End-to-End Training of Both Translation Models in the Back-Translation Framework

DongNyeong Heo  
Handong Global University, Pohang, Republic of Korea  
sjglsk@gmail.com

Heeyoul Choi  
Handong Global University, Pohang, Republic of Korea  
heeyoul@gmail.com

Abstract

Semi-supervised learning algorithms in neural machine translation (NMT) have significantly improved translation quality compared to the supervised learning methods by using additional monolingual corpora. Among them, back-translation is a theoretically well-structured and cutting-edge method. Given two pre-trained NMT models between source and target languages, one NMT model translates a monolingual sentence to a latent sentence, and the other reconstructs the monolingual input sentence given the latent sentence. Based on this auto-encoding framework, previous work tried to apply the variational auto-encoder’s (VAE) training framework to the back-translation. However, the discrete property of the latent sentence made it impossible to use backpropagation in the end-to-end fashion. In this paper, we propose a categorical reparameterization trick that makes NMT models generate differentiable sentences. Based on the proposed method, end-to-end learning is possible so that two NMT models for the back-translation can be trained as a unified model. In addition, we propose several regularization techniques that are especially advantageous to this framework. Our experiments demonstrate that our method can achieve better BLEU scores than the previous baseline, on the datasets of the WMT18 translation task.

1 Introduction

Supervised learning algorithms in the neural machine translation (NMT) task have shown outstanding performances along with successes of deep learning (Bahdanau et al., 2014; Vaswani et al., 2017). Those algorithms perform well if there is a large amount of bilingual corpus. However, just a few pairs of languages have large bilingual corpora while most of the other pairs do not. In addition, though a language pair has a large bilingual corpus, it should be updated on a regular basis because language is not static over time. New words appear and existing words disappear, corresponding to changes in culture, society, and generations. Therefore, supervised learning algorithms for the NMT task suffer endlessly from a data-hungry situation and expensive data collection for bilingual corpus.

Unlike bilingual corpora, monolingual corpora are easy to collect. Therefore, semi-supervised learning algorithms that use additional monolingual corpora have been suggested to the NMT task. They showed significant performance improvements compared to the supervised learning based NMT algorithms (Gulcehre et al., 2015; Sennrich et al., 2015; He et al., 2016; Hoang et al., 2018; Edunov et al., 2018; Artetxe et al., 2017; Lample et al., 2017; Yang et al., 2018; Caswell et al., 2019; Xu et al., 2020).

The back-translation (BT) method is a cutting-edge approach among semi-supervised learning algorithms (Sennrich et al., 2015). The pre-trained target-to-source NMT (TS-NMT) model, which is trained with only a bilingual corpus, translates target-side monolingual sentences to source-side sentences. Then, each translated source-side sentence is used as the synthetic pair with the corresponding target-side monolingual input sentence. By adding these synthetic pairs to the original source-to-target bilingual corpus, the total size of the training corpus increases as in the data augmentation (Goodfellow et al., 2016). Then, this large corpus is used to train the source-to-target NMT (ST-NMT) model. The same process with source-side monolingual sentences can be applied to train the TS-NMT model.

Since (Sennrich et al., 2015), there has been theoretical developments of the BT framework, especially based on variational auto-encoder (VAE) (Cotterell and Kreutzer, 2018; Zhang et al., 2018; Xu et al., 2020; Kingma and Welling, 2013). Considering the translated sentence as the inferred la-
tent variable of the corresponding monolingual input sentence, BT can be understood as the auto-encoding framework like VAE. For example, the TS-NMT model infers a ‘latent sentence’ in the source language domain given a target-side monolingual sentence as an input, then the ST-NMT model reconstructs the input target-side monolingual sentence from the latent sentence. This process is a target-to-source-to-target (TST) process. In this process, the TS-NMT model approximates the posterior of the latent sentence as an ‘inference model’, and the ST-NMT model estimates the likelihood of the monolingual sentence as a ‘reconstruction model’. Likewise, the source-to-target-to-source (STS) process is conducted with the opposite order and roles with source-side monolingual corpus.

