Unsupervised Neural Machine Translation with Weight Sharing

Zhen Yang[1,2]**, Wei Chen**[1]**, Feng Wang**[1,2]***, Bo Xu**[1]
[1]Institute of Automation, Chinese Academy of Sciences
[2]University of Chinese Academy of Sciences
`{yangzhen2014, wei.chen.media, feng.wang, xubo}@ia.ac.cn`

Abstract

Unsupervised neural machine translation (NMT) is a recently proposed approach for machine translation which aims to train the model without using any labeled data. The models proposed for unsupervised NMT often use only one shared encoder to map the pairs of sentences from different languages to a shared-latent space, which is weak in keeping the unique and internal characteristics of each language, such as the style, terminology, and sentence structure. To address this issue, we introduce an extension by utilizing two independent encoders but sharing some partial weights which are responsible for extracting high-level representations of the input sentences. Besides, two different generative adversarial networks (GANs), namely the local GAN and global GAN, are proposed to enhance the cross-language translation. With this new approach, we achieve significant improvements on English-German, English-French and Chinese-to-English translation tasks.

1 Introduction

Neural machine translation (Kalchbrenner and Blunsom, 2013; Sutskever et al., 2014; Cho et al., 2014; Bahdanau et al., 2014), directly applying a single neural network to transform the source sentence into the target sentence, has now reached impressive performance (Shen et al., 2015; Wu et al., 2016; Johnson et al., 2016; Gehring et al., 2017; Vaswani et al., 2017). The NMT typically consists of two sub neural networks. The encoder network reads and encodes the source sentence into a

context vector, and the decoder network generates the target sentence iteratively based on the context vector. NMT can be studied in supervised and unsupervised learning settings. In the supervised setting, bilingual corpora is available for training the NMT model. In the unsupervised setting, we only have two independent monolingual corpora with one for each language and there is no bilingual training example to provide alignment information for the two languages. Due to lack of alignment information, the unsupervised NMT is considered more challenging. However, this task is very promising, since the monolingual corpora is usually easy to be collected.

Motivated by recent success in unsupervised cross-lingual embeddings (Artetxe et al., 2016; Zhang et al., 2017b; Conneau et al., 2017), the models proposed for unsupervised NMT often assume that a pair of sentences from two different languages can be mapped to a same latent representation in a shared-latent space (Lample et al., 2017; Artetxe et al., 2017b). Following this assumption, Lample et al. (2017) use a single encoder and a single decoder for both the source and target languages. The encoder and decoder, acting as a standard auto-encoder (AE), are trained to reconstruct the inputs. And Artetxe et al. (2017b) utilize a shared encoder but two independent decoders. With some good performance, they share a glaring defect, i.e., only one encoder is shared by the source and target languages. Although the shared encoder is vital for mapping sentences from different languages into the shared-latent space, it is weak in keeping the uniqueness and internal characteristics of each language, such as the style, terminology and sentence structure. Since each language has its own characteristics, the source and target languages should be encoded and learned independently. Therefore, we conjecture that the shared encoder may be a factor limit-

---

[1]Feng Wang is the corresponding author of this paper

ing the potential translation performance.

In order to address this issue, we extend the encoder-shared model, i.e., the model with one shared encoder, by leveraging two independent encoders with each for one language. Similarly, two independent decoders are utilized. For each language, the encoder and its corresponding decoder perform an AE, where the encoder generates the latent representations from the perturbed input sentences and the decoder reconstructs the sentences from the latent representations. To map the latent representations from different languages to a shared-latent space, we propose the weight-sharing constraint to the two AEs. Specifically, we share the weights of the last few layers of two encoders that are responsible for extracting high-level representations of input sentences. Similarly, we share the weights of the first few layers of two decoders. To enforce the shared-latent space, the word embeddings are used as a reinforced encoding component in our encoders. For cross-language translation, we utilize the back-translation following (Lample et al., 2017). Additionally, two different generative adversarial networks (GAN) (Yang et al., 2017), namely the local and global GAN, are proposed to further improve the cross-language translation. We utilize the local GAN to constrain the source and target latent representations to have the same distribution, whereby the encoder tries to fool a local discriminator which is simultaneously trained to distinguish the language of a given latent representation. We apply the global GAN to finetune the corresponding generator, i.e., the composition of the encoder and decoder of the other language, where a global discriminator is leveraged to guide the training of the generator by assessing how far the generated sentence is from the true data distribution [1]. In summary, we mainly make the following contributions:

- We propose the **weight-sharing constraint** to unsupervised NMT, enabling the model to utilize an **independent encoder for each language.** To **enforce the shared-latent space, we also propose the embedding-reinforced encoders and two different GANs for our model.**

- We conduct extensive experiments on

English-German, English-French and Chinese-to-English translation tasks. Experimental results show that the proposed approach consistently achieves great success.

