Variational Neural Machine Translation

Biao Zhang[1,2], **Deyi Xiong**[1*], **Jinsong Su**[2], **Hong Duan**[2] **and Min Zhang**[1]
Provincial Key Laboratory for Computer Information Processing Technology
Soochow University, Suzhou, China 215006[1]
Xiamen University, Xiamen, China 361005[2]
zb@stu.xmu.edu.cn, {jssu,hduan}@xmu.edu.cn
{dyxiong, minzhang}@suda.edu.cn

Abstract

Models of neural machine translation are often from a discriminative family of encoder-decoders that learn a conditional distribution of a target sentence given a source sentence. In this paper, we propose a variational model to learn this conditional distribution for neural machine translation: a variational encoder-decoder model that can be trained end-to-end. Different from the vanilla encoder-decoder model that generates target translations from hidden representations of source sentences alone, the variational model introduces a *continuous latent variable* to explicitly model underlying semantics of source sentences and to guide the generation of target translations. In order to perform efficient posterior inference and large-scale training, we build a *neural posterior approximator* conditioned on both the source and the target sides, and equip it with a reparameterization technique to estimate the variational lower bound. Experiments on both Chinese-English and English-German translation tasks show that the proposed variational neural machine translation achieves significant improvements over the vanilla neural machine translation baselines.

1 Introduction

Neural machine translation (NMT) is an emerging translation paradigm that builds on a single and unified end-to-end neural network, instead of using a variety of sub-models tuned in a long training pipeline. It requires a much smaller memory than phrase- or syntax-based statistical machine translation (SMT) that typically has a huge phrase/rule table. Due to these advantages over traditional SMT system, NMT has recently attracted growing interests from both deep learning and machine translation community (Kalchbrenner and Blunsom, 2013; Cho et al., 2014; Sutskever et al., 2014; Bahdanau et al., 2014; Luong et al., 2015a; Luong et al., 2015b; Shen et al., 2015; Meng et al., 2015; Tu et al., 2016).

Current NMT models mainly take a discriminative *encoder-decoder* framework, where a *neural encoder* transforms source sentence $\mathbf{x}$ into distributed representations, and a *neural decoder* generates the corresponding target sentence $\mathbf{y}$ according to these representations[1] (Cho et al., 2014; Sutskever et al., 2014; Bahdanau et al., 2014). Typically, the underlying semantic representations of source and target sentences are learned in an implicit way in this framework, which heavily relies on the attention mechanism (Bahdanau et al., 2014) to identify semantic alignments between source and target words. Due to potential errors in these alignments, the attention-based context vector may be insufficient to capture the entire meaning of a source sentence, hence resulting in undesirable translation phenomena (Tu et al., 2016).

Unlike the vanilla encoder-decoder framework, we model underlying semantics of bilingual sentence pairs explicitly. We assume that there exists a continuous latent variable $\mathbf{z}$ from this underlying semantic space. And this variable, together with $\mathbf{x}$,

[*] Corresponding author

[1] In this paper, we use bold symbols to denote variables, and plain symbols to denote their values. Without specific statement, all variables are multivariate.

Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 521–530,
Austin, Texas, November 1-5, 2016.

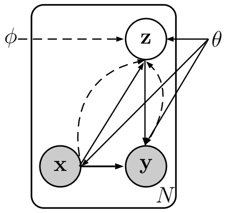

Figure 1: Illustration of VNMT as a directed graph. We use solid lines to denote the generative model $p_\theta(\mathbf{z}|\mathbf{x})p_\theta(\mathbf{y}|\mathbf{z},\mathbf{x})$, and dashed lines to denote the variational approximation $q_\phi(\mathbf{z}|\mathbf{x})$ to the intractable posterior $p(\mathbf{z}|\mathbf{x},\mathbf{y})$. Both variational parameters ϕ and generative model parameters θ are learned jointly.

guides the translation process, i.e. $p(\mathbf{y}|\mathbf{z},\mathbf{x})$. With this assumption, the original conditional probability evolves into the following formulation:

$$p(\mathbf{y}|\mathbf{x}) = \int_z p(\mathbf{y}, z|\mathbf{x})d_z = \int_z p(\mathbf{y}|z,\mathbf{x})p(z|\mathbf{x})d_z \tag{1}$$

This brings in the benefits that the latent variable $\mathbf{z}$ can serve as a global semantic signal that is complementary to the attention-based context vector for generating good translations when the model learns undesirable attentions. However, although this latent variable enables us to explicitly model underlying semantics of translation pairs, the incorporation of it into the above probabilistic model has two challenges: 1) the posterior inference in this model is intractable; 2) large-scale training, which lays the ground for the data-driven NMT, is accordingly problematic.