However, training the BT framework like the VAE is challenging for several reasons. First, the non-differentiable latent sentence makes the backpropagation impossible to train the inference model. Note that each word in the latent sentence is discontinuously sampled from the estimated posterior distribution. Second, the latent sentence should be a realistic sentence in that language domain, not an arbitrary sentence, because the latent sentence is a translation result. Note that in the conventional VAE models, the latent space is modeled to have an isotropic Gaussian distribution without any other regularizations on the space, even though there are several works that regularize the space to disentangle the dimensions (Higgins et al., 2017; Kim and Mnih, 2018; Hahn and Choi, 2019). If we do not address this issue, the inference model can be trained to mistranslate the monolingual input sentence focusing on easy reconstruction by the reconstruction model. To avoid these challenges, previous works trained only the reconstruction model in an iteration (Cotterell and Kreutzer, 2018; Zhang et al., 2018; Xu et al., 2020). However, it is an inefficient optimization process because the inference model is not updated for the total objective function directly.

In this paper, we propose a new algorithm that handles the above challenges, so that the two NMT models in the BT framework can be trained by end-to-end backpropagation as in VAE. To overcome the non-differentiable issue, we propose a new reparameterization trick for categorical distribution, the inherent distribution of a sentence. Our proposed trick generates differentiable sentences that facilitate end-to-end backpropagation so that the inference model can be trained together with the reconstruction model. The comparison of previous works and our method is illustrated in Fig. 1. Also, in order to regularize the latent sentence to be realistic, we use the output distribution of the pre-trained language model (Bengio et al., 2003; Paszke et al., 2019) as the prior distribution of the latent sentence. Regularization with the language model is expected to force the latent sentence to be a plausible sentence. Finally, we propose several regularization tricks that are advantageous to this unique framework.

In experiments, we analyze the benefits of our proposed method and tricks with an ablation study on a low-resource dataset from the WMT18 translation task (Bojar et al., 2018). Our proposed method significantly improves the BLEU scores (Papineni et al., 2002) by around 1.1 and 0.8 on English-German and English-Turkish datasets from the WMT18 task, respectively.
2 Related Works

2.1 Back-translation (BT) Methods for NMT

Using a monolingual corpus for the NMT task often improves the translation performance because it can give additional information to understand the language. As a practical method, BT was proposed in (Sennrich et al., 2015). The central concept of BT is augmenting the bilingual corpus with synthetic pairs. The pre-trained TS-NMT model, which is trained by only bilingual corpus, translates target-side monolingual sentences into source-side synthetic sentences. Then, the synthetic source-side sentences and the corresponding target-side monolingual input sentences become synthetic pairs, which can be used as additional pairwise data to train the ST-NMT model.

About the translation methods for the synthetic sentence, various translation methods have been used, such as greedy, beam search, stochastic sampling, filtered sampling (Cheng, 2019; He et al., 2016; Imamura et al., 2018; Edunov et al., 2018; Fadaee and Monz, 2018; Graça et al., 2019; Caswell et al., 2019; Wu et al., 2019). Among them, stochastic sampling or filtered sampling demonstrated promising performances in BT. However, it needs the condition that the pre-trained model should be trained in enough iterations with enough size of the bilingual corpus. Otherwise, the noise can cause undesirable learning signals because the fundamental translation quality is not acceptable. Therefore, selecting a translation method is still an undetermined topic.

On the other side, iterative back-translation (IBT) was proposed, which conducts a single BT training process multiple times (Hoang et al., 2018). In the IBT method, both ST-NMT and TS-NMT models are trained by the BT method iteratively, given both the source-side and target-side of monolingual corpora. By translating synthetic sentences iteratively through the improved NMT models, the quality of the synthetic pairs is improved gradually. As a result, it outperforms the single BT methods significantly. The IBT method can be implemented in two ways: epoch-level IBT and batch-level IBT. While epoch-level IBT newly translates whole sentences after the model converges at the epoch, batch-level IBT translates a mini-batch of sentences at every iteration. Even though epoch-level IBT achieves slightly better BLEU scores, batch-level IBT is significantly faster than epoch-level IBT only with small degradation in the BLEU score (Xu et al., 2020). In our work, we follow the batch-level IBT method.

Based on the IBT method, a probabilistic framework has been proposed considering the translated synthetic sentence as an inferred latent variable that is the aligned representation of a monolingual input sentence (Cotterell and Kreutzer, 2018; Zhang et al., 2018). As we described in Section 1, this BT framework is naturally connected to VAE. In the case of the STS process, the ST-NMT is the inference model (approximated posterior estimator), and the TS-NMT is the reconstruction model (likelihood estimator). In the TST process, ST-NMT and TS-NMT play the opposite roles.

