMT16 Ro→En. Then, we carefully label the alignments between tokens for each sentence pair (*i.e. Nu - not, acord - happy, cu with, asta - this*) and detect the translation mistakes (*acord - happy*). Finally, we use the exported dictionary to replace the wrongly translated tokens (*happy* → *agree*), obtaining the modified synthetic sentence.

Next, we compare the quality of the original synthetic source sentence and the modified one using sentence-level metrics: sentence-BLEU (Papineni et al., 2002), TER (Snover et al., 2006), ChrF (Popović, 2015) and COMET (Rei et al., 2020). As shown in Figure 1, we can find that at the early stage, the modification would improve the quality of synthetic source sentences. For example, the improved sentences account for 89% according to ChrF. However, for synthetic source sentences constructed in the late stage, the number of mistranslated tokens decreases and thus the modification brings improvements to about 20% sentences.

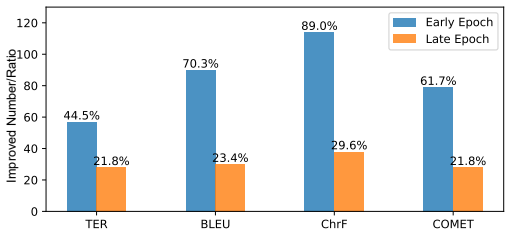

Figure 1: The number (ratio) of improved sentences with different metrics during training.

The above preliminary experiments demonstrate that the quality of pseudo-parallel sentences can be improved by exported token translations with careful mistake detection. However, how to design the algorithm that automatically completes such operations to construct high-quality synthetic source sentences becomes the key issue, which will be described in the next section.

4. Our Method - Self-Correction

Our approach entails a two-phase training procedure. In the first stage, the self-correction method effectively enhances the data quality. We employ the modified examples to train the UNMT model. After the model achieves convergence, the self-correction method assumes the other role of generating semantically equivalent yet distinct samples. Consequently, we introduce a second stage to bolster output consistency when the model encounters semantically equivalent yet diverse inputs, further enhancing the capability of UNMT models.

[3]Considering that exported token translations have noise, we remove the tokens whose max CSLS score less than 0.10. Then, we compute the translation accuracy of the tokens covered by MUSE dictionary.

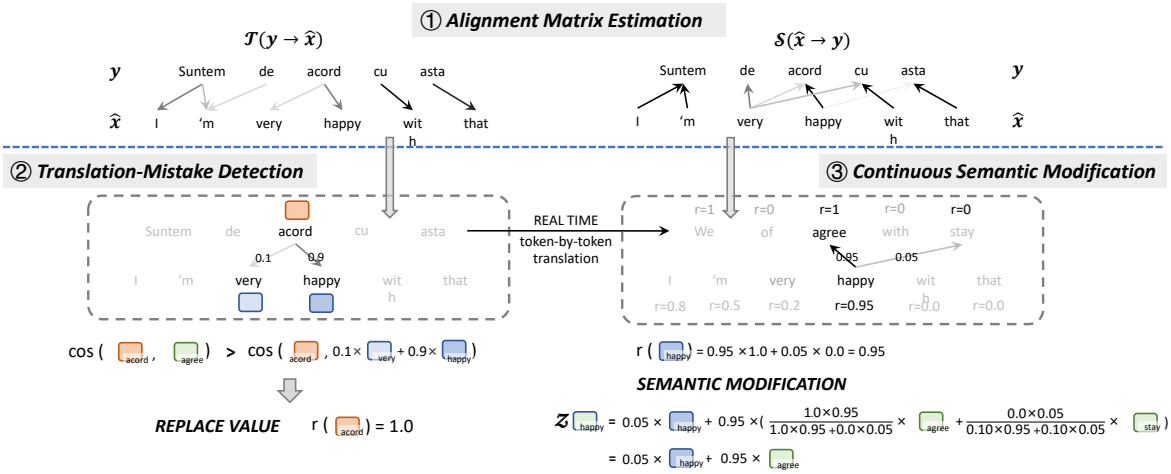

Figure 2: Illustration of our self-correction method. After obtaining the alignment matrix ① $\boldsymbol{S}$ and $\boldsymbol{T}$, we use them for translation-mistake detection ② and continuous semantic modification ③. Specifically, in ②, "acord" has larger cosine scores with "agree" than weighted aligned tokens "very" and "happy", thus should be viewed as the mis-translated token. In ③, "happy" is potentially aligned with "acord" ("agree"), we first obtain the replacement values for it using $\boldsymbol{S}$ and then interpolate the $\texttt{Emb}$(happy) with the embeddings of token-by-token translation, $\texttt{Emb}$(agree).

4.1. Stage 1: UNMT with Modified Pseudo Parallel Sentences

As we discussed in §3.2, UNMT usually generates translation mistakes that can be corrected with token-by-token translations. As shown in Figure 2, our method consists of *alignment matrix estimation*, *translation-mistake detection*, and *continuous semantic modification*.

