Variational Neural Machine Translation

Biao Zhang\textsuperscript{1,2}, Deyi Xiong\textsuperscript{2} and Jinsong Su\textsuperscript{1}
Xiamen University, Xiamen, China 361005\textsuperscript{1}
Soochow University, Suzhou, China 215006\textsuperscript{2}
zeb@stu.xmu.edu.cn, jssu@xmu.edu.cn
dyxiong@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 an efficient posterior inference, we build a neural posterior approximator that is conditioned only on the source side. Additionally, we employ a reparameterization technique to estimate the variational lower bound so as to enable standard stochastic gradient optimization and large-scale training for the variational model. Experiments on NIST Chinese-English translation tasks show that the proposed variational neural machine translation achieves significant improvements over both state-of-the-art statistical and 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 interest from both deep learning and machine translation community [Kalchbrenner and Blunsom, 2013; Cho et al., 2014; Sutskever et al., 2014; Bahdanau et al., 2014; Jean et al., 2015; Luong et al., 2015a; Luong et al., 2015b; Shen et al., 2015; Meng et al., 2015; Tu et al., 2016].

Most NMT models take a discriminative encoder-decoder framework, where a neural encoder transforms source sentence $x$ into a distributed representation, and a neural decoder generates the corresponding target sentence $y$ according to the distributed representation $[\text{Cho et al.,} \ 2014; \text{Sutskever et al.,} \ 2014; \text{Bahdanau et al.,} \ 2014]$. Typically, the underlying semantic representations of source and target sentences are learned in an implicit way in this encoder-decoder framework. They are encoded into hidden states of the encoder and the decoder.

Unlike the vanilla encoder-decoder framework, we model underlying semantics of bilingual sentence pairs explicitly. We assume that there exists a continuous latent variable $z$ from this underlying semantic space. And this variable together with $x$ guide the translation process, i.e. $p(y|z, x)$. With this assumption, the original conditional probability evolves into the following formulation:

$$p(y|x) = \sum_z p(y, z|x) = \sum_z p(y|z, x)p(z)$$

(1)

Although this latent variable enables us to explicitly model underlying semantics of translation pairs, the incorporation of it into the above probability model has two challenges: 1) the poste-
Graph. We use solid lines to denote the generative model $p_{\theta}(z)p_{\phi}(y|z,x)$, and dashed lines to denote the variational approximation $q_{\phi}(z|x)$ to the intractable posterior $p_{\theta}(z|x)$. Both variational parameters $\phi$ and generative model parameters $\theta$ are learned jointly.

Augmented with the posterior approximation and reparameterization, our VNMT can 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 NIST Chinese-English translation tasks show that VNMT achieves significant improvements over a state-of-the-art SMT and NMT system.

2 Background: Variational Autoencoder

In this section, we briefly reviews the variational autoencoder (VAE) (Kingma and Welling, 2014) (Rezende et al., 2014), one of the most classical variational neural models. Yet another reason to particularly introduce the variational autoencoder in this paper is to ensure the integrity of the proposed variational NMT as it also belongs to the family of encoder-decoders.

Given an observed variable $x$, VAE introduces a continuous latent variable $z$, and assumes that $x$ is generated from $z$, i.e.,

$$p_{\theta}(x, z) = p_{\theta}(x|z)p_{\theta}(z)$$  \hspace{1cm} (2)

where $\theta$ denotes the parameter of the model, $p_{\theta}(z)$ is the prior, and $p_{\theta}(x|z)$ is the conditional distribution that models the generation procedure. Typically, $p_{\theta}(z)$ is treated as a simple Gaussian distribution, and deep non-linear neural networks are used to perform the generation, i.e., $p_{\theta}(x|z)$.

Similar to our model, the integration of $z$ in Eq. (2) raises challenges for posterior inference as 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 $q_{\phi}(z|x)$, where $\phi$ is the variational parameter. For the posterior approximation, VAE equips $q_{\phi}(z|x)$ with a diagonal Gaussian $\mathcal{N}(\mu, \text{diag}(\sigma^2))$, and parameterize its mean $\mu$ and variance $\sigma^2$ with deep neural networks respectively.

Reparameterization reparameterizes $z$ as a function of $\mu$ and $\sigma$, rather than using the standard sampling method. In practice, VAE leverages the “location-scale” property of Gaussian distribution, and uses the following reparameterization:

$$\tilde{z} = \mu + \sigma \odot \epsilon$$  \hspace{1cm} (3)
where $\epsilon$ is a standard Gaussian variable that plays a role of introducing noises, and $\odot$ denotes an element-wise product.