However, it is infeasible to train the BT framework like VAE because of the non-differentiable latent sentence in the inference stage. To avoid the problem, a few methods have been proposed including the expectation-maximization algorithm (Cotterell and Kreutzer, 2018; Zhang et al., 2018) and backpropagation by ignoring the update of the inference model (Xu et al., 2020). Even though their training algorithms are feasible to the BT framework, the inference model loses the learning signals propagated from the reconstruction model. In this paper, we propose a new trick that reparameterizes the inferred latent sentence, so that end-to-end backpropagation is feasible as in conventional VAE.

2.2 Binary Reparameterization Trick

Learning discrete representations in neural networks has several advantages. First, it is proper to represent discrete variables, such as characters and words in natural language. Second, it can be efficiently implemented at the hardware level. Lower memory cost and faster matrix multiplication than those of continuous representations are attractive properties (Hubara et al., 2016; Rastegari et al., 2016).

However, because of the non-differentiable property of discrete representations, backpropagation could not go through them. To overcome this challenge, straight-through estimator (STE) was proposed (Bengio et al., 2013), which estimates the gradient for discretizing operations (e.g. sampling or argmax operations) as 1 in the backward propagation stage. Although STE imposes a bias on the lower layer’s gradient estimation, it is empirically demonstrated as an appropriate gradient estimator in the training of existing binary neural networks.
(Hubara et al., 2016; Rastegari et al., 2016).

Another problem of STE is that implementation is inconvenient with the usual automatic differentiation tool such as Pytorch (Paszke et al., 2019), because it should implement an auxiliary gradient estimator in backward propagation instead of the existing automatic differentiation tool. To overcome such an issue in STE, a reparameterization trick for a binary variable was proposed (Raiko et al., 2014). The operation of this trick is formulated as follows.

\[ s \sim \text{Bern}(p), \quad s \in \{0, 1\}, \quad (1) \]
\[ c = s(1 - p) + (1 - s)(-p), \quad (2) \]
\[ z = p + \text{detach}(c), \quad (3) \]

where \( p \) is a normalized probability of the binary variable. \( \text{Bern}(p) \) and \( c \) are the Bernoulli distribution with a parameter \( p \) and its random variable. \( \text{detach}(c) \) is the operation that detaches its input, \( c \), from the gradient computation graph in backward propagation stage. As a result, the final output, \( z \), is a binary variable, \( z \in \{0, 1\} \), and it can flow the gradient through the first term, \( p \). Although it is a variant of STE, it is easy to implement and work with other network architectures (Rim et al., 2021).

### 3 Proposed Method

In this section, we explain our proposed method for end-to-end training of the BT framework like VAE. In Section 3.1, we propose a reparameterization trick to handle the non-differentiable property of the latent sentence. In Section 3.2, we derive the objective functions for the training in the BT framework with our proposed reparameterization trick. Also, we propose to use the language model’s output distribution as an appropriate prior distribution of the latent sentence. Finally, in Section 3.3, we propose several regularization techniques that can be advantageous to the training of our approach.

#### 3.1 Categorical Reparameterization Trick

The distribution of a sentence can be represented as a sequence of categorical distributions of words, and a sampled sentence from the distribution is non-differentiable. To make backpropagation feasible through the sentence, we propose a reparameterization trick for categorical distribution that is inspired by the binary reparameterization trick, as shown in Section 2.2. Because we handle the categorical distribution (or multinoulli distribution), the Bernoulli distribution in Eq. 1 is replaced by the multinoulli distribution. Then, a one-hot vector for one word, 

\[ s \in \{0, 1\}^{|V|} \text{ s.t. } \sum_{i=1}^{|V|} s_i = 1, \]

is sampled from the distribution where \( V \) is the vocabulary set. Also, other processes are done by element-wise calculations, which can be formulated as follows.

\[ s \sim \text{Mult}(p), \quad (4) \]
\[ c = s \odot (1 - p) + (1 - s) \odot (-p), \quad (5) \]
\[ z = p + \text{detach}(c), \quad (6) \]

where \( \odot \) is the element-wise multiplication. \( \text{Mult}(p) \) is the multinoulli distribution given the normalized probability vector. Based on \( p \) that is computed by the inference model, we compute the non-differentiable conjugate part, \( c \), that is determined by the sample, \( s \). By adding them together, it outputs the one-hot encoded vector \( z \). Note that \( z \) consists of the deterministic part \( p \) and the stochastic part \( s \). Therefore, backpropagation can flow into the lower layers through \( p \). Fig. 1 illustrates the whole process of this trick in the BT framework. We call it a categorical reparameterization trick (CRT).