- Last but not least, we introduce the **directional self-attention to model temporal order information for the proposed model.** Experimental results reveal that it deserves more efforts for researchers to investigate the temporal order information within self-attention layers of NMT.

2 Related Work

Several approaches have been proposed to train NMT models without direct parallel corpora. The scenario that has been widely investigated is one where two languages have little parallel data between them but are well connected by one pivot language. The most typical approach in this scenario is to independently translate from the source language to the pivot language and from the pivot language to the target language (Saha et al., 2016; Cheng et al., 2017). To improve the translation performance, Johnson et al. (2016) propose a multilingual extension of a standard NMT model and they achieve substantial improvement for language pairs without direct parallel training data.

Recently, motivated by the success of cross-lingual embeddings, researchers begin to show interests in exploring the more ambitious scenario where an NMT model is trained from monolingual corpora only. Lample et al. (2017) and Artetxe et al. (2017b) simultaneously propose an approach for this scenario, which is based on pre-trained cross lingual embeddings. Lample et al. (2017) utilizes a single encoder and a single decoder for both languages. The entire system is trained to reconstruct its perturbed input. For cross-lingual translation, they incorporate back-translation into the training procedure. Different from (Lample et al., 2017), Artetxe et al. (2017b) use two independent decoders with each for one language. The two works mentioned above both use a single shared encoder to guarantee the shared latent space. However, a concomitant defect is that the shared encoder is weak in keeping the uniqueness of each language. Our work also belongs to this more ambitious scenario, and to the best of our knowledge, we are one among the first endeavors to investigate how to train an NMT model with monolingual corpora only.
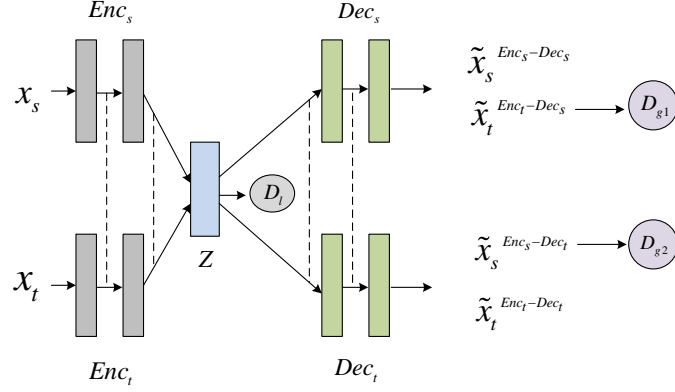
Figure 1: The architecture of the proposed model. We implement the shared-latent space assumption using a weight sharing constraint where the connection of the last few layers in Enc_s and Enc_t are tied (illustrated with dashed lines) and the connection of the first few layers in Dec_s and Dec_t are tied. $\tilde{x}_s^{Enc_s-Dec_s}$ and $\tilde{x}_t^{Enc_t-Dec_t}$ are self-reconstructed sentences in each language. $\tilde{x}_s^{Enc_s-Dec_t}$ is the translated sentence from source to target and $\tilde{x}_t^{Enc_t-Dec_s}$ is the translation in reversed direction. D_l is utilized to assess whether the hidden representation of the encoder is from the source or target language. D_{g1} and D_{g2} are used to evaluate whether the translated sentences are realistic for each language respectively. Z represents the shared-latent space.

3 The Approach

3.1 Model Architecture

The model architecture, as illustrated in figure 1, is based on the AE and GAN. It consists of seven sub networks: including two encoders Enc_s and Enc_t, two decoders Dec_s and Dec_t, the local discriminator D_l, and the global discriminators D_{g1} and D_{g2}. For the encoder and decoder, we follow the newly emerged Transformer (Vaswani et al., 2017). Specifically, the encoder is composed of a stack of four identical layers [2]. Each layer consists of a multi-head self-attention and a simple position-wise fully connected feed-forward network. The decoder is also composed of four identical layers. In addition to the two sub-layers in each encoder layer, the decoder inserts a third sublayer, which performs multi-head attention over the output of the encoder stack. For more details about the multi-head self-attention layer, we refer the reader to (Vaswani et al., 2017). We implement the local discriminator as a multi-layer perceptron and implement the global discriminator based on the convolutional neural network (CNN). Several ways exist to interpret the roles of the sub networks are summarised in table 1. The proposed system has several striking components , which are critical either for the system to be trained in an

unsupervised manner or for improving the translation performance.