With these two techniques, VAE bridges the gap between the generative model $p_\theta(x|z)$ and the posterior inference model $q_\phi(z|x)$, and operates as an end-to-end neural network. This facilitates its optimization since we can apply the standard backpropagation to the following variational lower bound to compute its gradient.

$$
\mathcal{L}_{\text{VAE}}(\theta, \phi; x) = - \text{KL}(q_\phi(z|x)||p_\theta(z)) + \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] \leq \log p_\theta(x)
$$  \hspace{1cm} (4)

KL$(Q||P)$ is the Kullback-Leibler divergence between two distributions $Q$ and $P$. Intuitively, VAE can be considered as a specific regularized version of the standard autoencoder. Because of the variations in Eq. (3), VAE learns the representation of the latent variable not as single points, but as soft ellipsoidal regions in the latent space, forcing the representation to fill the space rather than memorizing the training data as isolated representations. Therefore, the latent variable $z$ is able to capture the variations in the observed variable $x$.

We follow the spirit of VAE to introduce a latent variable $z$ into the translation model $p(y|x)$. We will give the detailed description in the next section.

3 Variational Neural Machine Translation

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

$$
\mathcal{L}_{\text{VNMT}}(\theta, \phi; x, y) = - \text{KL}(q_\phi(z|x)||p_\theta(z)) + \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(y|z, x)]
$$  \hspace{1cm} (5)

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

Notice that we approximate the posterior to be conditioned on $x$ alone, rather than $y$ or $(x, y)$. This is sound and reasonable because bilingual sentences are semantic equivalent, which means that either $y$ or $x$ is capable of inferring the underlying semantics of sentence pairs, i.e., the representation of latent variable $z$.

3.1 Variational Neural Encoder

As shown in Figure 2(a), the variational neural encoder aims at encoding an input source sentence $(x_1, x_2, \ldots, x_{T_x})$ into a continuous vector. In this paper, we adopt the encoder architecture proposed by Bahdanau et al. (2014), which is a bidirectional RNN that consists of a forward RNN and backward RNN. The forward RNN reads source sentence from left to right while the backward RNN in the opposite direction (see the parallel arrows in
Figure 2 (a):
\[
\begin{align*}
\vec{h}_i &= \text{RNN}(\vec{h}_{i-1}, E_{x_i}) \\
\hat{h}_i &= \text{RNN}(\hat{h}_{i+1}, E_{x_i})
\end{align*}
\]
where $E_{x_i} \in \mathbb{R}^{d_w}$ is the $d_w$-dimensional embedding for source word $x_i$, and $\vec{h}_i, \hat{h}_i \in \mathbb{R}^{d_f}$ are $d_f$-dimensional 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 $(h_1, h_2, \ldots, h_{T_f})$, where
\[
\begin{bmatrix}
\vec{h}_i^T \\
\hat{h}_i^T
\end{bmatrix}
\]
In this way, each annotation vector $h_i \in \mathbb{R}^{2d_f}$ encodes information about the $i$-th word with respect to all the other surrounding words in the source sentence. Therefore, these annotation vectors are desirable for the following modeling.

### 3.2 Variational Neural Approximator

As the posterior inference model $p(z|x)$ is intractable in most cases, we adopt an approximation method to simplify the posterior inference. Conventional models usually employ the mean-field approaches. However, a major limitation of this approach is its inability to capture the true posterior of $z$ due to its oversimplification. Following the spirit of VAE, we use neural networks for better approximation in this paper.

Similar to previous work (Kingma and Welling, 2014; Rezende et al., 2014), we let $q_\phi(z|x)$ be a multivariate Gaussian distribution with a diagonal covariance structure:
\[
q_\phi(z|x) = \mathcal{N}(z; \mu, \sigma^2 I)
\]
where the mean $\mu$ and s.d. $\sigma$ of the approximate posterior are the outputs of the neural network as shown in Figure 2 (b). The reason of choosing Gaussian distribution is twofold: 1) it is a natural choice for describing continuous variables; 2) it belongs to the family of “location-scale” distributions, which is required for the following reparameterization.