There could be a question about what is the benefit of our CRT compared to the Gumbel-softmax trick (Jang et al., 2017), which is a well-known reparameterization trick for the categorical distribution. We believe that there are three benefits with the CRT.

First, the CRT is computationally less expensive than the Gumbel-softmax trick because the Gumbel-softmax trick operates the softmax function twice while the CRT operates it only once. Note that the Gumbel-softmax operates two softmax functions, one for computing the normalized word probability vector and the other for Gumbel-softmax itself. Please refer to (Jang et al., 2017) for the details of the Gumbel-softmax trick. The softmax function is expensive when the number of classes is large, like natural language processing tasks. In practice, we measured the spending time of reparameterization processes of the Gumbel-softmax trick in Pytorch library’s implementation (Paszke et al., 2019) and our CRT. The vocabulary size, sentence length, and mini-batch size were set to 30000, 50, and 60, respectively. In the result, the Gumbel-softmax trick spent 3.35 seconds while our CRT spent 0.98 seconds which is more than three times faster.

More importantly, the CRT can reparameterize all the words while Gumbel-softmax reparameterizes only one word with a maximum probabil-
Figure 2: Free-running inference with reparameterization by our CRT or the Gumbel-softmax trick (marked as GST). Note that the third word’s distribution at the third step (green) is changed at the next step (pink). While our CRT can output the previous word, \( w_c \), the GST only outputs the maximally probable word, \( w_a \), as the final output. Then, the backward propagation through the GST is misguided.

ity. This is a critical property, especially for the BT framework within the Transformer architecture (Vaswani et al., 2017), which is a standard model with the state-of-the-art performances in various natural language processing tasks. Contrary to the recurrent neural network (RNN) based NMT models (Bahdanau et al., 2014), in free-running inference, Transformer newly computes the distributions of previous words at every inference step, while the RNN-based model computes only the new word’s distribution at the inference step. Therefore, a probability of a previous word sampled at the previous step might not be the maximum in its newly computed distribution at the current step. Note that stochastic operations, such as dropout (Srivastava et al., 2014), can differ the output word’s distribution whenever it is computed. Also, if the previous word is sampled by stochastic sampling at the previous step, then its probability might not be the maximum at the next step.

Fig. 2 gives an example of the above case with the vocabulary size of 4, \( \{w_a, w_b, w_c, w_d\} \). At the third step of the free-running inference, the word, \( w_c \), is sampled from the distribution (marked in green) calculated at the step. However, for the next steps, the newly computed third distribution (marked in pink) might be different from the original one. In such case, Gumbel-softmax might sample another word (\( w_a \) in the figure) as the final output of the inference model. Then, there is the discrepancy between the conditional input word (\( w_d \)) of \( w_a \), and the final output word (\( w_a \)) at the third step. This is critical to the training of the inference model in the BT framework because the backpropagated gradient goes to a wrong word. When gradient from the reconstruction model flows through the word, \( w_d \), it enforces the inference model to update the probability of \( w_d \) given the conditional words of the final output sequence, (BOS, \( w_a, w_b, w_a \)). However, the inference model updates its probability of \( w_d \) assuming that the conditional words were those of input sequence, (BOS, \( w_a, w_b, w_c \)). Contrary to the GST, our CRT can avoid this problem by just selecting \( w_c \) instead of the sampling operation in the reparameterization process, Eq. 4.

Lastly, our CRT can control the amount of backpropagated gradient while the Gumbel-softmax cannot. In practice, the backpropagated gradient can optimize the inference model which might lead to learning undesirable degenerating solutions, such as copying for the easiest reconstruction. Therefore, the scale of the gradient needs to be adjusted to find a desirable solution. The CRT can adjust the backpropagated gradient by just multiplying the word probability with a coefficient in the reparameterization process. Specifically, we multiply the coefficient, \( \lambda \), to the probability term, \( p \), in Eqs 5 and 6 as follows.

\[
c = s \odot (1 - \lambda p) + (1 - s) \odot (-\lambda p), \quad (7)
\]
\[
z = \lambda p + \text{detach}(c). \quad (8)
\]

Then, \( \lambda \) plays a role of the learning rate only for the inference model. We can set two coefficients, \( \lambda_x \) and \( \lambda_y \), for two languages, respectively, if they need different controls. However, it is hard to implement the same idea in the Gumbel-softmax trick.