Specifically, at each training step, we sample a sentence $\mathbf{y}$ from the monolingual corpora $\mathcal{D}_{l_2}$, which is translated into $\hat{\mathbf{x}}$ by UNMT model $\mathcal{M}$ using inference mode. Then, we have a pseudo sentence-pair $(\hat{\mathbf{x}}, \mathbf{y})$, $\hat{\mathbf{x}} = \mathcal{M}(\mathbf{y})$. The word embedding sequences of $\hat{\mathbf{x}}$ and $\mathbf{y}$ can be denoted as $\hat{\boldsymbol{x}} = [\hat{\boldsymbol{x}}_1, \hat{\boldsymbol{x}}_2, \cdots, \hat{\boldsymbol{x}}_n]$ and $\boldsymbol{y} = [\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_m]$. Generally, the quality of $(\hat{\mathbf{x}}, \mathbf{y})$ is determined by $\hat{\mathbf{x}}$.

Alignment Matrix Estimation To effectively detect the translation mistakes in $\hat{\mathbf{x}}$, we need to figure out the token alignments between $\hat{\mathbf{x}}$ and $\mathbf{y}$ and use $\mathbf{y}$ to check the quality of $\hat{\mathbf{x}}$. Specifically, we treat the alignment matrix estimation as the optimal transport problem, measuring the semantic distance, aligning semantically similar tokens, and obtaining the amount of flow traveling between them.

$$\begin{aligned} &\min_{\boldsymbol{F}\geq 0} \sum_{i,j=1} \boldsymbol{F}_{i,j} d(i,j) \\ &\text{subject to}: \sum_j \boldsymbol{F}_{i,j} = w_i, \forall i \in \{1,\cdots,n\} \\ &\qquad\quad \sum_i \boldsymbol{F}_{i,j} = w'_j, \forall j \in \{1,\cdots,m\} \end{aligned} \tag{4}$$

where $\boldsymbol{F}$ is the transportation matrix and $d(i,j) = \|\hat{\boldsymbol{x}}_i - \boldsymbol{y}_j\|_2$ is the cost function. Considering that important tokens usually have larger norms (Schakel and Wilson, 2015; Yokoi et al., 2020), we use the norm of corresponding tokens as the weights w_i, w'_j to represent the importance. Because the sum of the rows and columns in $\boldsymbol{F}$ are the corresponding norms instead of 1.0, we separately normalize it by rows and columns as the alignment matrices $\boldsymbol{S} \in \mathbb{R}^{n\times m}$ $(\hat{\mathbf{x}} \to \mathbf{y})$ and $\boldsymbol{T} \in \mathbb{R}^{m\times n}$ $(\mathbf{y} \to \hat{\mathbf{x}})$:

$$\boldsymbol{S}_{i,j} = \frac{e^{\boldsymbol{F}^*_{i,j}}}{\sum_j e^{\boldsymbol{F}^*_{i,j}}}, \quad \boldsymbol{T}_{j,i} = \frac{e^{\boldsymbol{F}^*_{j,i}}}{\sum_i e^{\boldsymbol{F}^*_{j,i}}} \tag{5}$$

$$\boldsymbol{F}^*_{i,j} = \begin{cases} -\inf, & \text{if } \boldsymbol{F}_{i,j} = 0, \\ \boldsymbol{F}_{i,j}, & \text{otherwise} \end{cases} \tag{6}$$

Translation-Mistake Detection After obtaining the alignment matrix $\boldsymbol{T}$, we can roughly estimate the tokens in $\hat{\mathbf{x}}$ which are aligned to tokens in $\mathbf{y}$. For a specific token $\mathbf{y}_j$ with embedding $\boldsymbol{y}_j$, it would have multiple aligned tokens in $\hat{\mathbf{x}}$. Therefore, we first compute cosine similarity from $\mathbf{y}_j$ to each token in $\hat{\mathbf{x}}$, obtaining cosine vector $\boldsymbol{m} \in \mathbb{R}^n$. Then, we use the j-th row in alignment matrix $\boldsymbol{T}$ to compute the aligned cosine scores c_j for $\mathbf{y}_j$:

$$c_j = \boldsymbol{m} \cdot \boldsymbol{T}_j \tag{7}$$

Considering that the token embeddings are further optimized during training, we do not directly use the exported dictionary (§3.1) but calculate the cosine similarity from $\mathbf{y}_j$ to the arbitrary token in opposite vocabulary $\mathcal{V}_{l_1}$ to obtain the token-by-token

translations in real-time:

$$c_j^{'} = \max_{\mathbf{x}_k^{'} \in \mathcal{V}_{l_1}} \cos(\boldsymbol{y}_j, \boldsymbol{x}_k^{'})$$
$$\mathbf{x}_j^{'} = \operatorname{argmax}_{\mathbf{x}_k^{'} \in \mathcal{V}_{l_1}} \cos(\boldsymbol{y}_j, \boldsymbol{x}_k^{'}) \quad (8)$$

We define $\Delta c_j = c_j^{'} - c_j$ as the token-level translation reliability. For j-th token, if $\Delta c_j > \sigma$, we believe that the token generated by UNMT is not better than the token-by-token translation, thus should be replaced and we set the replacement value $r_{y_j} = 1$. Otherwise, $r_{y_j} = 0$. All the replacement values form the replacement vector $\boldsymbol{r}_y = [r_{y_1}, r_{y_2}, \cdots, r_{y_m}] \in \mathbb{R}^m$. σ is the hyper-parameter which should be varied during training:

$$\sigma = \sigma_{\text{min}} + (st/st_{\text{max}}) \times (\sigma_{\text{max}} - \sigma_{\text{min}}) \quad (9)$$

where st means training step. we set $\sigma_{\text{min}} = 0.1$ and $\sigma_{\text{max}} = 0.5$ in our experiments.