Networks	Roles
$\{Enc_s, Dec_s\}$	AE for source language
$\{Enc_t, Dec_t\}$	AE for target language
$\{Enc_s, Dec_t\}$	translation $source \rightarrow target$
$\{Enc_t, Dec_s\}$	translation $target \rightarrow source$
$\{Enc_s, D_l\}$	1st local GAN (GAN_{l1})
$\{Enc_t, D_l\}$	2nd local GAN (GAN_{l2})
$\{Enc_t, Dec_s, D_{g1}\}$	1st global GAN (GAN_{g1})
$\{Enc_s, Dec_t, D_{g2}\}$	2nd global GAN (GAN_{g2})

Table 1: Interpretation of the roles for the subnetworks in the proposed system.

Directional self-attention Compared to recurrent neural network, a disadvantage of the simple self-attention mechanism is that the temporal order information is lost. Although the Transformer applies the positional encoding to the sequence before processed by the self-attention, how to model temporal order information within an attention is still an open question. Following (Shen et al., 2017), we build the encoders in our model on the directional self-attention which utilizes the positional masks to encode temporal order information into attention output. More concretely, two positional masks, namely the forward mask M^f and

---

[2]The layer number is selected according to our preliminary experiment, which is presented in appendix A.

backward mask M^b, are calculated as:

$$M_{ij}^f = \begin{cases} 0 & i < j \\ -\infty & otherwise \end{cases} \quad (1)$$

$$M_{ij}^b = \begin{cases} 0 & i > j \\ -\infty & otherwise \end{cases} \quad (2)$$

With the forward mask M^f, the later token only makes attention connections to the early tokens in the sequence, and vice versa with the backward mask. Similar to (Zhou et al., 2016; Wang et al., 2017), we utilize a self-attention network to process the input sequence in forward direction. The output of this layer is taken by an upper self-attention network as input, processed in the reverse direction.

Weight sharing Based on the shared-latent space assumption, we apply the weight sharing constraint to relate the two AEs. Specifically, we share the weights of the last few layers of the Enc_s and Enc_t, which are responsible for extracting high-level representations of the input sentences. Similarly, we also share the first few layers of the Dec_s and Dec_t, which are expected to decode high-level representations that are vital for reconstructing the input sentences. Compared to (Cheng et al., 2016; Saha et al., 2016) which use the fully shared encoder, we only share partial weights for the encoders and decoders. In the proposed model, the independent weights of the two encoders are expected to learn and encode the hidden features about the internal characteristics of each language, such as the terminology, style, and sentence structure. The shared weights are utilized to map the hidden features extracted by the independent weights to the shared-latent space.

Embedding reinforced encoder We use pre-trained cross-lingual embeddings in the encoders that are kept fixed during training. And the fixed embeddings are used as a reinforced encoding component in our encoder. Formally, given the input sequence embedding vectors $E = \{e_1, \ldots, e_t\}$ and the initial output sequence of the encoder stack $H = \{h_1, \ldots, h_t\}$, we compute H_r as:

$$H_r = g \odot H + (1 - g) \odot E \quad (3)$$

where H_r is the final output sequence of the encoder which will be attended by the decoder (In Transformer, H is the final output of the encoder), g is a gate unit and computed as:

$$g = \sigma(W_1 E + W_2 H + b) \quad (4)$$

where W_1, W_2 and b are trainable parameters and they are shared by the two encoders. The motivation behind is twofold. Firstly, taking the fixed cross-lingual embedding as the other encoding component is helpful to reinforce the shared-latent space. Additionally, from the point of multi-channel encoders (Xiong et al., 2017), providing encoding components with different levels of composition enables the decoder to take pieces of source sentence at varying composition levels suiting its own linguistic structure.

3.2 Unsupervised Training

Based on the architecture proposed above, we train the NMT model with the monolingual corpora only using the following four strategies:

Denoising auto-encoding Firstly, we train the two AEs to reconstruct their inputs respectively. In this form, each encoder should learn to compose the embeddings of its corresponding language and each decoder is expected to learn to decompose this representation into its corresponding language. Nevertheless, without any constraint, the AE quickly learns to merely copy every word one by one, without capturing any internal structure of the language involved. To address this problem, we utilize the same strategy of denoising AE (Vincent et al., 2008) and add some noise to the input sentences (Hill et al., 2016; Artetxe et al., 2017b). To this end, we shuffle the input sentences randomly. Specifically, we apply a random permutation ε to the input sentence, verifying the condition:

$$|\varepsilon(i) - i| \leq \min(k([\frac{steps}{s}] + 1), n), \forall i \in \{1, n\} \quad (5)$$

where n is the length of the input sentence, $steps$ is the global steps the model has been updated, k and s are the tunable parameters which can be set by users beforehand. This way, the system needs to learn some useful structure of the involved languages to be able to recover the correct word order. In practice, we set $k = 2$ and $s = 100000$.