In order to address these issues, we propose a variational encoder-decoder model to neural machine translation (VNMT), motivated by the recent success of variational neural models (Rezende et al., 2014; Kingma and Welling, 2014). Figure 1 illustrates the graphic representation of VNMT. As deep neural networks are capable of learning highly non-linear functions, we employ them to fit the latent-variable-related distributions, i.e. the prior and posterior, to make the inference tractable. The former is modeled to be conditioned on the source side alone $p_\theta(\mathbf{z}|\mathbf{x})$, because the source and target part of a sentence pair usually share the same semantics so that the source sentence should contain the prior information for inducing the underlying semantics. The latter, instead, is approximated from all observed variables $q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})$, i.e. both the source and the target sides. In order to efficiently train parameters, we apply a reparameterization technique (Rezende et al., 2014; Kingma and Welling, 2014) on the variational lower bound. This enables us to use standard stochastic gradient optimization for training the proposed model. Specifically, there are three essential components in VNMT (The detailed architecture is illustrated in Figure 2):

- A *variational neural encoder* transforms source/target sentence into distributed representations, which is the same as the encoder of NMT (Bahdanau et al., 2014) (see section 3.1).
- A *variational neural inferer* infers the representation of $\mathbf{z}$ according to the learned source representations (i.e. $p_\theta(\mathbf{z}|\mathbf{x})$) together with the target ones (i.e. $q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})$), where the reparameterization technique is employed (see section 3.2).
- And a *variational neural decoder* integrates the latent representation of $\mathbf{z}$ to guide the generation of target sentence (i.e. $p(\mathbf{y}|\mathbf{z},\mathbf{x})$) together with the attention mechanism (see section 3.3).

Augmented with the posterior approximation and reparameterization, our VNMT can still be trained end-to-end. This makes our model not only efficient in translation, but also simple in implementation. To train our model, we employ the conventional maximum likelihood estimation. Experiments on both Chinese-English and English-German translation tasks show that VNMT achieves significant improvements over several strong baselines.

2 Background: Variational Autoencoder

This section briefly reviews the variational autoencoder (VAE) (Kingma and Welling, 2014; Rezende et al., 2014). Given an observed variable $\mathbf{x}$, VAE introduces a continuous latent variable $\mathbf{z}$, and assumes that $\mathbf{x}$ is generated from $\mathbf{z}$, i.e.,

$$p_\theta(\mathbf{x},\mathbf{z}) = p_\theta(\mathbf{x}|\mathbf{z})p_\theta(\mathbf{z}) \tag{2}$$

where θ denotes the parameters of the model. $p_\theta(\mathbf{z})$ is the prior, e.g, a simple Gaussian distribution. $p_\theta(\mathbf{x}|\mathbf{z})$ is the conditional distribution that models the generation procedure, typically estimated via a deep non-linear neural network.

Similar to our model, the integration of $\mathbf{z}$ in Eq. (2) imposes challenges on the posterior inference as

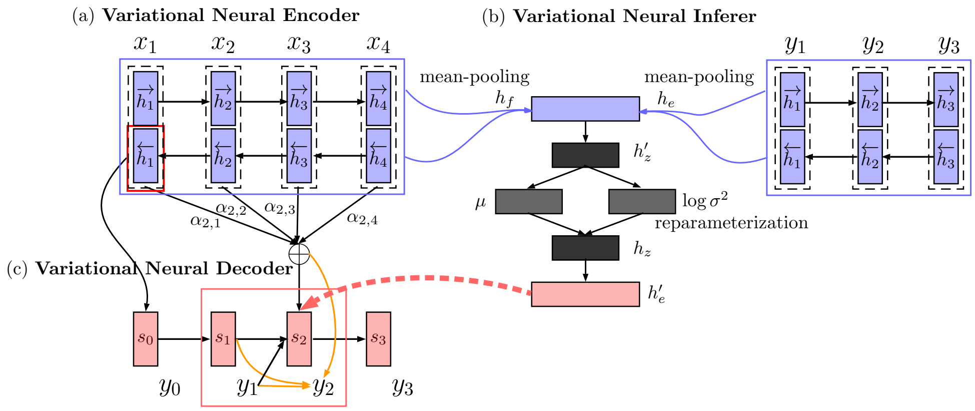

Figure 2: Neural architecture of VNMT. We use blue, gray and red color to indicate the encoder-related $(\mathbf{x}, \mathbf{y})$, underlying semantic $(\mathbf{z})$ and decoder-related $(\mathbf{y})$ representation respectively. The yellow lines show the flow of information employed for target word prediction. The dashed red line highlights the incorporation of latent variable $\mathbf{z}$ into target prediction. f and e represent the source and target language respectively.

well as large-scale learning. To tackle these problems, VAE adopts two techniques: *neural approximation* and *reparameterization*.

Neural Approximation employs deep neural networks to approximate the posterior inference model $q_\phi(\mathbf{z}|\mathbf{x})$, where ϕ denotes the variational parameters. For the posterior approximation, VAE regards $q_\phi(\mathbf{z}|\mathbf{x})$ as a diagonal Gaussian $\mathcal{N}(\mu, \mathrm{diag}(\sigma^2))$, and parameterizes its mean μ and variance σ^2 with deep neural networks.

Reparameterization reparameterizes $\mathbf{z}$ as a function of μ and σ, rather than using the standard sampling method. In practice, VAE leverages the "location-scale" property of Gaussian distribution, and uses the following reparameterization:

$$\tilde{z} = \mu + \sigma \odot \epsilon \tag{3}$$

where ϵ is a standard Gaussian variable that plays a role of introducing noises, and $\odot$ denotes an element-wise product.