Continuous Semantic Modification After detecting the mis-translated tokens, we adopt the token-by-token translations $\mathbf{x}^{'}$ to modify $\hat{\mathbf{x}}$.

For the k-th token embedding in $\hat{\boldsymbol{x}}_k$, we replace it with continuous representation $\boldsymbol{z}_k$. Considering that k-th token can be aligned to multiple target tokens, we use the k-th row in the alignment matrix $\boldsymbol{S}$ to obtain the replacement vector $\boldsymbol{z}_k$:

$$\boldsymbol{z}_k = (1 - \boldsymbol{S}_k \cdot \boldsymbol{r}_y)\hat{\boldsymbol{x}}_k + (\boldsymbol{S}_k \otimes \boldsymbol{r}_y) \cdot \boldsymbol{x}^{'} \quad (10)$$

where $\otimes$ is the element-wise product.

Finally, we feed the modified embedding sequence as well as the original pseudo-parallel sentences into the model. The loss function is:

$$\mathcal{L}_{mbt} = \mathbb{E}_{\mathbf{y} \sim \mathcal{D}_{l_2}}[-\log P(\mathbf{y}|\hat{\mathbf{x}}) - \log P(\mathbf{y}|\mathbf{z})] \quad (11)$$

4.2. Stage 2: Consistency Training

During training, the translation performance of the UNMT model steadily increases. When UNMT converges, the other role of self-correction comes to play - constructing semantically equivalent but distinct examples (as discussed in §6.1). In this scenario, facilitating UNMT to adapt to diverse inputs is beneficial. Therefore, we use the consistency loss (Xie et al., 2020; Wu et al., 2021) to encourage the UNMT model to have similar outputs. Specifically, the bi-directional Kullback-Leibler divergence is adopted as the consistency constraint:

$$\mathcal{L}_{cons} = \mathbb{E}_{\mathbf{y} \sim \mathcal{D}_{l_2}}[\mathcal{L}_{KL}(P(\mathbf{y}|\hat{\mathbf{x}}) \| P(\mathbf{y}|\mathbf{z})) + \mathcal{L}_{KL}(P(\mathbf{y}|\boldsymbol{z}) \| P(\mathbf{y}|\hat{\mathbf{x}}))] \quad (12)$$

The loss function in stage 2 can be written as:

$$\mathcal{L}_{cbt} = \mathcal{L}_{mbt} + \lambda \mathcal{L}_{cons} \quad (13)$$

where λ is the hyper-parameter during training.

5. Experiments

5.1. Main Experiments

5.1.1. Experimental Settings

Datasets For training, we collect monolingual corpora of En (179.9M), De (50.0M), Fr (65.4M), and Ro (2.8M) from WMT News Crawl. For evaluation, we respectively adopt WMT *newsdev*2014 / *newstest*2014, *newsdev*2016 / *newstest*2016, *newsdev*2016 / *newstest*2016 as development/test sets for En-Fr, En-De and En-Ro.[4]

Model Following previous settings, we evaluate the UNMT model fine-tuned on XLM and MASS pre-trained model, which has a 6-layer encoder and a 6-layer decoder with the hidden dimension 1024. Specifically, the XLM models are released by Conneau and Lample (2019), and MASS models are released by Song et al. (2019).

Training Settings During training, we use Adam optimizer with an initial learning rate 1e-4, β_1=0.9, and β_2=0.98. To be comparable with previous studies that usually adopt 8 NVIDIA V100 GPUs with 2000 tokens per GPU (16*k* tokens), we use 4 NVIDIA A100-40G GPUs with a batch size of 4000 tokens per GPU for UNMT training without gradient accumulation (16*k* tokens). For a fair comparison with previous studies, we report the BLEU scores computed by `multi-bleu.perl` scripts.

Method Comparison We compare our method with several existing approaches.

- *XLM* (Conneau and Lample, 2019), *SemFace* (Ren et al., 2021), and *MASS* (Song et al., 2019) are the strong baselines that adopt vanilla on-the-fly BT to train UNMT.
- *Adversarial-Training* (AT) (Sun et al., 2020) improves the robustness of UNMT models by inserting gradient-based noise.
- *Self-Training-Offline* (ST-Offline) (Sun et al., 2021) adopts offline forward translation to construct pseudo-parallel sentences to alleviate the data imbalance problem.
- *Self-Training-Online* (ST-Online) (He et al., 2022) employs the online forward translation to construct pseudo sentence pairs to relieve the translaionese problem in UNMT.
- *Quality-Filtering* (QF) (Lu and Zhang, 2021) utilizes self-paced learning to help UNMT concentrate on high-quality examples.