Back-translation In spite of denoising auto-encoding, the training procedure still involves a single language at each time, without considering our final goal of mapping an input sentence from the source/target language to the target/source language. For the cross language training, we utilize the back-translation approach for our unsupervised training procedure. Back-translation has shown its great effectiveness on improving NMT

model with monolingual data and has been widely investigated by (Sennrich et al., 2015a; Zhang and Zong, 2016). In our approach, given an input sentence in a given language, we apply the corresponding encoder and the decoder of the other language to translate it to the other language [3]. By combining the translation with its original sentence, we get a pseudo-parallel corpus which is utilized to train the model to reconstruct the original sentence from its translation.

Local GAN Although the weight sharing constraint is vital for the shared-latent space assumption, it alone does not guarantee that the corresponding sentences in two languages will have the same or similar latent code. To further enforce the shared-latent space, we train a discriminative neural network, referred to as the local discriminator, to classify between the encoding of source sentences and the encoding of target sentences. The local discriminator, implemented as a multi-layer perceptron with two hidden layers of size 256, takes the output of the encoder, i.e., H_r calculated as equation 3, as input, and produces a binary prediction about the language of the input sentence. The local discriminator is trained to predict the language by minimizing the following cross-entropy loss:

$$L_{D_l}(\theta_{D_l}) = \\ -\mathbb{E}_{x \in x_s}[\log p(f = s | Enc_s(x))] \qquad (6) \\ -\mathbb{E}_{x \in x_t}[\log p(f = t | Enc_t(x))]$$

where θ_{D_l} represents the parameters of the local discriminator and $f \in \{s, t\}$. The encoders are trained to fool the local discriminator:

$$L_{Enc_s}(\theta_{Enc_s}) = \\ -\mathbb{E}_{x \in x_s}[\log p(f = t | Enc_s(x))] \qquad (7)$$

$$L_{Enc_t}(\theta_{Enc_t}) = \\ -\mathbb{E}_{x \in x_t}[\log p(f = s | Enc_t(x))] \qquad (8)$$

where θ_{Enc_s} and θ_{Enc_t} are the parameters of the two encoders.

Global GAN We apply the global GANs to fine tune the whole model so that the model is able to generate sentences undistinguishable from the true data, i.e., sentences in the training corpus. Different from the local GANs which updates the parameters of the encoders locally, the

global GANs are utilized to update the whole parameters of the proposed model, including the parameters of encoders and decoders. The proposed model has two global GANs: GAN_{g1} and GAN_{g2}. In GAN_{g1}, the Enc_t and Dec_s act as the generator, which generates the sentence $\tilde{x}_t$ [4] from x_t. The D_{g1}, implemented based on CNN, assesses whether the generated sentence $\tilde{x}_t$ is the true target-language sentence or the generated sentence. The global discriminator aims to distinguish among the true sentences and generated sentences, and it is trained to minimize its classification error rate. During training, the D_{g1} feeds back its assessment to finetune the encoder Enc_t and decoder Dec_s. Since the machine translation is a sequence generation problem, following (Yang et al., 2017), we leverage policy gradient reinforcement training to back-propagate the assessment. We apply a similar processing to GAN_{g2} (The details about the architecture of the global discriminator and the training procedure of the global GANs can be seen in appendix B and C).

There are two stages in the proposed unsupervised training. In the first stage, we train the proposed model with denoising auto-encoding, back-translation and the local GANs, until no improvement is achieved on the development set. Specifically, we perform one batch of denoising auto-encoding for the source and target languages, one batch of back-translation for the two languages, and another batch of local GAN for the two languages. In the second stage, we fine tune the proposed model with the global GANs.

4 Experiments and Results

We evaluate the proposed approach on English-German, English-French and Chinese-to-English translation tasks [5]. We firstly describe the datasets, pre-processing and model hyper-parameters we used, then we introduce the baseline systems, and finally we present our experimental results.

4.1 Data Sets and Preprocessing

In English-German and English-French translation, we make our experiments comparable with previous work by using the datasets from the

---

[3] Since the quality of the translation shows little effect on the performance of the model (Sennrich et al., 2015a), we simply use greedy decoding for speed.

[4] The $\tilde{x}_t$ is $\tilde{x}_t^{Enc_t - Dec_s}$ in figure 1. We omit the superscript for simplicity.

[5] The reason that we do not conduct experiments on English-to-Chinese translation is that we do not get public test sets for English-to-Chinese.

WMT 2014 and WMT 2016 shared tasks respectively. For Chinese-to-English translation, we use the datasets from LDC, which has been widely utilized by previous works (Tu et al., 2017; Zhang et al., 2017a).