With these two techniques, VAE tightly incorporates both the generative model $p_\theta(\mathbf{x}|\mathbf{z})$ and the posterior inference model $q_\phi(\mathbf{z}|\mathbf{x})$ into an end-to-end neural network. This facilitates its optimization since we can apply the standard backpropagation to compute the gradient of the following variational lower bound:

$$\begin{aligned}\mathcal{L}_{\text{VAE}}(\theta, \phi; \mathbf{x}) = &- \mathrm{KL}(q_\phi(\mathbf{z}|\mathbf{x}) || p_\theta(\mathbf{z})) \\ &+ \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] \le \log p_\theta(\mathbf{x})\end{aligned} \tag{4}$$

$\mathrm{KL}(Q||P)$ is the Kullback-Leibler divergence between Q and P. Intuitively, VAE can be considered as a regularized version of the standard autoencoder. It makes use of the latent variable $\mathbf{z}$ to capture the variations ϵ in the observed variable $\mathbf{x}$.

3 Variational Neural Machine Translation

Different from previous work, we introduce a latent variable $\mathbf{z}$ to model the underlying semantic space as a global signal for translation. Formally, given the definition in Eq. (1) and Eq. (4), the variational lower bound of VNMT can be formulated as follows:

$$\begin{aligned}\mathcal{L}_{\text{VNMT}}(\theta, \phi; \mathbf{x}, \mathbf{y}) = -\mathrm{KL}(q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y}) || p_\theta(\mathbf{z}|\mathbf{x})) \\ + \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})}[\log p_\theta(\mathbf{y}|\mathbf{z}, \mathbf{x})]\end{aligned} \tag{5}$$

where $p_\theta(\mathbf{z}|\mathbf{x})$ is our prior model, $q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y})$ is our posterior approximator, and $p_\theta(\mathbf{y}|\mathbf{z}, \mathbf{x})$ is the decoder with the guidance from $\mathbf{z}$. Based on this formulation, VNMT can be decomposed into three components, each of which is modeled by a neural network: a *variational neural inferer* that models $p_\theta(\mathbf{z}|\mathbf{x})$ and $q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y})$ (see part (b) in Figure 2), a *variational neural decoder* that models $p_\theta(\mathbf{y}|\mathbf{z}, \mathbf{x})$ (see part (c) in Figure 2), and a *variational neural encoder* that provides distributed representations of a source/target sentence for the above two modules (see part (a) in Figure 2). Following the information flow illustrated in Figure 2, we describe part (a), (b) and (c) successively.

3.1 Variational Neural Encoder

As shown in Figure 2 (a), the variational neural encoder aims at encoding an input sequence $(w_1, w_2,$

..., w_T) into continuous vectors. In this paper, we adopt the encoder architecture proposed by Bahdanau et al. (2014), which is a bidirectional RNN with a forward and backward RNN. The forward RNN reads the sequence from left to right while the backward RNN in the opposite direction (see the parallel arrows in Figure 2 (a)):

$$\begin{aligned}\overrightarrow{h}_i &= \text{RNN}(\overrightarrow{h}_{i-1}, E_{w_i}) \\ \overleftarrow{h}_i &= \text{RNN}(\overleftarrow{h}_{i+1}, E_{w_i})\end{aligned} \tag{6}$$

where $E_{w_i} \in \mathbb{R}^{d_w}$ is the embedding for word w_i, and $\overrightarrow{h}_i$, $\overleftarrow{h}_i$ are hidden states generated in two directions. Following Bahdanau et al. (2014), we employ the Gated Recurrent Unit (GRU) as our RNN unit due to its capacity in capturing long-distance dependencies.

We further concatenate each pair of hidden states at each time step to build a set of *annotation* vectors ($\mathbf{h}_1$, $\mathbf{h}_2$, ..., $\mathbf{h}_T$), $\mathbf{h}_i^T = \left[\overrightarrow{h}_i^T; \overleftarrow{h}_i^T\right]$. In this way, each annotation vector $\mathbf{h}_i$ encodes information about the i-th word with respect to all the other surrounding words in the sequence. Therefore, these annotation vectors are desirable for the following modeling.

We use this encoder to represent both the source sentence $\{x_i\}_{i=1}^{T_f}$ and the target sentence $\{y_i\}_{i=1}^{T_e}$ (see the blue color in Figure 2). Accordingly, our encoder generates both the source annotation vectors $\{\mathbf{h}_i\}_{i=1}^{T_f} \in \mathbb{R}^{2d_f}$ and the target annotation vectors $\{\mathbf{h}'_i\}_{i=1}^{T_e} \in \mathbb{R}^{2d_e}$. The source vectors flow into the inferer and decoder while the target vectors the posterior approximator.

3.2 Variational Neural Inferer

A major challenge of variational models is how to model the latent-variable-related distributions. In VNMT, we employ neural networks to model both the prior $p_\theta(\mathbf{z}|\mathbf{x})$ and the posterior $q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y})$, and let them subject to a multivariate Gaussian distribution with a diagonal covariance structure.[2] As shown in Figure 1, these two distributions mainly differ in their conditions.

3.2.1 Neural Posterior Approximator

Exactly modeling the true posterior $p(\mathbf{z}|\mathbf{x}, \mathbf{y})$ exactly usually intractable. Therefore, we adopt an approximation method to simplify the posterior inference. Conventional models typically employ the *mean-field* approaches. However, a major limitation of this approach is its inability to capture the true posterior of $\mathbf{z}$ due to its oversimplification. Following the spirit of VAE, we use neural networks for better approximation in this paper, and assume the approximator has the following form:

$$q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y}) = \mathcal{N}(\mathbf{z}; \mu(\mathbf{x}, \mathbf{y}), \sigma(\mathbf{x}, \mathbf{y})^2\mathbf{I}) \tag{7}$$

The mean μ and s.d. σ of the approximate posterior are the outputs of neural networks based on the observed variables $\mathbf{x}$ and $\mathbf{y}$ as shown in Figure 2 (b).