3.2 End-to-End Training in BT

In this paper, we propose a new objective functions following (Xu et al., 2020). The final objective function can be decomposed into two terms with two types of processes: bilingual and monolingual processes. The objective term for the bilingual process is cross-entropy as follows:

\[
J_{ST}(\theta) = \sum_{(x^b, y^b) \in D^b} \log p_\theta(y^b|x^b), \quad (9)
\]
where \( x^b \) and \( y^b \) are a source-side sentence and its paired target-side sentence in the bilingual corpus, \( D^b \), respectively. \( \theta \) is the parameter of the ST-NMT model. Likewise, the objective function for the TS-NMT model, \( J_{TS}(\phi) \), is computed in the same way by switching the source and target sentences and replacing the \( \theta \) with the TS-NMT model’s parameter, \( \phi \).

On the other hand, the objective function for the monolingual process is defined by likelihood of the input sentence as follows:

\[
J_{TST} = \sum_{y^m \in D^m} \log p(y^m), \tag{10}
\]

where \( y^m \) is a target-side monolingual sentence in its monolingual corpus, \( D^m \). Note that the formulations are provided only for the TST process, but the STS process is just symmetric to the TST process.

As usual, \( \log p(y^m) \) in Eq. 10 can be marginalized with latent sentences as follows:

\[
\log p(y^m) = \log \sum_{\hat{x} \sim p(\hat{x})} p(y^m | \hat{x}) p(\hat{x}),
\]

where \( \hat{x} \) is the inferred source-side latent sentence that is the aligned representation of \( y^m \) in the source language domain, and \( p(\hat{x}) \) is a prior distribution of \( \hat{x} \). By introducing an approximated posterior distribution, \( q(\hat{x} | y) \), which is easy to sample \( \hat{x} \) given \( y^m \), we can derive the evidence lower bound objective (ELBO) of the marginal probability, \( \log p(y^m) \), based on Jensen’s inequality as follows:

\[
\log p(y^m) \geq \sum_{\hat{x} \sim q(\hat{x} | y^m)} q(\hat{x} | y^m) \log \frac{p(y^m | \hat{x}) p(\hat{x})}{q(\hat{x} | y^m)},
\]

\[
= \mathbb{E}_{\hat{x} \sim q(\hat{x} | y^m)} [\log p(y^m | \hat{x})] - KL(q(\hat{x} | y^m) \parallel p(\hat{x})],
\]

where \( KL \) is Kullback-Liebler divergence (KL). In this TST process of the BT framework, the inference model, \( q(\hat{x} | y^m) \), is modeled by the TS-NMT model with parameter \( \phi \). Also, the reconstruction model, \( p(y^m | \hat{x}) \), is modeled by the ST-NMT with parameter \( \theta \).

Contrary to conventional VAE, the latent sentence, \( \hat{x} \), should be a realistic sample or a valid sentence in the source-side language. Therefore, we use the language model’s output distribution (Bengio et al., 2003; Paszke et al., 2019) as the prior distribution in the KL term, \( p(\hat{x}) \), instead of isotropic Gaussian in VAE. In practice, ELBO of the TST monolingual process can be formulated as follows.

\[
J_{TST}^{ELBO}(\phi, \theta) = \sum_{y^m \in D^m} \mathbb{E}_{\hat{x} \sim q(\hat{x} | y^m)} \left[ \log p(\hat{x} | y^m) \right] - \alpha_x D_{KL}[q(\hat{x} | y^m) \parallel p(\hat{x})], \tag{11}
\]

where \( p(\hat{x}) \) is the pre-trained and fixed source-side language model, and \( \alpha_x \) is a hyperparameter to control the effect of the KL term in the final objective function. To make backpropagation feasible through the latent sentence, we apply the CRT to the latent sentence, \( \hat{x} \).

Finally, the total objective functions for \( \theta \) and \( \phi \) are as follows:

\[
J_T(\theta, \phi) = J_{ST}(\theta) + J_{TST}^{ELBO}(\phi, \theta), \tag{12}
\]

\[
J_S(\phi, \theta) = J_{TS}(\phi) + J_{TST}^{ELBO}(\theta, \phi). \tag{13}
\]

3.3 Regularization Techniques for BT

Though we expect the models to learn desirable translation mapping with the proposed objective functions, the models might converge to undesirable degenerating solutions. Although the original objective function of the monolingual process (Eq. 10) is maximized by the lower bound (Eq. 11) which includes a language model as prior distribution, it does not guarantee translation quality. In this section, we propose additional regularization techniques to improve the translation performance.