WMT14 English-French Similar to (Lample et al., 2017), we use the full training set of 36M sentence pairs and we lower-case them and remove sentences longer than 50 words, resulting in a parallel corpus of about 30M pairs of sentences. To guarantee no exact correspondence between the source and target monolingual sets, we build monolingual corpora by selecting English sentences from 15M random pairs, and selecting the French sentences from the complementary set. Sentences are encoded with byte-pair encoding (Sennrich et al., 2015b), which has an English vocabulary of about 32000 tokens, and French vocabulary of about 33000 tokens. We report results on $newstest2014$.

WMT16 English-German We follow the same procedure mentioned above to create monolingual training corpora for English-German translation, and we get two monolingual training data of 1.8M sentences each. The two languages share a vocabulary of about 32000 tokens. We report results on $newstest2016$.

LDC Chinese-English For Chinese-to-English translation, our training data consists of 1.6M sentence pairs randomly extracted from LDC corpora [6]. Since the data set is not big enough, we just build the monolingual data set by randomly shuffling the Chinese and English sentences respectively. In spite of the fact that some correspondence between examples in these two monolingual sets may exist, we never utilize this alignment information in our training procedure (see Section 3.2). Both the Chinese and English sentences are encoded with byte-pair encoding. We get an English vocabulary of about 34000 tokens, and Chinese vocabulary of about 38000 tokens. The results are reported on $NIST02$.

Since the proposed system relies on the pre-trained cross-lingual embeddings, we utilize the monolingual corpora described above to train the embeddings for each language independently by using word2vec (Mikolov et al., 2013). We then apply the public implementation [7] of the method proposed by (Artetxe et al., 2017a) to map these embeddings to a shared-latent space [8].

4.2 Model Hyper-parameters and Evaluation

Following the base model in (Vaswani et al., 2017), we set the dimension of word embedding as 512, dropout rate as 0.1 and the head number as 8. We use beam search with a beam size of 4 and length penalty $\alpha = 0.6$. The model is implemented in TensorFlow (Abadi et al., 2015) and trained on up to four K80 GPUs synchronously in a multi-GPU setup on a single machine.

For model selection, we stop training when the model achieves no improvement for the tenth evaluation on the development set, which is comprised of 3000 source and target sentences extracted randomly from the monolingual training corpora. Following (Lample et al., 2017), we translate the source sentences to the target language, and then translate the resulting sentences back to the source language. The quality of the model is then evaluated by computing the BLEU score over the original inputs and their reconstructions via this two-step translation process. The performance is finally averaged over two directions, i.e., from source to target and from target to source. BLEU (Papineni et al., 2002) is utilized as the evaluation metric. For Chinese-to-English, we apply the script *mteval-v11b.pl* to evaluate the translation performance. For English-German and English-French, we evaluate the translation performance with the script *multi-belu.pl* [9].

4.3 Baseline Systems

Word-by-word translation (WBW) The first baseline we consider is a system that performs word-by-word translations using the inferred bilingual dictionary. Specifically, it translates a sentence word-by-word, replacing each word with its nearest neighbor in the other language.

Lample et al. (2017) The second baseline is a previous work that uses the same training and testing sets with this paper. Their model belongs to the standard attention-based encoder-decoder framework, which implements the encoder using a bidirectional long short term memory network (LSTM) and implements the decoder using a sim-

---

[6]LDC2002L27, LDC2002T01, LDC2002E18, LDC2003E07, LDC2004T08, LDC2004E12, LDC2005T10

[7]https://github.com/artetxem/vecmap

[8]The configuration we used to run these open-source toolkits can be found in appendix D

[9]https://github.com/moses-smt/mosesdecoder/blob/617e8c8/scripts/generic/multi-bleu.perl;mteval-v11b.pl

	en-de	de-en	en-fr	fr-en	zh-en
Supervised	24.07	26.99	30.50	30.21	40.02
Word-by-word	5.85	9.34	3.60	6.80	5.09
Lample et al. (2017)	9.64	13.33	15.05	14.31	-
The proposed approach	**10.86**	**14.62**	**16.97**	**15.58**	**14.52**

Table 2: The translation performance on English-German, English-French and Chinese-to-English test sets. The results of (Lample et al., 2017) are copied directly from their paper. We do not present the results of (Artetxe et al., 2017b) since we use different training sets.

ple forward LSTM. They apply one single encoder and decoder for the source and target languages.

Supervised training We finally consider exactly the same model as ours, but trained using the standard cross-entropy loss on the original parallel sentences. This model can be viewed as an upper bound for the proposed unsupervised model.