[4]Appendix shows more details about the datasets.

Methods	En↔Fr		En↔De		En↔Ro		Avg.
	En→Fr	Fr→En	En→De	De→En	En→Ro	Ro→En	
XLM (Conneau and Lample, 2019)	33.4	33.3	26.4	34.3	33.3	31.8	32.1
SemFace (Ren et al., 2021)	34.3	35.0	28.8	35.2	34.5	32.9	33.5
MASS (Song et al., 2019)	37.5	34.9	28.3	35.2	35.2	33.1	34.0
XLM (*our re-impl.*)	37.3/36.0	34.7/34.4	27.1/27.3	33.9/33.7	34.5/35.0	32.7/32.5	33.4/33.2
+ AT (Sun et al., 2020)	37.8	34.9	27.6	34.4	-	-	-
+ ST-Offline (Sun et al., 2021)	35.6	34.9	-	-	36.0	33.6	-
+ QF (Lu and Zhang, 2021)	37.4/36.2	34.9/34.6	27.4/27.3	34.4/34.6	35.3/35.4	33.4/33.1	33.8/33.5
+ ST-Online (He et al., 2022)	37.4/36.4	34.8/34.3	**28.1/28.3**	34.6/34.5	35.5/35.7	33.5/33.1	34.0/33.7
+ Self-Correction (SC) (*ours*)	**38.0/36.9**†	**35.0/34.7**†	**28.1/28.3**†	**34.9/34.8**†	**36.2/36.2**†	**34.5/34.2**†	**34.5/34.2**
w/o Consistency Training	37.9/36.9†	34.8/34.6	27.4/27.6*	34.7/34.5†	35.7/35.4†	33.8/33.5†	34.1/33.8
MASS (*our re-impl.*)	37.1/36.1	34.7/33.5	27.4/27.3	34.9/34.8	34.9/35.0	33.2/32.4	33.7/33.2
+ QF (Lu and Zhang, 2021)	37.3/36.1	34.9/34.6	28.1/27.8	35.2/35.1	35.8/35.9	33.9/33.6	34.2/33.8
+ ST-Online (He et al., 2022)	37.7/36.6	**35.1/34.9**	28.4/28.4	35.4/35.3	35.8/36.0	33.6/33.6	34.3/34.1
+ Self-Correction (SC) (*ours*)	**37.9/36.7**†	**35.1/34.9***	**28.8/29.0**†	**36.2/35.7**†	**36.4/36.3**†	**34.2/33.8**†	**34.8/34.4**
w/o Consistency Training	37.6/36.4*	34.8/34.7*	27.9/28.1†	35.6/35.5†	35.7/36.0†	33.8/33.3†	34.2/34.0

Table 3: Unsupervised translation performance on WMT14 En-Fr, WMT16 En-De, WMT16 En-Ro. For previous studies with open-source code, we re-implement their method using our settings and report *BLEU*/*detokenzied sacreBLEU*. For others, we report *BLEU* scores provided in their paper. * and † separately indicate the gains are statistically significant than baselines with p<0.05 and p<0.001.

5.1.2. Experimental Results

Main Results The experimental results are presented in Table 3. We re-implement XLM and MASS as the strong baselines, which achieve comparable or better results compared with the reported BLEU scores in their papers. By comparison, our method outperforms the baselines (XLM/MASS) by a large margin, separately obtaining 1.1 BELU improvements. For specific language directions, our method would obtain significant performance. For example, our method separately obtains 36.2/34.5 BLEU on En→Ro, Ro→En (+1.7/+1.8 BLEU) when using XLM as the *c*PLM. Furthermore, compared with other methods, such as *Quality-Filtering* and *Self-Training*, our method also has better translation performance.

Finally, we show the results when removing the second stage (consistency training) in Table 3. We can find that the improvements would decrease without consistency training (+1.1 BLEU → + 0.7 BLEU for XLM, + 1.1 BLEU → + 0.4 BLEU for MASS). It demonstrates that requesting UNMT models to produce consistent outputs would benefit the translation performance.

Combination with Previous Methods Finally, we combine our method with *Self-Training* (ST) and *Quality Filtering* (QF). The results are shown in Table 4. We can find that combining our method with either *Self-Training* or *Quality Filtering* would further boost the translation performance. For the combination method SC+ST, the improvements are significant (28.1 → 29.3, + 1.2 BLEU), demonstrating that our method is orthogonal to *Self-Training*. By contrast, the improvements of SC+QF are smaller. *Quality Filtering* also focuses on the quality of pseudo-parallel sentences, which assigns higher weights for the high-quality tokens or sentences. It makes the two methods not completely orthogonal, leading to smaller improvements.