3.3.1 Freezing Evaluating Reconstruction Model (FER)

To maximize utilization of pairwise corpus, we first pre-train the parameters of both NMT models, \( \theta \) and \( \phi \), with the bilingual corpus. Therefore, in the BT framework, the reconstruction model would enforce the inference model to infer desirable latent sentences from the beginning of the monolingual processes. However, this desirable enforcement could be faded as the training goes on. Therefore, we propose a novel regularization technique that can preserve this enforcement which is inspired by the target Q-network update technique in the deep Q-network algorithm (Mnih et al., 2013). We duplicate the reconstruction model into two models, evaluating and learning, and freeze the parameter of the evaluating model for \( k \) iterations. The
evaluating model computes the loss that is backpropagated to the inference model. On the other hand, the learning model computes the loss to update only itself. The evaluating model’s parameters are updated every \( k \)-th iteration by copying it from the learning model, as in the deep Q-network algorithms. By doing this, the inference model can be trained with more stable targets.

### 3.3.2 Sharing Embedding Parameters (SEP)

To improve the performance of our proposed method, we apply a well-known regularization trick, parameter sharing between word embedding matrices of both NMT models. By doing this, not only we reduce the number of parameters, but also we expect the word embedding space is regularized by both inference and reconstruction models in iterations.

### 3.3.3 Annealing Stochasticity (AS)

Lastly, as argued in (Edunov et al., 2018), using stochastic sampling in the inference method gives more chances to find a better solution than the greedy method. However, adding more noise to the inference from the beginning of the training can cause the reconstruction model to learn undesirable translation mapping, especially when the pairwise corpus is small. Therefore, we anneal the ratio of stochastic sampling during the training process as in the scheduled sampling approach in NMT (Bengio et al., 2015). That is, we infer the latent sentence by only the greedy method at the beginning, and slowly increase the ratio of stochastic sampling as the training goes on.

### 4 Experiments and Results

To evaluate our proposed method, we present an ablation study with small part of English-German (En-De) dataset and the results on English-Turkish (En-Tr) dataset from WMT18 translation task (Bojar et al., 2018).

For the NMT models in the experiments, we use the base Transformer architecture (Vaswani et al., 2017). For the ablation study, we used three encoder and decoder layers, while we used the six layers as in the original architecture for other experiments. Before using monolingual corpus, we first pre-trained NMT models with bilingual corpus. In this pre-training stage, we used Adam optimizer (Kingma and Ba, 2014) with the batch size of 60 and the learning rate of 0.001. In addition, we used the inverse root square scheduler for the learning rate schedule during the whole training process, as in fairseq toolkit (Ott et al., 2019). These pre-trained models were used as the bilingual NMT baselines in result tables.

For the language models, we adopted the Transformer-based language model (Paszke et al., 2019), where a linear layer is added as a decoder on top of the encoder of Transformer. We pre-trained both source-side and target-side language models on the integrated datasets of both bilingual and monolingual corpora. We used the same optimization setting as the pre-training stage of the NMT models above.

For the baseline BT, we reimplemented the batch-level IBT method in (Xu et al., 2020), which is described above (Section 2.1). By following the main concept of the batch-level IBT method, we trained the baseline BT with our training strategy described in Section 3.2, except the CRT and the KL terms. Because the baseline BT could not flow gradient through the latent sentence, \( J_{\text{ELBO}}^{\text{STS}}(\theta, \phi) \) and \( J_{\text{ELBO}}^{\text{TST}}(\phi, \theta) \) in Eq. 13 cannot update \( \theta \) and \( \phi \), respectively. The optimization setting was the same as for the pre-training stage of bilingual NMT models. By following the empirical results of (Fadaee and Monz, 2018), each mini-batch consists of 12 bilingual sentences and 48 monolingual sentences for the bilingual and each monolingual process, respectively.

#### 4.1 Ablation Study on Low-Resource Dataset

As mentioned above, we use a small part of the En-De dataset from the WMT18 translation task for an ablation study. In this experiment, we set English to the source-side language, \( x \), and German to the target-side language, \( y \). The bilingual corpus is composed of 0.1M pairs of sentences, and the sizes of English and German monolingual corpora are all 0.5M sentences. We filtered out too long sentences that were longer than 50 words. We used a joined vocabulary set, \( V \), whose total size is 10,000. We used En-De ‘Newstest2017’ as the validation dataset with 3,000 pairwise sentences.