4.4 Results and Analysis

4.4.1 Number of weight-sharing layers

We firstly investigate how the number of weight-sharing layers affects the translation performance. In this experiment, we vary the number of weight-sharing layers in the AEs from 0 to 4. Sharing one layer in AEs means sharing one layer for the encoders and in the meanwhile, sharing one layer for the decoders. The BLEU scores of English-to-German, English-to-French and Chinese-to-English translation tasks are reported in figure 2. Each curve corresponds to a different translation task and the x-axis denotes the number of weight-sharing layers for the AEs. We find that the number of weight-sharing layers shows much effect on the translation performance. And the best translation performance is achieved when only one layer is shared in our system. When all of the four layers are shared, i.e., only one shared encoder is utilized, we get poor translation performance in all of the three translation tasks. This verifies our conjecture that the shared encoder is detrimental to the performance of unsupervised NMT especially for the translation tasks on distant language pairs. More concretely, for the related language pair translation, i.e., English-to-French, the encoder-shared model achieves -0.53 BLEU points decline than the best model where only one layer is shared. For the more distant language pair English-to-German, the encoder-shared model achieves more significant decline, i.e., -0.85 BLEU points decline. And for the most distant

language pair Chinese-to-English, the decline is as large as -1.66 BLEU points. We explain this as that the more distant the language pair is, the more different characteristics they have. And the shared encoder is weak in keeping the unique characteristic of each language. Additionally, we also notice that using two completely independent encoders, i.e., setting the number of weight-sharing layers as 0, results in poor translation performance too. This confirms our intuition that the shared layers are vital to map the source and target latent representations to a shared-latent space. In the rest of our experiments, we set the number of weight-sharing layer as 1.
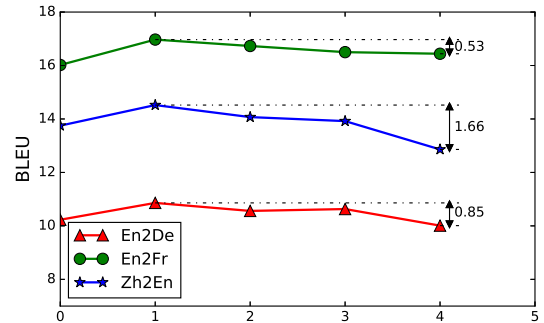

Figure 2: The effects of the weight-sharing layer number on English-to-German, English-to-French and Chinese-to-English translation tasks.

4.4.2 Translation results

Table 2 shows the BLEU scores on English-German, English-French and English-to-Chinese test sets. As it can be seen, the proposed approach obtains significant improvements than the word-by-word baseline system, with at least +5.01 BLEU points in English-to-German translation and up to +13.37 BLEU points in English-to-French translation. This shows that the proposed

	en-de	de-en	en-fr	fr-en	zh-en
Without weight sharing	10.23	13.84	16.02	14.82	13.75
Without embedding-reinforced encoder	10.45	14.17	16.55	15.27	14.10
Without directional self-attention	10.60	14.21	16.82	15.30	14.29
Without local GANs	10.51	14.35	16.40	15.07	14.12
Without Global GANs	10.34	14.05	16.19	15.21	14.09
Full model	**10.86**	**14.62**	**16.97**	**15.58**	**14.52**

Table 3: Ablation study on English-German, English-French and Chinese-to-English translation tasks. Without weight sharing means no layers are shared in the two AEs.

model only trained with monolingual data effectively learns to use the context information and the internal structure of each language. Compared to the work of (Lample et al., 2017), our model also achieves up to +1.92 BLEU points improvement on English-to-French translation task. We believe that the unsupervised NMT is very promising. However, there is still a large room for improvement compared to the supervised upper bound. The gap between the supervised and unsupervised model is as large as 12.3-25.5 BLEU points depending on the language pair and translation direction.

4.4.3 Ablation study

To understand the importance of different components of the proposed system, we perform an ablation study by training multiple versions of our model with some missing components: the local GANs, the global GANs, the directional self-attention, the weight-sharing, the embedding-reinforced encoders, etc. Results are reported in table 3. We do not test the the importance of the auto-encoding, back-translation and the pre-trained embeddings because they have been widely tested in (Lample et al., 2017; Artetxe et al., 2017b). Table 3 shows that the best performance is obtained with the simultaneous use of all the tested elements. The most critical component is the weight-sharing constraint, which is vital to map sentences of different languages to the shared-latent space. The embedding-reinforced encoder also brings some improvement on all of the translation tasks. When we remove the directional self-attention, we get up to -0.3 BLEU points decline. This indicates that it deserves more efforts to investigate the temporal order information in self-attention mechanism. The GANs also significantly improve the translation performance of our system. Specifically, the global GANs

achieve improvement up to +0.78 BLEU points on English-to-French translation and the local GANs also obtain improvement up to +0.57 BLEU points on English-to-French translation. This reveals that the proposed model benefits a lot from the cross-domain loss defined by GANs.