Method	En→De	De→En
XLM	27.1 / 27.3	33.9 / 33.7
+ QF	27.4 / 27.3	34.4 / 34.6
+ ST-Online	28.1 / 28.3	34.6 / 34.5
+ SC (*ours*)	28.1 / 28.3†	34.9 / 34.8†
w/o Consistency	27.4 / 27.6†	34.7 / 34.5†
+ SC + QF	**28.4** / **28.7**†	**35.1** / **35.0**†
w/o Consistency	27.7 / 27.9†	34.9 / 34.8†
+ SC + ST-Online	**29.3** / **29.4**†	**35.4** / **35.3**†
w/o Consistency	28.8 / 29.0†	35.2 / 35.1†

Table 4: Unsupervised translation performance (tokenized BLEU/detokenized sacreBLEU) of combined methods on WMT16 En-De.

5.2. Experiments on LLMs

UNMT models usually exhibit poor performance in distant languages while large language models (LLMs) well perform in multiple languages due to the massive amounts of training data. Therefore, we evaluate the UNMT performance of LLMs on En→Zh.[5] Specifically, we first construct pseudo-

[5]It is worth noting that a limited amount of parallel data may inadvertently exist in LLM training data. Nev-

parallel sentences as training data. Subsequently, we employ instruction tuning as the baseline and compare it with our self-correction approach, which harnesses modified pseudo-source sentences during the instruction tuning process.

5.2.1. Experimental Settings

Dataset For LLMs, *on-th-fly* back-translation with a great number of monolingual corpora is not practical, as the inference speed is very slow. Therefore, we use in-context learning (ICL) (Brown et al., 2020) to translate 50k Chinese sentences (randomly sampled from 13.8M WMT News Crawl Data) into English offline. The ICL template is:

$$\text{Translate: } [l_2]\ \mathbf{y}\ [l_1]\ \mathbf{x}$$

Then, we obtain the pseudo-parallel sentence $(\hat{\mathbf{x}}, \mathbf{y})$. After filtering noisy pairs, we keep about 40k training examples for instruction tuning and adopt WMT En→Zh *newsdev2017/newstest2017* as the development/test set. Considering that pseudo-parallel sentences are constructed offline, the data quality remains consistent throughout the training process. Therefore, σ is consistently set as 0.30.

Model We choose XGLM (from 1.7B to 7.5B) (Lin et al., 2022) models for experiments, which are trained using multiple monolingual corpora without explicitly involving cross-lingual corpora in training. The baseline model is trained based on XLM.

Training Settings We use the mini-batch with 32k tokens to train models for 5 epochs. During inference, we use the 8-shot ICL for vanilla and instruction-tuned models. For evaluation, we use the `TokenizeChinese.py`[6] to cut the Chinese sentences into characters and report `sacreBLEU` (Post, 2018) and `COMET` (Rei et al., 2020) scores.

5.2.2. Experimental Results

The experimental results shown in Table 5 demonstrate the advantages of LLMs in the translation of distant language pairs compared to conventional UNMT. First, LLMs outperform UNMT_{XLM} in terms of translation quality measured by the COMET score, which is more relevant to human judgments than the BLEU metric. Secondly, it is worth emphasizing that UNMT_{XLM} leverages millions of monolingual sentences during training, whereas LLMs utilize just 40k pseudo-sentence pairs. These results underscore the untapped potential of LLMs in translating distant languages.

Furthermore, LLMs after instruction tuning exhibit higher translation performance when using the same ICL templates for inference. Our proposed self-correction methodology proves effective for both UNMT_{XLM} and LLMs, resulting in enhancements across different model sizes. To be specific, our self-correction increases the COMET score from 82.68 to 83.28 for $\text{XGLM}_{7.5\text{B}}$.

6. Analysis

6.1. Effectiveness of Self-Correction

We first evaluate the effectiveness of our method. During training, our self-correction method serves a dual purpose. As illustrated in Figure 3, we track the curve of training loss and back-translated metrics (*i.e* Δ BLEU = BLEU($\mathbf{y}, \mathbf{z} \rightarrow \mathbf{y}$) - BLEU($\mathbf{y}, \hat{\mathbf{x}} \rightarrow \mathbf{y}$)) as indirect indicators of the quality of the modified examples.[7] We observe that in the early stage of training, the loss and back-translated BLEU/ChrF/COMET scores of modified examples significantly surpass those of the original data, affirming that self-correction indeed enhances data quality. As the model approaches convergence, ΔBLEU/ΔChrF/ΔCOMET and the loss gap gradually narrow, signifying that self-correction automatically adjusts the extent of modification. It then assumes a new role, generating semantically equivalent yet diverse examples when UNMT becomes proficient at constructing good training examples.

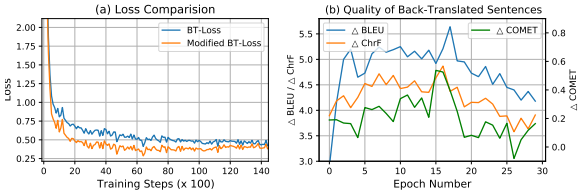

Figure 3: Comparison between original examples and modified ones: (a) training loss; (b) validation back-translated metric scores.