About the training, we followed the same training strategy as for the BT baseline but with the CRT and KL terms as our proposed end-to-end training of the BT framework. After the hyper-parameter search, we set 0.01 for both \( \lambda_x \) and \( \lambda_y \). We set 0.0025 for both coefficients for the KL terms, \( \alpha_x \) and \( \alpha_y \). We individually applied the regularization tricks described in Section 3.3. For FER regular-
ization, we set \( k = 75 \) as the iteration number for freezing the evaluating model. For AS regularization, we linearly increased the ratio of stochastic sampling from 0 to 0.5 during 300,000 iterations.

Table 1: BLEU scores for ablation study on the low-resource En-De dataset, tested on ‘Newstest2018’. The beam search was used with the width of 5. ‘E2E BT’ means our proposed end-to-end BT.

| Model               | En-to-De | De-to-En |
|---------------------|----------|----------|
| Bilingual NMT       | 12.32    | 14.19    |
| Baseline BT         | 19.97    | 22.48    |
| E2E BT              | 20.91    | 23.42    |
| E2E BT +FER         | 21.08    | 23.36    |
| E2E BT +SEP         | 20.96    | 23.73    |
| E2E BT +AS          | 20.80    | 23.30    |

We evaluated the ablation study on the ‘Newstest 2018’ dataset from the WMT18 translation task as presented in Table 1. Compared to ‘Bilingual NMT’, all BT methods significantly improve the performance. Also, our proposed ‘E2E BT’ method significantly improves the performance compared to the ‘Baseline BT’ method. In addition, adding regularization techniques, ‘+FER’ and ‘+SEP’ shows improvement from ‘E2E BT’. Even though ‘+AS’ does not show improvement, we believe that the impacts of regularization techniques depend on the experiment environment, such as the size of the corpus and the characteristics of languages. Therefore, it is worth to trying various combinations of regularizations in other experiments.

4.2 Experiments on the En-Tr Dataset

In this section, we evaluate the proposed methods on the En-Tr dataset from the WMT18 translation task. In this experiment, we set English to the source-side language, \( x \), and Turkish to the target-side language, \( y \). The size of the bilingual corpus was 0.2M pairs of sentences. The sizes of the monolingual corpora for English and German were both 4.8M sentences. We filtered out too long sentences that were longer than 60 words. We used a joined vocabulary set, \( V \), with the size of 10,000. We used En-Tr ‘Newstest2016’ as the validation dataset with 3,000 pairwise sentences.

We followed the same training strategy as in the ablation study experiment but with the following different configurations considering the size of datasets. We set 0.01 for both \( \lambda_x \) and \( \lambda_y \), and set 0.0001 for both coefficients for the KL terms, \( \alpha_x \) and \( \alpha_y \). For FER regularization, we set \( k \) to 100. For AS regularization, we followed the same configuration with the ablation study’s training. Note that we combined both FER and AS regularizations. We did not add SEP regularization because it did not show improvements.

The translation performances are presented in Table 2 which includes results on the ‘Newstest’ of 2017 and 2018 datasets of the WMT translation task. We reimplemented baselines (‘Bilingual NMT’ and ‘Baseline BT’) with more monolingual data and novel optimization techniques compared to the reported performances in (Xu et al., 2020). Compared to our baselines ‘Baseline BT’, the proposed method ‘E2E BT + Regs’ improves the BLEU score by around 0.8 on average.

5 Conclusion and Future Work

In this paper, we proposed a reparameterization trick that makes sentences differentiable and used this trick in the back-translation framework in the neural machine translation task. Compared to previous works that could not use backpropagation in the end-to-end fashion, our proposed method makes it feasible. In addition, we practically apply the VAE’s well-structured training strategy to this back-translation framework. The experiment results demonstrate that our proposed method is beneficial to find better translation models which outperform the baseline. In the future, this method can be applied to other translation tasks with larger amounts of data or other applications with discrete representation in the middle of backpropagation flow.
References

Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. 2017. Unsupervised neural machine translation. arXiv preprint arXiv:1710.11041.

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

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. Advances in neural information processing systems, 28.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. Journal of machine learning research, 3(Feb):1137–1155.

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. 2013. Estimating or propagating gradients through stochastic neurons for conditional computation. arXiv preprint arXiv:1308.3432.