5 Conclusion and Future work

The models proposed recently for unsupervised NMT use a single encoder to map sentences from different languages to a shared-latent space. We conjecture that the shared encoder is problematic for keeping the unique and inherent characteristic of each language. In this paper, we propose the weight-sharing constraint in unsupervised NMT to address this issue. To enhance the cross-language translation performance, we also propose the embedding-reinforced encoders, local GAN and global GAN into the proposed system. Additionally, the directional self-attention is introduced to model the temporal order information for our system.

We test the proposed model on English-German, English-French and Chinese-to-English translation tasks. The experimental results reveal that our approach achieves significant improvement and verify our conjecture that the shared encoder is really a bottleneck for improving the unsupervised NMT. The ablation study shows that each component of our system achieves some improvement for the final translation performance.

Unsupervised NMT opens exciting opportunities for the future research. However, there is still a large room for improvement compared to the supervised NMT. In the future, we would like to investigate how to utilize the monolingual data more effectively, such as incorporating the language model and syntactic information into unsupervised NMT. Besides, we decide to make more

efforts to explore how to reinforce the temporal order information for the proposed model.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems .

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2016. Learning principled bilingual mappings of word embeddings while preserving monolingual invariance. In *Conference on Empirical Methods in Natural Language Processing*. pages 2289–2294.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2017a. Learning bilingual word embeddings with (almost) no bilingual data. In *Meeting of the Association for Computational Linguistics*. pages 451–462.

Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. 2017b. Unsupervised neural machine translation .

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

Yong Cheng, Yang Liu, Qian Yang, Maosong Sun, and Wei Xu. 2016. Neural machine translation with pivot languages. *arXiv preprint arXiv:1611.04928* .

Yong Cheng, Qian Yang, Yang Liu, Maosong Sun, Wei Xu, Yong Cheng, Qian Yang, Yang Liu, Maosong Sun, and Wei Xu. 2017. Joint training for pivot-based neural machine translation. In *Twenty-Sixth International Joint Conference on Artificial Intelligence*. pages 3974–3980.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* .

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Herv Jgou. 2017. Word translation without parallel data .

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning .

Felix Hill, Kyunghyun Cho, and Anna Korhonen. 2016. Learning distributed representations of sentences from unlabelled data. *TACL* .

Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, et al. 2016. Google's multilingual neural machine translation system: Enabling zero-shot translation. *arXiv preprint arXiv:1611.04558* .

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. *EMNLP* pages 1700–1709.

Guillaume Lample, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2017. Unsupervised machine translation using monolingual corpora only .

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. pages 3111–3119.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. *Association for Computational Linguistics* pages 311–318.

Amrita Saha, Mitesh M Khapra, Sarath Chandar, Janarthanan Rajendran, and Kyunghyun Cho. 2016. A correlational encoder decoder architecture for pivot based sequence generation. *arXiv preprint arXiv:1606.04754* .

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

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015b. Neural machine translation of rare words with subword units. *Computer Science* .

Shiqi Shen, Yong Cheng, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. 2015. Minimum risk training for neural machine translation. *arXiv preprint arXiv:1512.02433* .

Tao Shen, Tianyi Zhou, Guodong Long, Jing Jiang, Shirui Pan, and Chengqi Zhang. 2017. Disan: Directional self-attention network for rnn/cnn-free language understanding .

Ilya Sutskever, Oriol Vinyals, and Quoc VV Le. 2014. Sequence to sequence learning with neural networks. *Advances in neural information processing systems* pages 3104–3112.

Zhaopeng Tu, Yang Liu, Shuming Shi, and Tong Zhang. 2017. Learning to remember translation history with a continuous cache .

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

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. 2008. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*. ACM, pages 1096–1103.

Mingxuan Wang, Zhengdong Lu, Jie Zhou, and Qun Liu. 2017. Deep neural machine translation with linear associative unit. *arXiv preprint arXiv:1705.00861* .

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144* .

Hao Xiong, Zhongjun He, Xiaoguang Hu, and Hua Wu. 2017. Multi-channel encoder for neural machine translation. *arXiv preprint arXiv:1712.02109* .

Zhen Yang, Wei Chen, Feng Wang, and Bo Xu. 2017. Improving neural machine translation with conditional sequence generative adversarial nets .

Jiacheng Zhang, Yang Liu, Huanbo Luan, Jingfang Xu, and Maosong Sun. 2017a. Prior knowledge integration for neural machine translation using posterior regularization. In *Meeting of the Association for Computational Linguistics*. pages 1514–1523.