We first synthesize the source-side information via a mean-pooling operation over the annotation vectors:
\[
h_f = \frac{1}{T_f} \sum_{i} h_i
\]
With this source representation, we perform a non-linear transformation that projects it onto our concerned latent semantic space:
\[
h'_z = g(W_z^{(1)} h_f + b_z^{(1)})
\]
where $W_z^{(1)} \in \mathbb{R}^{d_z \times 2d_f}, b_z^{(1)} \in \mathbb{R}^{d_z}$ 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 $\tanh(\cdot)$ throughout our experiments.

In this latent space, we further obtain the above-mentioned Gaussian parameters $\mu$ and $\log \sigma^2$ through linear regression:
\[
\begin{align*}
\mu &= W_\mu h_f' + b_\mu \\
\log \sigma^2 &= W_\sigma h_f' + b_\sigma
\end{align*}
\]
where $W_\mu, W_\sigma \in \mathbb{R}^{d_z \times d_z}, b_\mu, b_\sigma \in \mathbb{R}^{d_z}$ are the parameters, and $\mu, \log \sigma^2$ are both $d_z$-dimension vectors.

Similar to the Eq. (3), the final representation for latent variable $z$ can be reparameterized as $h_z = \mu + \epsilon \cdot \sigma \sim \mathcal{N}(0, I)$. During decoding, we set $h_z$ to be the mean of $p(z|x)$, i.e., $\mu$. Intuitively, the reparameterization bridges the gap between the model $p_\theta(y|z, x)$ and the inference model $q_\phi(z|x)$. In other words, it connects these two neural networks. This is important since it enables the stochastic gradient optimization via standard backpropagation.

To perform translation in the target language, we further project the representation of latent variable $h_z$ onto the target space:
\[
h_e = g(W_z^{(2)} h_z + b_z^{(2)})
\]
where $W_z^{(2)} \in \mathbb{R}^{d'_e \times d_z}, b_z^{(2)} \in \mathbb{R}^{d'_e}$ are parameters, and $d'_e$ is the dimensionality of the target space. The transformed $h_e$ is then integrated into our decoder. Notice that because of the noise from $\epsilon$, the representation $h_e$ is not fixed for the same source sentence and model parameters. This is crucial for VNMT to learn to be insensitive to small noises.

### 3.3 Variational Neural Decoder

Given the source sentence $x$ and the latent variable $z$, our decoder defines the probability over transla-
tion $y$ as a joint probability of ordered conditionals:

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

(14)

where,

$$p(y_j|y_{<j}, z, x) = g'(y_{j-1}, s_j, c_j)$$

The feed forward model $g'()$ (see the yellow arrows in Figure 2) and context vector $c_j = \sum_i \alpha_{ij} h_i$ (see the “$\alpha$” 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., $h_e$, into the computation of $s_j$, which is denoted by the bold red dashed arrow in Figure 2(c).

Formally, the hidden state $s_j$ in our decoder is calculated by

$$s_j = (1 - z_j) \odot s_{j-1} + z_j \odot \tilde{s}_j,$$

where

$$\tilde{s}_j = \tanh(Wy_j + U[r_j \odot s_{j-1}] + Cc_j + Vh_e)$$

$$z_j = \sigma(Wz_j + Uz_{j-1} + Cc_j + Vh_e)$$

$$r_j = \sigma(Wr_j + Ur_{j-1} + Cc_j + Vh_e)$$

Here, $r_j$, $z_j$, $\tilde{s}_j$ denotes the reset gate, update gate and candidate activation in GRU respectively, and $E_{y_j} \in \mathbb{R}^{d_y}$ is the $d_y$-dimensional word embedding for target word. $W$, $W_z$, $W_r \in \mathbb{R}^{d_y \times d_w}$, $U$, $U_z$, $U_r \in \mathbb{R}^{d_x \times d_y}$, $C$, $C_z$, $C_r \in \mathbb{R}^{d_x \times 2d_y}$, and $V$, $V_z$, $V_r \in \mathbb{R}^{d_x \times d_e}$ are parameter weights. The initial hidden state $s_0$ is initialized in the same way by 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 $r_j$ and $z_j$. This allows our model to access the semantic information of $z$ indirectly since the prediction of $y_{j+1}$ depends on $s_j$.

### 3.4 Model Training

As shown in Eq. (5), to optimize our model, we need to compute an expectation over the approximate posterior, that is, $\mathbb{E}_{q_\phi(x|z)}[\cdot]$. This expectation, again, is intractable. Following VAE, we employing the Monte Carlo method to estimate this expectation:

$$\mathbb{E}_{q_\phi(x|z)}[\log p_\theta(y|z, x)] \simeq \frac{1}{L} \sum_{l=1}^{L} \log p_\theta(y|x, h_e^{(l)})$$

(15)

where $L$ is the number of samples.