Starting from the variational neural encoder, we first obtain the source- and target-side representation via a *mean-pooling* operation over the annotation vectors, i.e. $\mathbf{h}_f = \frac{1}{T_f}\sum_i^{T_f} \mathbf{h}_i, \mathbf{h}_e = \frac{1}{T_e}\sum_i^{T_e} \mathbf{h}'_i$. With these representations, we perform a non-linear transformation that projects them onto our concerned latent semantic space:

$$\mathbf{h}'_z = g(W_z^{(1)}[\mathbf{h}_f; \mathbf{h}_e] + b_z^{(1)}) \tag{8}$$

where $W_z^{(1)} \in \mathbb{R}^{d_z \times 2(d_f + d_e)}, b_z^{(1)} \in \mathbb{R}^{d_z}$ is the parameter matrix and bias term respectively, d_z is the dimensionality of the latent space, and $g(\cdot)$ is an element-wise activation function, which we set to be $\tanh(\cdot)$ throughout our experiments.

In this latent space, we obtain the abovementioned Gaussian parameters μ and $\log\sigma^2$ through linear regression:

$$\mu = W_\mu \mathbf{h}'_z + b_\mu, \;\; \log\sigma^2 = W_\sigma \mathbf{h}'_z + b_\sigma \tag{9}$$

where $\mu, \log\sigma^2$ are both d_z-dimension vectors.

3.2.2 Neural Prior Model

Different from the posterior, we model (rather than approximate) the prior as follows:

$$p_\theta(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}; \mu'(\mathbf{x}), \sigma'(\mathbf{x})^2\mathbf{I}) \tag{10}$$

We treat the mean μ' and s.d. σ' of the prior as neural functions of source sentence $\mathbf{x}$ alone. This is sound and reasonable because bilingual sentences are semantically equivalent, suggesting that either $\mathbf{y}$ or $\mathbf{x}$

[2] The reasons of choosing Gaussian distribution are twofold: 1) it is a natural choice for modeling continuous variables; 2) it belongs to the family of "location-scale" distributions, which is required for the following reparameterization.

is capable of inferring the underlying semantics of sentence pairs, i.e., the representation of latent variable $\mathbf{z}$.

The neural model for the prior $p_\theta(\mathbf{z}|\mathbf{x})$ is the same as that (i.e. Eq (8) and (9)) for the posterior $q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y})$, except for the absence of $\mathbf{h}_e$. Besides, the parameters for the prior are independent of those for the posterior.

To obtain a representation for latent variable $\mathbf{z}$, we employ the same technique as the Eq. (3) and reparameterized it as $\mathbf{h}_z = \mu + \sigma \odot \epsilon$, $\epsilon \sim \mathcal{N}(0, \mathbf{I})$. During decoding, however, due to the absence of target sentence $\mathbf{y}$, we set $\mathbf{h}_z$ to be the mean of $p_\theta(\mathbf{z}|\mathbf{x})$, i.e., μ'. Intuitively, the reparameterization bridges the gap between the generation model $p_\theta(\mathbf{y}|\mathbf{z}, \mathbf{x})$ and the inference model $q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y})$. In other words, it connects these two neural networks. This is important since it enables the stochastic gradient optimization via standard backpropagation.

We further project the representation of latent variable $\mathbf{h}_z$ onto the target space for translation:

$$\mathbf{h}'_e = g(W_z^{(2)} \mathbf{h}_z + b_z^{(2)}) \tag{11}$$

where $\mathbf{h}'_e \in \mathbb{R}^{d'_e}$. The transformed $\mathbf{h}'_e$ is then integrated into our decoder. Notice that because of the noise from ϵ, the representation $\mathbf{h}'_e$ is not fixed for the same source sentence and model parameters. This is crucial for VNMT to learn to avoid overfitting.

3.3 Variational Neural Decoder

Given the source sentence $\mathbf{x}$ and the latent variable $\mathbf{z}$, our decoder defines the probability over translation $\mathbf{y}$ as a joint probability of ordered conditionals:

$$p(\mathbf{y}|\mathbf{z}, \mathbf{x}) = \prod_{j=1}^{T_e} p(y_j|y_{<j}, \mathbf{z}, \mathbf{x}) \tag{12}$$

$$\text{where} \quad p(y_j|y_{<j}, \mathbf{z}, \mathbf{x}) = g'(y_{j-1}, s_{j-1}, c_j)$$

The feed forward model $g'(\cdot)$ (see the yellow arrows in Figure 2) and context vector $c_j = \sum_i \alpha_{ji} \mathbf{h}_i$ (see the "$\oplus$" in Figure 2) are the same as (Bahdanau et al., 2014). The difference between our decoder and Bahdanau et al.'s decoder (2014) lies in that in addition to the context vector, our decoder integrates the representation of the latent variable, i.e. $\mathbf{h}'_e$, into the computation of s_j, which is denoted by the bold dashed red arrow in Figure 2 (c).