Jiajun Zhang and Chengqing Zong. 2016. Exploiting source-side monolingual data in neural machine translation. In *Conference on Empirical Methods in Natural Language Processing*. pages 1535–1545.

Meng Zhang, Yang Liu, Huanbo Luan, and Maosong Sun. 2017b. Adversarial training for unsupervised bilingual lexicon induction. In *Meeting of the Association for Computational Linguistics*. pages 1959–1970.

Jie Zhou, Ying Cao, Xuguang Wang, Peng Li, and Wei Xu. 2016. Deep recurrent models with fast-forward connections for neural machine translation. *arXiv preprint arXiv:1606.04199* .

A Experiments on the layer number for encoders and decoders

To determine the number of layers for encoders and decoders in our system beforehand, we conduct experiments on English-German translation tasks to test how the amount of layers in encoders and decoders affects the translation performance. We vary the number of layers from 2 to 6 and the results are reported in table 4. We can find that the translation performance achieves substantial improvement with the layer number increasing from 2 to 4. However, with layer number set larger than 4, we get little improvement. To make a trade-off between the translation performance and the computation complexity, we set the layer number as 4 for our encoders and decoders.

layer num	en-de	de-en
2	11.57	14.01
3	12.43	14.99
4	12.86	15.62
5	12.91	15.83
6	12.95	15.79

Table 4: The experiments on the number of layers for encoders and decoders.

B The architecture of the global discriminator

The global discriminator is applied to classify the generated sentences as source language, target language or generated sentences. Following (Yang et al., 2017), we implement the global discriminator based on CNN. Since sentences generated by the generator (the composition of the encoder and decoder) have variable lengths, the CNN padding is used to transform the sentences to sequences with fixed length T, which is the maximum length set for the output of the generator. Given the generated sequences $x_1, \ldots, x_T$, we build the matrix $X_{1:T}$ as:

$$X_{1:T} = x_1; x_2; \ldots; x_T \qquad (9)$$

where $x_t \in R^k$ is the k-dimensional word embedding and the semicolon is the concatenation operator. For the matrix $X_{1:T}$, a kernel $w_j \in R^{l \times k}$ applies a convolutional operation to a window size of l words to produce a series of feature maps:

$$c_{ji} = \rho(BN(w_j \otimes X_{i:i+l-1} + b)) \qquad (10)$$

where $\otimes$ operator is the summation of element-wise production and b is a bias term. ρ is a non-linear activation function which is implemented as ReLu in this paper. To get the final feature with respect to kernel w_j, a max-over-time pooling operation is leveraged over the feature maps:

$$\widetilde{c}_j = max\{c_{j1}, \ldots, c_{jT-l+1}\} \qquad (11)$$

We use various numbers of kernels with different window sizes to extract different features, which are then concatenated to form the final sentence representation x_c. Finally, we pass x_c through a fully connected layer and a softmax layer to generate the probability $p(f_g|x_1, \ldots, x_T)$ as:

$$p(f_g|x_1, \ldots, x_T) = softmax(V * x_c) \qquad (12)$$

where V is the transformation matrix and $f_g \in \{true, generated\}$.

C The training procedure of the global GAN

We apply the global GANs to finetune the whole model. Here, we provide detailed strategies for training the global GANs. Firstly, we generate the machine-generated source language sentences by using Enc_t and Enc_s to decode the monolingual data in target language. Similarly, we get the generated sentences in target language with Enc_s and Dec_t by decoding source language monolingual data. We simply use the greedy sampling method instead of the beam search method for decoding. Next, we pre-train D_{g1} on the combination of true monolingual data and the generated data in the source language. Similarly, we also pre-train D_{g2} on the combination of true monolingual data and the generated data in the target language. Finally, we jointly train the generators and discriminators. The generators are trained with policy gradient training methods. For the details about the policy gradient training, we refer the reader to (Yang et al., 2017).

D The configurations for the open-source toolkits

We train the word embedding use the following script:

./word2vec -train text -output embedding.txt -cbow 0 -size 512 -window 10 -negative 10 -hs 0 -sample 1e- -threads 50 -binary 0 -min-count 5 -iter 10

After we get the embeddings for both the source and target languages, we use the open-source VecMap [10] to map these embeddings to a shared-latent space with the following scripts:

python3 normalize_embeddings.py unit center -i s_embedding.txt -o s_embedding.normalized.txt

python3 normalize_embeddings.py unit center -i t_embedding.txt -o t_embedding.normalized.txt

python3 map_embeddings.py –orthogonal s_embedding.normalized.txt t_embedding.normalized.txt s_embedding.mapped.txt t_embedding.mapped.txt –numerals –self_learning -v

---

[10] https://github.com/artetxem/vecmap