On the other hand, our self-correction method feeds the modified pseudo-sentence pair as well as the original example into the UNMT model, which necessitates double-forward computation at each training step. Therefore, we conducted the experiment wherein the original training example underwent forward computation twice. As shown in Table 6, 2× *Forward* also obtains slight improvements but cannot surpass *Self-Correction*, demonstrating the effectiveness of modified training examples.

ertheless, certain LLMs, such as GPT-3 and XGLM, do not explicitly perform translation tasks. We roughly think these models can be used to investigate UNMT.

[6] https://www.statmt.org/wmt17/tokenizeChinese.py

[7] This indirect assessment is necessary due to the continuous nature of our modifications, which complicates direct quality measurement.

Model	Num. Params	Vanilla ICL	Instruction SFT	+ Self-Correction
UNMT_{XLM}	336M	-	18.41 / 67.46	**21.07**† / **70.35**†
$\text{XGLM}_{\text{1.7B}}$	1.7B	14.19 / 73.90	15.84 / 76.15	**17.16**† / **77.14***
$\text{XGLM}_{\text{2.9B}}$	2.9B	18.31 / 78.61	19.97 / 80.60	**21.08**† / **80.97**
$\text{XGLM}_{\text{7.5B}}$	7.5B	22.27 / 81.86	23.47 / 82.68	**24.00*** / **83.28***

Table 5: UNMT performance (sacreBLEU/COMET) on WMT17 En→Zh. For UNMT_{XLM}, we report the baseline result in *Instruction SFT*. The statistical significance is measured with results of *Instruction SFT*.

	De→En	En→Ro
XLM	33.9	34.6
2× Forward	34.2	35.2
Self-Correction	**34.7**	**35.7**

Table 6: The comparison of translation performance when using double forward computation or our proposed self-correction without stage 2.

6.2. Training Efficiency

We also conducted an efficiency comparison of various methods. As shown in Table 7, almost all the methods enhance the performance while simultaneously impacting training speed. Specifically, *MASS+ST* needs double time forward computation, resulting in a 0.63x fraction, which is slower than *MASS+QF*. *CBD* (Nguyen et al., 2021) is a data augmentation method that employs two additional models to generate diverse pseudo-parallel sentences. While *CBD* achieves significant improvements, it exhibits the slowest training speed (0.35x). Compared with the aforementioned methods, our *Self-Correction (SC)* strikes a balance between the performance and the training speed.

	De→En	Ro→En	Training Speed
MASS	34.9	33.2	4636 (1.00x)
+ QF	35.2	33.9	4480 (0.97x)
+ ST	35.4	33.6	2924 (0.63x)
+ CBD	**36.3**	33.8	1033 (0.35x)
+ SC (*ours*)	36.2	**34.2**	2976 (0.64x)

Table 7: The comparison of different methods on training efficiency. Average speed (tokens/s) is measured on NVIDIA A100-40G and numbers in brackets are the fractions compared with MASS.

We further compare the training cost of stage 1 and stage 2 in our method. As shown in Table 8, stage 1 typically requires approximately 40∼50k training steps to achieve convergence, whereas stage 2 exhibits convergence with just a few hundred steps. The results demonstrate the efficiency of consistency training, leading to a noteworthy enhancement in translation performance while incurring only 0.2∼0.3% of the training cost.

	Stage 1	Stage 2	Fraction
En-Fr	49625	80	0.16%
En-De	47770	120	0.25%
En-Ro	41445	80	0.19%

Table 8: The comparison of training steps of Stage 1 and Stage 2. Fraction = $\text{STEP}_{\text{stage-1}}$ / $\text{STEP}_{\text{stage-2}}$.

6.3. Ablation Study on Hyper-Parameters

In stage 2, we use two hyper-parameters: σ, which governs the extent of modification, and λ, which balances the influence of the consistency loss. To investigate their impact on the translation performance, we set σ in {0.1, 0.3, 0.5, 0.7, 0.9}, λ in {5.0, 7.5, 10.0, 12.5, 15.0} and conduct experiments on WMT16 En→Ro. The results are shown in Figure 4. We find that setting $\sigma = 0.50$ and $\lambda = 10.0$ obtains the best translation quality. Notably, setting $\sigma = 0.10$ results in poor performance. This may be attributed to the fact that a smaller σ leads to substantial modifications of the pseudo-parallel sentences, thereby perturbing the original semantics. In contrast, larger σ only results in tiny alterations to the original sample, thereby diminishing the extent of improvement as well.

7. Related Work

UNMT Unsupervised neural machine translation is proposed in (Lample et al., 2018a) and (Artetxe et al., 2018b), which aims to build translation models using monolingual corpora only. Early studies train UNMT based on the cross-lingual word embedding (Artetxe et al., 2018a; Lample et al., 2018b), while advanced UNMT models are fine-tuned on cross-lingual pre-trained models (*c*PLMs). (Conneau and Lample, 2019; Song et al., 2019).