As all intractable computations are eliminated, the joint training objective for a training instance $(x, y)$ is defined as follows:

$$\mathcal{L}(\theta, \phi) \simeq \frac{1}{2} \sum_{k=1}^{d_z} [1 + \log(\sigma_k^2) - \mu_k^2 - \sigma_k^2] + \frac{1}{L} \sum_{l=1}^{L} \sum_{j=1}^{T_x} \log p_\theta(y_j|y_{<j}, x, h_e^{(l)})$$

(16)

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

The first term is the KL divergence in Eq. (5) 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 employed in our experiments), then our second term will be degenerated to the objective of conventional NMT. Intuitively, the VNMT is a regularized version of NMT whose regularization is exactly the first term.

Since the objective function in Eq. (16) is differentiable, we are able to optimize the model parameter $\theta$ and variational parameter $\phi$ jointly using standard gradient ascent techniques. The training procedure for VNMT is summarized in Algorithm 1.

### 4 Experiments

#### 4.1 Setup

To evaluate the effectiveness of the proposed VNMT, we conducted experiments on the NIST Chinese-English translation tasks. Our training
Table 1: BLEU scores on the NIST Chinese-English translation tasks. AVG = average BLEU scores on test sets. The number after VNMT denotes the setting for $d_z$. We highlight the best results in bold for each test set. “+”: significantly better than Moses ($p < 0.05$); “++”: significantly better than Moses ($p < 0.01$); “+++”: significantly better than GroundHog ($p < 0.05$).

| 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-1000   | 32.42 | 34.37 | 33.69 | 36.30 | 30.31 | 23.53 | 31.64|
| VNMT-2000   | 32.70 | 34.13 | 33.64 | 36.74 | 31.10 | 23.88 | 31.90|
| VNMT-3000   | 32.43 | 34.80 | 34.49 | 36.27 | 30.16 | 23.83 | 31.91|
| VNMT-4000   | 32.17 | 34.23 | 34.21 | 36.73 | 30.81 | 23.64 | 31.92|

Table 1: BLEU scores on the NIST Chinese-English translation tasks. AVG = average BLEU scores on test sets. The number after VNMT denotes the setting for $d_z$. We highlight the best results in bold for each test set. “+”: significantly better than Moses ($p < 0.05$); “++”: significantly better than Moses ($p < 0.01$); “+++”: significantly better than GroundHog ($p < 0.05$).

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

- **Moses** (Koehn and Hoang, 2007): the conventional phrase-based SMT system.
- **GroundHog** (Bahdanau et al., 2014): the attentional NMT system.

For Moses, we followed 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 toolkit with modified Kneser-Ney smoothing.

For GroundHog, we set the maximum length of training sentences to be 50 words, and preserved the most frequent 30K words as the source and target vocabulary respectively. Following Bahdanau et al. (2014), we set $d_{aw} = 620$, $d_f = 1000$, $d_e = 1000$, 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 the beam size to 10.

For our VNMT, we initialized its parameters with the RNNSearch in GroundHog. The settings of our model are the same as that of GroundHog, except for some parameters specific to VNMT. Following the VAE, we set the sampling number $L = 1$. Additionally, we set $d'_e = 2d_f = 2000$.

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

According to preliminary experiments. We used the Adadelta algorithm for parameterUpdate in Algorithm 1. With regard to the dimension of latent semantic space $d_z$, we tried several different settings: 1000, 2000, 3000, and 4000.

We implemented our VNMT based on GroundHog. Both NMT systems are trained on a Tesla K40 GPU. In one hour, the GroundHog system processes about 1100 batches, while our VNMT processes 400~750 batches when $d_z$ ranges from 4000 to 1000.

4.2 Translation Results

Table 1 summarizes the BLEU scores of different systems on the Chinese-English translation tasks. No matter which dimensionality we set for $d_z$, VNMT consistently improves translation quality in terms of BLEU on all test sets. Specifically, when $d_z = 4000$, VNMTX obtains the best average results that gain 0.71 and 1.20 BLEU points over Moses and GroundHog respectively. This indicates that explicitly modeling underlying semantics by a latent variable is indeed beneficial for neural machine translation.