Formally, the hidden state s_j in our decoder is calculated by[3]

$$\begin{aligned}
s_j &= (1 - u_j) \odot s_{j-1} + u_j \odot \tilde{s}_j, \\
\tilde{s}_j &= \tanh(W E_{y_j} + U[r_j \odot s_{j-1}] + C c_j + V \mathbf{h}'_e) \\
u_j &= \sigma(W_u E_{y_j} + U_u s_{j-1} + C_u c_j + V_u \mathbf{h}'_e) \\
r_j &= \sigma(W_r E_{y_j} + U_r s_{j-1} + C_r c_j + V_r \mathbf{h}'_e)
\end{aligned}$$

Here, r_j, u_j, $\tilde{s}_j$ denotes the reset gate, update gate and candidate activation in GRU respectively, and $E_{y_j} \in \mathbb{R}^{d_w}$ is the word embedding for target word. W, W_u, $W_r \in \mathbb{R}^{d_e \times d_w}$, U, U_u, $U_r \in \mathbb{R}^{d_e \times d_e}$, C, C_u, $C_r \in \mathbb{R}^{d_e \times 2d_f}$, and V, V_u, $V_r \in \mathbb{R}^{d_e \times d'_e}$ are parameter weights. The initial hidden state s_0 is initialized in the same way as Bahdanau et al. (2014) (see the arrow to s_0 in Figure 2).

In our model, the latent variable can affect the representation of hidden state s_j through the gate between r_j and u_j. This allows our model to access the semantic information of $\mathbf{z}$ indirectly since the prediction of y_{j+1} depends on s_j. In addition, when the model learns wrong attentions that lead to bad context vector c_j, the semantic representation $\mathbf{h_e}'$ can help to guide the translation process .

3.4 Model Training

We use the Monte Carlo method to approximate the expectation over the posterior in Eq. (5), i.e. $\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})}[\cdot] \simeq \frac{1}{L} \sum_{l=1}^{L} \log p_\theta(\mathbf{y}|\mathbf{x}, \mathbf{h}_z^{(l)})$, where L is the number of samples. The joint training objective for a training instance $(\mathbf{x}, \mathbf{y})$ is defined as follows:

$$\begin{aligned}
\mathcal{L}(\theta, \phi) \simeq & -\mathrm{KL}(q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y}) || p_\theta(\mathbf{z}|\mathbf{x})) \\
& + \frac{1}{L} \sum_{l=1}^{L} \sum_{j=1}^{T_e} \log p_\theta(y_j|y_{<j}, \mathbf{x}, \mathbf{h}_z^{(l)}) \quad (13)
\end{aligned}$$

$$\text{where } \mathbf{h}_z^{(l)} = \mu + \sigma \odot \epsilon^{(l)} \text{ and } \epsilon^{(l)} \sim \mathcal{N}(0, \mathbf{I})$$

The first term is the KL divergence between two Gaussian distributions which can be computed and differentiated without estimation (see (Kingma and Welling, 2014) for details). And the second term is the approximate expectation, which is also differentiable. Suppose that L is 1 (which is used in our experiments), then our second term will be degenerated to the objective of conventional NMT. Intuitively, VNMT is exactly a regularized version of

[3] We omit the bias term for clarity.

System	MT05	MT02	MT03	MT04	MT06	MT08	AVG
Moses	33.68	34.19	34.39	35.34	29.20	22.94	31.21
GroundHog	31.38	33.32	32.59	35.05	29.80	22.82	30.72
VNMT w/o KL	31.40	33.50	32.92	34.95	28.74	22.07	30.44
VNMT	32.25	**34.50**$^{++}$	33.78^{++}	**36.72**$^{\Uparrow++}$	**30.92**$^{\Uparrow++}$	**24.41**$^{\uparrow++}$	**32.07**

Table 1: BLEU scores on the NIST Chinese-English translation task. **AVG** = average BLEU scores on test sets. We highlight the best results in bold for each test set. "$\uparrow$/$\Uparrow$": significantly better than *Moses* ($p < 0.05$/$p < 0.01$); "+/++": significantly better than *GroundHog* ($p < 0.05$/$p < 0.01$);

NMT, where the introduced noise ϵ increases its robustness, and reduces overfitting. We verify this point in our experiments.

Since the objective function in Eq. (13) is differentiable, we can optimize the model parameter θ and variational parameter ϕ jointly using standard gradient ascent techniques.

4 Experiments

4.1 Setup

To evaluate the effectiveness of the proposed VNMT, we conducted experiments on both Chinese-English and English-German translation tasks. Our Chinese-English training data[4] consists of 2.9M sentence pairs, with 80.9M Chinese words and 86.4M English words respectively. We used the NIST MT05 dataset as the development set, and the NIST MT02/03/04/06/08 datasets as the test sets for the Chinese-English task. Our English-German training data[5] consists of 4.5M sentence pairs with 116M English words and 110M German words[6]. We used the newstest2013 (3000 sentences) as the development set, and the newstest2014 (2737 sentences) as the test set for English-German translation. We employed the case-insensitive BLEU-4 (Papineni et al., 2002) metric to evaluate translation quality, and paired bootstrap sampling (Koehn, 2004) for significance test.

We compared our model against two state-of-the-art SMT and NMT systems:

- *Moses* (Koehn et al., 2007): a phrase-based SMT system.
- *GroundHog* (Bahdanau et al., 2014): an attention-based NMT system.