Ondřej Bojar, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Philipp Koehn, and Christof Monz. 2018. Findings of the 2018 conference on machine translation (wmt18). In Proceedings of the Third Conference on Machine Translation, volume 2, pages 272–307.

Isaac Caswell, Ciprian Chelba, and David Grangier. 2019. Tagged back-translation. arXiv preprint arXiv:1906.06442.

Yong Cheng. 2019. Semi-supervised learning for neural machine translation. In Joint training for neural machine translation, pages 25–40. Springer.

Ryan Cotterell and Julia Kreutzer. 2018. Explaining and generalizing back-translation through wake-sleep. arXiv preprint arXiv:1806.04402.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. arXiv preprint arXiv:1808.09381.

Marzиеh Fadaee and Christof Monz. 2018. Back-translation sampling by targeting difficult words in neural machine translation. arXiv preprint arXiv:1808.09006.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. Deep Learning. MIT Press. http://www.deeplearningbook.org.

Miguel Graça, Yunsu Kim, Julian Schamper, Shahram Khadivi, and Hermann Ney. 2019. Generalizing back-translation in neural machine translation. arXiv preprint arXiv:1906.07286.

Caglar Gulcehre, Orhan Firat, Kelvin Xu, Kyunghyun Cho, Loïc Barrault, Hui-Chi Lin, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2015. On using monolingual corpora in neural machine translation. arXiv preprint arXiv:1503.03535.

Sanichul Hahn and Heeyoul Choi. 2019. Disentangling latent factors of variational auto-encoder with whitening. In International Conference on Artificial Neural Networks, pages 590–603. Springer.

Di He, Yingce Xia, Tao Qin, Liwei Wang, Nenghai Yu, Tie-Yan Liu, and Wei-Ying Ma. 2016. Dual learning for machine translation. Advances in neural information processing systems, 29:820–828.

Irina Higgins, Loïc Matthey, Arka Pal, Christopher P. Burgess, Xavier Glorot, Matthew M. Botvinick, Shakir Mohamed, and Alexander Lerchner. 2017. Beta-vae: Learning basic visual concepts with a constrained variational framework. In ICLR.

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.

Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. 2016. Binarized neural networks. Advances in neural information processing systems, 29.

Kenji Imamura, Atsushi Fujita, and Eiichiro Sumita. 2018. Enhancement of encoder and attention using target monolingual corpora in neural machine translation. In Proceedings of the 2nd Workshop on Neural Machine Translation and Generation, pages 55–63.

Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical reparametrization with gumble-softmax. In International Conference on Learning Representations (ICLR 2017). OpenReview. net.

Hyunjik Kim and Andriy Mnih. 2018. Disentangling by factorising. In International Conference on Machine Learning, pages 2649–2658. PMLR.

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

Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc’Aurelio Ranzato. 2017. Unsupervised machine translation using monolingual corpora only. arXiv preprint arXiv:1711.00043.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. 2013. Playing atari with deep reinforcement learning. arXiv preprint arXiv:1312.5602.
Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In Proceedings of NAACL-HLT 2019: Demonstrations.

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

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, and R. Garnett, editors, Advances in Neural Information Processing Systems 32, pages 8024–8035. Curran Associates, Inc.

Tapani Raiko, Mathias Berglund, Guillaume Alain, and Laurent Dinh. 2014. Techniques for learning binary stochastic feedforward neural networks. arXiv preprint arXiv:1406.2989.

Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi. 2016. Xnor-net: ImageNet classification using binary convolutional neural networks. In European conference on computer vision, pages 525–542. Springer.

Daniela N. Rim, Inseon Jang, and Heeyoul Choi. 2021. Deep neural networks and end-to-end learning for audio compression. arXiv preprint arXiv:2105.11681.

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

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.

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

Lijun Wu, Yiren Wang, Yingce Xia, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. 2019. Exploiting monolingual data at scale for neural 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 4207–4216.

Weijia Xu, Xing Niu, and Marine Carpuat. 2020. Dual reconstruction: a unifying objective for semi-supervised neural machine translation. arXiv preprint arXiv:2010.03412.

Zhen Yang, Wei Chen, Feng Wang, and Bo Xu. 2018. Unsupervised neural machine translation with weight sharing. arXiv preprint arXiv:1804.09057.

Zhirui Zhang, Shujie Liu, Mu Li, Ming Zhou, and Enhong Chen. 2018. Joint training for neural machine translation models with monolingual data. In Thirty-Second AAAI Conference on Artificial Intelligence.