With respect to the dimensionality $d_z$ of the latent variable, we do not observe consistent improvements on each test set as $d_z$ increases. The reason may be that our test sets have different data

---

4This corpus is a combination of LDC2003E14, LDC2004T07, LDC2005T06, LDC2005T10 and LDC2004T08 (Hong Kong Hansards/Laws/News).

5http://www.speech.sri.com/projects/srilm/download.html
The official differences between GroundHog and our VNMT model are much more profound sincere sincerity of both sides.

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 sequences of words in Table 2) 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 model and encode sentence semantics.

5 Related Work

There are roughly two lines of research related to our work: neural machine translation and variational neural model. We describe them in succession.

5.1 Neural Machine Translation

The idea of performing encoding and decoding from a source sentence to a target sentence originates from early neural translation models. Kalchbrenner and Blunsom (2013) adopt a con-

Table 2: Translation examples of different systems extracted from NIST 2004 datasets. We highlight relatively important part in red color.

| Source | 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. |
| Reference | 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. |
| Moses | 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. |
| GroundHog | the officials of the two countries have established a mechanism for holding talks on the agenda and pattern of the future talks. this indicates that the dialogues between the two countries have laid a foundation for the two countries to gradually resolve all outstanding issues, including the dispute over kashmir, and this has embodied the profound sincere sincerity of both sides. |
| VNMT | 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. |

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 these fragments and is able to convey the meaning of entire source sentence to the target side.

In order to show the deep difference between VNMT and GroundHog, we further divide our test sets into 6 disjoint groups according to the length of source sentences. Figure 3 shows the BLEU scores of these 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. Overall, these obvious improvements on all groups in terms of the length of source sentences indicate that VNMT outperforms the vanilla NMT, no matter how long source sentences are.

4.3 Translation Analysis

We highlight relatively important part in red color.
volutional neural network to encode source sentences, and then use a recurrent neural network (RNN) to generate target translations. Cho et al. (2014) further propose the Encoder-Decoder framework with two RNNs. However, the above-mentioned work mainly focus on computing the score/probability of bilingual phrases, rather than entire translations.

With regard to NMT, 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 the 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., 2015b), 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), character-level NMT (Ling et al., 2015; Costa-Jussà and Fonollosa, 2016), the exploitation of monolingual corpora (Gulcehre et al., 2015; Sennrich et al., 2015a) and memory network (Meng et al., 2015). 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.

In this respect, 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 to ours 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-related tasks, the adaptation and utilization of variational neural model to 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 conventional 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 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, e.g., conversations.
References

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

[Bowman et al.2015] 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.

[Cheng et al.2015] 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.

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

[Chung et al.2015] 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.

[Costa-Jussà and Fonollosa2016] M. R. Costa-Jussà and J. A. R. Fonollosa. 2016. Character-based Neural Machine Translation. ArXiv e-prints, March.

[Feng et al.2016] 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.

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

[Gulcehre et al.2015] C. Gulcehre, O. Firat, K. Xu, K. Cho, L. Barrault, H.-C. Lin, F. Bougares, H. Schwenk, and Y. Bengio. 2015. On Using Monolingual Corpora in Neural Machine Translation. ArXiv e-prints, March.

[Jean et al.2015] 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.

[Kalchbrenner and Blunsom2013] Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In Proc. of EMNLP, pages 1700–1709, October.

[Kingma and Welling2014] Diederik P. Kingma and Max Welling. 2014. Auto-Encoding Variational Bayes. In Proc. of ICLR.

[Kingma et al.2014] 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.

[Koehn and Hoang2007] Philipp Koehn and Hieu Hoang. 2007. Factored translation models. In Proc. of EMNLP-CoNLL, pages 868–876, June.

[Koehn2004] Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In Proc. of EMNLP.

[Ling et al.2015] W. Ling, I. Trancoso, C. Dyer, and A. W Black. 2015. Character-based Neural Machine Translation. ArXiv e-prints, November.

[Luong et al.2015a] 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.

[Luong et al.2015b] 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.

[Meng et al.2015] 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.

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

[Papineni et al.2002] 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.

[Rezende et al.2014] 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.

[Sennrich et al.2015a] R. Sennrich, B. Haddow, and A. Birch. 2015a. Improving Neural Machine Translation Models with Monolingual Data. ArXiv e-prints, November.

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

[Shen et al.2015] 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.

[Sutskever et al.2014] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. CoRR, abs/1409.3215.
[Tu et al.2016] Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. 2016. Coverage-based neural machine translation. *CoRR*, abs/1601.04811.