Recently, various methods have been proposed to enhance UNMT, which can be categorized into pre-training-based (PT) and fine-tuning-based (FT) methods. Typically, PT and FT are orthogonal to each other. PT mainly improves the cross-lingual capability of *c*PLMs to provide better parameters for the UNMT initialization (Ren et al., 2019, 2021; Ai and Fang, 2022, 2023; Lu et al., 2023). FT focuses on the training strategy of UNMT. Specifically, Sun et al. (2020) improves the robustness

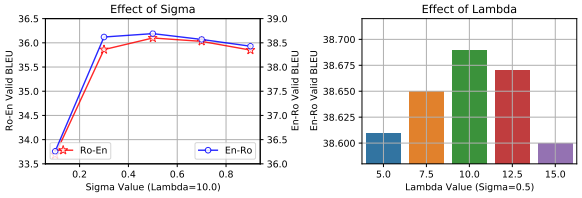

Figure 4: Ablation study of hyper-parameters - σ and λ in training stage-2.

of UNMT models with adversarial denoising training. Keung et al. (2020); Tran et al. (2020) propose unsupervised bi-text mining methods from monolingual corpora to improve UNMT. Lu and Zhang (2021) adopt curriculum learning to help the UNMT model concentrate on high-quality training examples. Nguyen et al. (2021) adopt additional UNMT models to construct diverse training examples for the current one. Sun et al. (2021) and He et al. (2022) utilize forward translation (Zhang and Zong, 2016) to construct pseudo parallel sentences to relieve the data imbalance and translationese issues.

BT Data Filtering in NMT Back-translation (Sennrich et al., 2016) is an important technique in NMT. Previous studies (Hoang et al., 2018; Burlot and Yvon, 2018) demonstrate that the quality of BT data matters for translation performance. Imankulova et al. (2017); Junczys-Dowmunt (2018); Khatri and Bhattacharyya (2020); Xu et al. (2022) utilize sentence-level metrics as the weight to filter noisy synthetic sentence pairs. Ramnath et al. (2021) provide hints about the quality of BT data, enabling the NMT model to learn from noisy examples.

Our work belongs to the FT-based method. In contrast to most previous studies, we highlight the potential of utilizing token translations from the embedding table to improve the quality of BT data for UNMT, thereby enhancing translation performance.

8. Conclusion

In this work, we investigate the data quality of UNMT and find that token-by-token translations that exist in the embedding table would alleviate many mistakes in the constructed pseudo-parallel sentences. Based on the observation, we propose a self-correction method to improve the data quality during UNMT training. Experimental results demonstrate that our method significantly outperforms the strong baselines. Further experiments on the method combination and large language models also show the effectiveness of our method, obtaining better translation performance.

Acknowledgements

This work is supported by the National Key R&D Program of China 2022ZD0160602 and the Natural Science Foundation of China 62122088.

9. Bibliographical References

Xi Ai and Bin Fang. 2022. Vocabulary-informed language encoding. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4883–4891, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Xi Ai and Bin Fang. 2023. On-the-fly cross-lingual masking for multilingual pre-training. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 855–876, Toronto, Canada. Association for Computational Linguistics.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018a. A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 789–798, Melbourne, Australia. Association for Computational Linguistics.

Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. 2018b. Unsupervised neural machine translation. In *International Conference on Learning Representations*.

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

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Franck Burlot and François Yvon. 2018. Using monolingual data in neural machine translation: a systematic study. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 144–155, Brussels, Belgium. Association for Computational Linguistics.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Zhiwei He, Xing Wang, Rui Wang, Shuming Shi, and Zhaopeng Tu. 2022. Bridging the data gap between training and inference for unsupervised neural machine translation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6611–6623, Dublin, Ireland. Association for Computational Linguistics.

Vu Cong Duy Hoang, Philipp Koehn, Gholamreza Haffari, and Trevor Cohn. 2018. Iterative back-translation for neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 18–24, Melbourne, Australia. Association for Computational Linguistics.

Aizhan Imankulova, Takayuki Sato, and Mamoru Komachi. 2017. Improving low-resource neural machine translation with filtered pseudo-parallel corpus. In *Proceedings of the 4th Workshop on Asian Translation (WAT2017)*, pages 70–78, Taipei, Taiwan. Asian Federation of Natural Language Processing.

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

Phillip Keung, Julian Salazar, Yichao Lu, and Noah A. Smith. 2020. Unsupervised bitext mining and translation via self-trained contextual embeddings. *Transactions of the Association for Computational Linguistics*, 8:828–841.

Jyotsana Khatri and Pushpak Bhattacharyya. 2020. Filtering back-translated data in unsupervised neural machine translation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4334–4339, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. 2015. From word embeddings to document distances. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 957–966, Lille, France. PMLR.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018a. Unsupervised machine translation using monolingual corpora only. In *International Conference on Learning Representations*.

Guillaume Lample, Alexis Conneau, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018b. Word translation without parallel data. In *International Conference on Learning Representations*.