Additionally, we also compared with a variant of *VNMT*, which does not contain the KL part in the objective (*VNMT w/o KL*). This is achieved by setting $\mathbf{h}_z$ to μ'.

For *Moses*, we adopted all the default settings except for the language model. We trained a 4-gram language model on the Xinhua section of the English Gigaword corpus (306M words) using the SRILM[7] toolkit with modified Kneser-Ney smoothing. Importantly, we used all words in the vocabulary.

For *GroundHog*, we set the maximum length of training sentences to be 50 words, and preserved the most frequent 30K (Chinese-English) and 50K (English-German) words as both the source and target vocabulary , covering approximately 98.9%/99.2% and 97.3%/93.3% on the source and target side of the two parallel corpora respectively . All other words were represented by a specific token "UNK". Following Bahdanau et al. (2014), we set $d_w = 620$, $d_f = 1000$, $d_e = 1000$, and $M = 80$. All other settings are the same as the default configuration (for *RNNSearch*). During decoding, we used the beam-search algorithm, and set beam size to 10.

For *VNMT*, we initialized its parameters with the trained *RNNSearch* model. The settings of our model are the same as that of *GroundHog*, except for some parameters specific to VNMT. Following VAE, we set the sampling number $L = 1$. Additionally, we set $d'_e = d_z = 2d_f = 2000$ according to preliminary experiments. We used the Adadelta algorithm for model training with $\rho = 0.95$. With regard to the source and target encoders, we shared their recurrent parameters but not word embeddings.

We implemented our VNMT based on *GroundHog*[8]. Both NMT systems are trained on a Telsa K40

[4] This corpus consists of LDC2003E14, LDC2004T07, LDC2005T06, LDC2005T10 and LDC2004T08 (Hong Kong Hansards/Laws/News).

[5] This corpus is from the WMT'14 training data (Jean et al., 2015; Luong et al., 2015a)

[6] The preprocessed data can be found and downloaded from http://nlp.stanford.edu/projects/nmt/

[7] http://www.speech.sri.com/projects/srilm/download.html

[8] Our code is publicly available at

System	MT05	MT02	MT03	MT04	MT06	MT08
GroundHog	18.23	22.20	20.19	21.67	19.11	13.41
VNMT	**21.31**	**26.02**	**23.78**	**25.81**	**21.81**	**15.59**

Table 2: BLEU scores on the new dataset. All improvements are significant at $p < 0.01$.

System	Architecture	BLEU
Existing end-to-end NMT systems		
Jean et al. (2015)	RNNSearch	16.46
Jean et al. (2015)	RNNSearch + unk replace	18.97
Jean et al. (2015)	RNNsearch + unk replace + large vocab	19.40
Luong et al. (2015a)	LSTM with 4 layers + dropout + local att. + unk replace	20.90
Our end-to-end NMT systems		
this work	RNNSearch	16.40
	VNMT	17.13^{++}
	VNMT + unk replace	19.58^{++}

Table 3: BLEU scores on the English-German translation task.

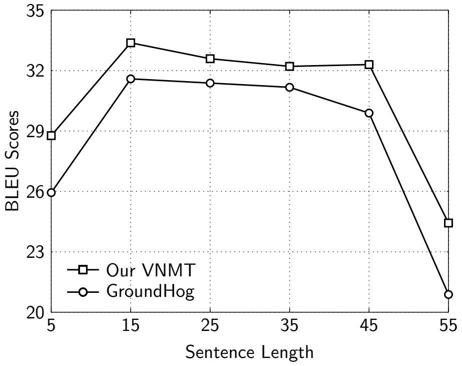

Figure 3: BLEU scores on different groups of source sentences in terms of their length.

GPU. In one hour, *GroundHog* processes about 1100 batches, while our *VNMT* processes 630 batches.

4.2 Results on Chinese-English Translation

Table 1 summarizes the BLEU scores of different systems on the Chinese-English translation tasks. Clearly VNMT significantly improves translation quality in terms of BLEU on most cases, and obtains the best average results that gain 0.86 and 1.35 BLEU points over *Moses* and *GroundHog* respectively. Besides, without the KL objective, *VNMT w/o KL* obtains even worse results than GroundHog. These results indicate the following two points: 1) explicitly modeling underlying semantics by a latent variable indeed benefits neural machine translation, and 2) the improvements of our model are not from enlarging the network.

4.3 Results on Long Sentences

We further testify VNMT on long sentence translation where the vanilla NMT usually suffers from attention failures (Tu et al., 2016; Bentivogli et al., 2016). We believe that the global latent variable can play an important role on long sentence translation.

Our first experiment is carried out on 6 disjoint groups according to the length of source sentences in our test sets. Figure 3 shows the BLEU scores of two neural models. We find that the performance curve of our VNMT model always appears to be on top of that of *GroundHog* with a certain margin. Specifically, on the final group with the longest source sentences, our VNMT obtains the biggest improvement (3.55 BLEU points). Overall, these obvious improvements on all groups in terms of the length of source sentences indicate that the global guidance from the latent variable benefits our VNMT model.

Our second experiment is carried out on a synthetic dataset where each new source sentence is a concatenation of neighboring source sentences in the original test sets. As a result, the average length of source sentences in the new dataset (> 50) is almost twice longer than the original one. Translation results is summarized in Table 2, where our VNMT obtains significant improvements on all new test sets. This further demonstrates the advantage of introducing the latent variable.

4.4 Results on English-German Translation

Table 3 shows the results on English-German translation. We also provide several existing NMT sys-

https://github.com/DeepLearnXMU/VNMT.

Source	两 国 官员 确定 了 今后 会谈 的 日程 和 模式 , 建立 起 进行 持续 对话 的 机制 , 此举 标志 着 巴 印 对话 进程 在 中断 两 年 后 重新 启动 , 为 两 国 逐步 解决 包括 克什米尔 争端 在内 的 所有 悬而未决 的 问题 奠定 了 基础 , 体现 了 双方 可贵 的 和平 诚意 。
Reference	*the officials of the two countries have established the mechanism for continued dialogue down the road, including a confirmed schedule and model of the talks. this symbolizes the restart of the dialogue process between pakistan and india after an interruption of two years and has paved a foundation for the two countries to sort out gradually all the questions hanging in the air, including the kashmir dispute. it is also a realization of their precious sincerity for peace.*
Moses	*officials of the two countries set the agenda for future talks , and the pattern of a continuing dialogue mechanism . this marks a break in the process of dialogue between pakistan and india , two years after the restart of the two countries including kashmir dispute to gradually solve all the outstanding issues have laid the foundation of the two sides showed great sincerity in peace .*
GroundHog	*the two countries have decided to set up a mechanism for conducting continuous dialogue on the agenda and mode of the talks . this indicates that the ongoing dialogue between the two countries has laid the foundation for the gradual settlement of all outstanding issues including the dispute over kashmir .*
VNMT	*the officials of the two countries set up a mechanism for holding a continuous dialogue on the agenda and mode of the future talks, and this indicates that the ongoing dialogue between pakistan and india has laid a foundation for resolving all outstanding issues , including the kashmir disputes , and this serves as a valuable and sincere peace sincerity .*

Table 4: Translation examples of different systems. We highlight important parts in red color.

tems that use the same training, development and testing data. The results show that VNMT significantly outperforms GroundHog and achieves a significant gain of 0.73 BLEU points ($p < 0.01$). With unknown word replacement (Jean et al., 2015; Luong et al., 2015a), VNMT reaches the performance level that is comparable to the previous state-of-the-art NMT results.

4.5 Translation Analysis

Table 4 shows a translation example that helps understand the advantage of VNMT over NMT . As the source sentence in this example is long (more than 40 words), the translation generated by *Moses* is relatively messy and incomprehensible. In contrast, translations generated by neural models (both *GroundHog* and *VNMT*) are much more fluent and comprehensible. However, there are essential differences between *GroundHog* and our *VNMT*. Specifically, *GroundHog* does not translate the phrase "官员" at the beginning of the source sentence. The translation of the clause "体现 了 双方 可贵 的 和平 诚意 。" at the end of the source sentence is completely lost. In contrast, our VNMT model does not miss or mistake these fragments and can convey the meaning of entire source sentence to the target side.

From these examples, we can find that although attention networks can help NMT trace back to relevant parts of source sentences for predicting target translations, capturing the semantics of entire sentences still remains a big challenge for neural machine translation. Since NMT implicitly models variable-length source sentences with fixed-size hidden vectors, some details of source sentences (e.g., the red sequence of words in Table 4) may not be encoded in these vectors at all. VNMT seems to be able to capture these details through a latent variable that explicitly model underlying semantics of source sentences. The promising results suggest that VNMT provides a new mechanism to deal with sentence semantics.

5 Related Work

5.1 Neural Machine Translation

Neural machine translation starts from the sequence to sequence learning, where Sutskever et al. (2014) employ two multilayered Long Short-Term Memory (LSTM) models that first encode a source sentence into a single vector and then decode the translation word by word until a special end token is generated. In order to deal with issues caused by encoding all source-side information into a fixed-length vector, Bahdanau et al. (2014) introduce attention-based

NMT that aims at automatically concentrating on relevant source parts for predicting target words during decoding. The incorporation of attention mechanism allows NMT to cope better with long sentences, and makes it really comparable to or even superior to conventional SMT.

Following the success of attentional NMT, a number of approaches and models have been proposed for NMT recently, which can be grouped into different categories according to their motivations: dealing with rare words or large vocabulary (Jean et al., 2015; Luong et al., 2015b; Sennrich et al., 2015), learning better attentional structures (Luong et al., 2015a), integrating SMT techniques (Cheng et al., 2015; Shen et al., 2015; Feng et al., 2016; Tu et al., 2016), memory network (Meng et al., 2015), etc. All these models are designed within the discriminative encoder-decoder framework, leaving the explicit exploration of underlying semantics with a variational model an open problem.

5.2 Variational Neural Model

In order to perform efficient inference and learning in directed probabilistic models on large-scale dataset, Kingma and Welling (2014) as well as Rezende et al. (2014) introduce variational neural networks. Typically, these models utilize an neural inference model to approximate the intractable posterior, and optimize model parameters jointly with a reparameterized variational lower bound using the standard stochastic gradient technique. This approach is of growing interest due to its success in various tasks.