Elizaveta Levina and Peter Bickel. 2001. The earth mover's distance is the mallows distance: Some insights from statistics. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, volume 2, pages 251–256. IEEE.

Xi Victoria Lin, Todor Mihaylov, Mikel Artetxe, Tianlu Wang, Shuohui Chen, Daniel Simig, Myle Ott, Naman Goyal, Shruti Bhosale, Jingfei Du, Ramakanth Pasunuru, Sam Shleifer, Punit Singh Koura, Vishrav Chaudhary, Brian O'Horo, Jeff Wang, Luke Zettlemoyer, Zornitsa Kozareva, Mona Diab, Veselin Stoyanov, and Xian Li. 2022. Few-shot learning with multilingual generative language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9019–9052, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jinliang Lu, Yu Lu, and Jiajun Zhang. 2023. Take a closer look at multilinguality! improve multilingual pre-training using monolingual corpora only. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 2891–2907, Singapore. Association for Computational Linguistics.

Jinliang Lu and Jiajun Zhang. 2021. Exploiting curriculum learning in unsupervised neural machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 924–934, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Xuan-Phi Nguyen, Shafiq Joty, Thanh-Tung Nguyen, Kui Wu, and Ai Ti Aw. 2021. Cross-model back-translated distillation for unsupervised machine translation. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8073–8083. PMLR.

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*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. 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.

Sahana Ramnath, Melvin Johnson, Abhirut Gupta, and Aravindan Raghuveer. 2021. HintedBT: Augmenting Back-Translation with quality and transliteration hints. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1717–1733, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Shuo Ren, Yu Wu, Shujie Liu, Ming Zhou, and Shuai Ma. 2019. Explicit cross-lingual pre-training for unsupervised machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 770–779, Hong Kong, China. Association for Computational Linguistics.

Shuo Ren, Long Zhou, Shujie Liu, Furu Wei, Ming Zhou, and Shuai Ma. 2021. SemFace: Pre-training encoder and decoder with a semantic interface for neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4518–4527, Online. Association for Computational Linguistics.

Adriaan MJ Schakel and Benjamin J Wilson. 2015. Measuring word significance using distributed representations of words. *arXiv preprint arXiv:1508.02297*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Matthew Snover, Bonnie Dorr, Rich Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers*, pages 223–231, Cambridge, Massachusetts, USA. Association for Machine Translation in the Americas.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. MASS: Masked sequence to sequence pre-training for language generation. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5926–5936. PMLR.

Haipeng Sun, Rui Wang, Kehai Chen, Xugang Lu, Masao Utiyama, Eiichiro Sumita, and Tiejun Zhao. 2020. Robust unsupervised neural machine translation with adversarial denoising training. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4239–4250, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Haipeng Sun, Rui Wang, Kehai Chen, Masao Utiyama, Eiichiro Sumita, and Tiejun Zhao. 2021. Self-training for unsupervised neural machine translation in unbalanced training data scenarios. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3975–3981, Online. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Chau Tran, Yuqing Tang, Xian Li, and Jiatao Gu. 2020. Cross-lingual retrieval for iterative self-supervised training. In *Advances in Neural Information Processing Systems*, volume 33, pages 2207–2219. Curran Associates, Inc.

Lijun Wu, Juntao Li, Yue Wang, Qi Meng, Tao Qin, Wei Chen, Min Zhang, Tie-Yan Liu, et al. 2021. R-drop: Regularized dropout for neural networks. *Advances in Neural Information Processing Systems*, 34:10890–10905.

Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. 2020. Unsupervised data augmentation for consistency training. *Advances in neural information processing systems*, 33:6256–6268.

Jiahao Xu, Yubin Ruan, Wei Bi, Guoping Huang, Shuming Shi, Lihui Chen, and Lemao Liu. 2022. On synthetic data for back translation. In *Proceedings of the 2022 Conference of the North*

American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 419–430, Seattle, United States. Association for Computational Linguistics.

Sho Yokoi, Ryo Takahashi, Reina Akama, Jun Suzuki, and Kentaro Inui. 2020. Word rotator's distance. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2944–2960, Online. Association for Computational Linguistics.

Jiajun Zhang and Chengqing Zong. 2016. Exploiting source-side monolingual data in neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545, Austin, Texas. Association for Computational Linguistics.

Appendix

Experiments Details

Datasets For a fair comparison with previous studies, we use the data (En, De, Fr, Ro) provided in (Song et al., 2019), as shown in Table 9. The Chinese monolingual sentences are downloaded from WMT News Crawl and we use all the sentences for XLM En-Zh experiments. For LLM experiments, we sample 50k monolingual sentences for training efficiency.

Data	Lan.	# Sent.	Source
En-De	En	50.0M	(Song et al., 2019)
	De	50.0M	
En-Fr/Ro/Zh	En	179.9M	News Crawl 07-17
	Fr	65.4M	News Crawl 07-17
	Ro	2.8M	News Crawl 07-17 + WMT16
	Zh	13.8M	News Crawl 07-21

Table 9: Data statistics for unsupervised machine translation training.