Kingma et al. (2014) revisit the approach to semi-supervised learning with generative models and further develop new models that allow effective generalization from a small labeled dataset to a large unlabeled dataset. Chung et al. (2015) incorporate latent variables into the hidden state of a recurrent neural network, while Gregor et al. (2015) combine a novel spatial attention mechanism that mimics the foveation of human eyes, with a sequential variational auto-encoding framework that allows the iterative construction of complex images. Very recently, Miao et al. (2015) propose a generic variational inference framework for generative and conditional models of text.

The most related work is that of Bowman et al. (2015), where they develop a variational autoencoder for unsupervised generative language modeling. The major difference is that they focus on the monolingual language model, while we adapt this technique to bilingual translation. Although variational neural models have been widely used in NLP tasks and the variational decoding has been investigated for SMT (Li et al., 2009), the adaptation and utilization of variational neural model to neural machine translation, to the best of our knowledge, has never been investigated before.

6 Conclusion and Future Work

In this paper, we have presented a variational model for neural machine translation that incorporates a continuous latent variable to model the underlying semantics of sentence pairs. We approximate the posterior distribution with neural networks and reparameterize the variational lower bound. This enables our model to be an end-to-end neural network that can be optimized through the stochastic gradient algorithms. Comparing with the conventional attention-based NMT, our model is better at translating long sentences. It also greatly benefits from a special regularization term brought with this latent variable. Experiments on Chinese-English and English-German translation tasks verified the effectiveness of our model.

In the future, since the latent variable in our model is at the sentence level, we want to explore more fine-grained latent variables for neural machine translation, such as the *Recurrent Latent Variable Model* (Chung et al., 2015). We are also interested in applying our model to other similar tasks.

Acknowledgments

The authors were supported by National Natural Science Foundation of China (Grant Nos 61303082, 61672440, 61622209 and 61403269), Natural Science Foundation of Fujian Province (Grant No. 2016J05161), Natural Science Foundation of Jiangsu Province (Grant No. BK20140355), and Research fund of the Provincial Key Laboratory for Computer Information Processing Technology in Soochow University (Grant No. KJS1520). We also thank the anonymous reviewers for their insightful comments.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. In *Proc. of ICLR*.

L. Bentivogli, A. Bisazza, M. Cettolo, and M. Federico. 2016. Neural versus Phrase-Based Machine Translation Quality: a Case Study. *ArXiv e-prints*, August.

S. R. Bowman, L. Vilnis, O. Vinyals, A. M. Dai, R. Jozefowicz, and S. Bengio. 2015. Generating Sentences from a Continuous Space. *ArXiv e-prints*, November.

Y. Cheng, S. Shen, Z. He, W. He, H. Wu, M. Sun, and Y. Liu. 2015. Agreement-based Joint Training for Bidirectional Attention-based Neural Machine Translation. *ArXiv e-prints*, December.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proc. of EMNLP*, pages 1724–1734, October.

Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron C. Courville, and Yoshua Bengio. 2015. A recurrent latent variable model for sequential data. In *Proc. of NIPS*.

S. Feng, S. Liu, M. Li, and M. Zhou. 2016. Implicit Distortion and Fertility Models for Attention-based Encoder-Decoder NMT Model. *ArXiv e-prints*, January.

Karol Gregor, Ivo Danihelka, Alex Graves, and Daan Wierstra. 2015. DRAW: A recurrent neural network for image generation. *CoRR*, abs/1502.04623.

Sébastien Jean, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. 2015. On using very large target vocabulary for neural machine translation. In *Proc. of ACL-IJCNLP*, pages 1–10, July.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In *Proc. of EMNLP*, pages 1700–1709, October.

Diederik P Kingma and Max Welling. 2014. Auto-Encoding Variational Bayes. In *Proc. of ICLR*.

Diederik P. Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. 2014. Semi-supervised learning with deep generative models. In *Proc. of NIPS*, pages 3581–3589.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proc. of ACL*, pages 177–180.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proc. of EMNLP*.

Zhifei Li, Jason Eisner, and Sanjeev Khudanpur. 2009. Variational decoding for statistical machine translation. In *Proc. of ACL*, pages 593–601, August.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015a. Effective approaches to attention-based neural machine translation. In *Proc. of EMNLP*, pages 1412–1421, September.

Thang Luong, Ilya Sutskever, Quoc Le, Oriol Vinyals, and Wojciech Zaremba. 2015b. Addressing the rare word problem in neural machine translation. In *Proc. of ACL-IJCNLP*, pages 11–19, July.

F. Meng, Z. Lu, Z. Tu, H. Li, and Q. Liu. 2015. A Deep Memory-based Architecture for Sequence-to-Sequence Learning. *ArXiv e-prints*, June.

Y. Miao, L. Yu, and P. Blunsom. 2015. Neural Variational Inference for Text Processing. *ArXiv e-prints*, November.

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

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. 2014. Stochastic backpropagation and approximate inference in deep generative models. In *Proc. of ICML*, pages 1278–1286.

R. Sennrich, B. Haddow, and A. Birch. 2015. Neural Machine Translation of Rare Words with Subword Units. *ArXiv e-prints*, August.

S. Shen, Y. Cheng, Z. He, W. He, H. Wu, M. Sun, and Y. Liu. 2015. Minimum Risk Training for Neural Machine Translation. *ArXiv e-prints*, December.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. *CoRR*, abs/1409.3215.

Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. 2016. Coverage-based neural machine translation. *CoRR*, abs/1601.04811.