NEURAL MACHINE TRANSLATION WITH LATENT SEMANTIC OF IMAGE AND TEXT

Joji Toyama*, **Masanori Misono***†, **Masahiro Suzuki, Kotaro Nakayama & Yutaka Matsuo**
Graduate School of Engineering, †Graduate School of Information Science and Technology
The University of Tokyo
Hongo, Tokyo, Japan
{toyama,misono,masa,k-nakayama,matsuo}@weblab.t.u-tokyo.ac.jp

ABSTRACT

Although attention-based Neural Machine Translation have achieved great success, attention-mechanism cannot capture the entire meaning of the source sentence because the attention mechanism generates a target word depending heavily on the relevant parts of the source sentence. The report of earlier studies has introduced a latent variable to capture the entire meaning of sentence and achieved improvement on attention-based Neural Machine Translation. We follow this approach and we believe that the capturing meaning of sentence benefits from image information because human beings understand the meaning of language not only from textual information but also from perceptual information such as that gained from vision. As described herein, we propose a neural machine translation model that introduces a continuous latent variable containing an underlying semantic extracted from texts and images. Our model, which can be trained end-to-end, requires image information only when training. Experiments conducted with an English–German translation task show that our model outperforms over the baseline.

1 INTRODUCTION

Neural machine translation (NMT) has achieved great success in recent years (Sutskever et al., 2014; Bahdanau et al., 2015). In contrast to statistical machine translation, which requires huge phrase and rule tables, NMT requires much less memory. However, the most standard model, NMT with attention (Bahdanau et al., 2015) entails the shortcoming that the attention mechanism cannot capture the entire meaning of a sentence because it generates a target word while depending heavily on the relevant parts of the source sentence (Tu et al., 2016). To overcome this problem, Variational Neural Machine Translation (VNMT), which outperforms NMT with attention introduces a latent variable to capture the underlying semantic from source and target (Zhang et al., 2016). We follow the motivation of VNMT, which is to capture underlying semantic of a source.

Image information is related to language. For example, we human beings understand the meaning of language by linking perceptual information given by the surrounding environment and language (Barsalou, 1999). Although it is natural and easy for humans, it is difficult for computers to understand different domain's information integrally. Solving this difficult task might, however, bring great improvements in natural language processing. Several researchers have attempted to link language and images such as image captioning by Xu et al. (2015) or image generation from sentences by Reed et al. (2016). They described the possibility of integral understanding of images and text. In machine translation, we can expect an improvement using not only text information but also image information because image information can bridge two languages.

As described herein, we propose the neural machine translation model which introduces a latent variable containing an underlying semantic extracted from texts and images. Our model includes an explicit latent variable $\mathbf{z}$, which has underlying semantics extracted from text and images by introducing a Variational Autoencoder (VAE) (Kingma et al., 2014; Rezende et al., 2014). Our model,

[*] First two authors contributed equally.

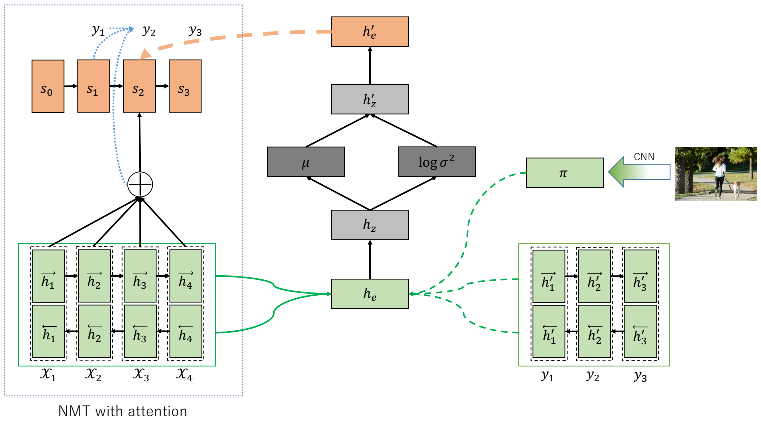

y_1
y_2
y_3
s_0
s_1
s_2
s_3
h'_e
h'_z
μ
$\log \sigma^2$
h_z
h_e
π
CNN
$\overrightarrow{h_1}$ $\overrightarrow{h_2}$ $\overrightarrow{h_3}$ $\overrightarrow{h_4}$
$\overleftarrow{h_1}$ $\overleftarrow{h_2}$ $\overleftarrow{h_3}$ $\overleftarrow{h_4}$
x_1 x_2 x_3 x_4
$\overrightarrow{h'_1}$ $\overrightarrow{h'_2}$ $\overrightarrow{h'_3}$
$\overleftarrow{h'_1}$ $\overleftarrow{h'_2}$ $\overleftarrow{h'_3}$
y_1 y_2 y_3
NMT with attention

Figure 1: Architecture of Proposed Model.
Green dotted lines denote that $\boldsymbol{\pi}$ and encoded $\mathbf{y}$ are used only when training.

which can be trained end-to-end, requires image information only when training. As described herein, we tackle the task with which one uses a parallel corpus and images in training, while using a source corpus in translating. It is important to define the task in this manner because we rarely have a corresponding image when we want to translate a sentence. During translation, our model generates a semantic variable $\mathbf{z}$ from a source, integrates variable $\mathbf{z}$ into a decoder of neural machine translation system, and then finally generates the translation. The difference between our model and VNMT is that we use image information in addition to text information.

For experiments, we used Multi30k (Elliott et al., 2016), which includes images and the corresponding parallel corpora of English and German. Our model outperforms the baseline with two evaluation metrics: METEOR (Denkowski & Lavie, 2014) and BLEU (Papineni et al., 2002). Moreover, we obtain some knowledge related to our model and Multi30k. Finally, we present some examples in which our model either improved, or worsened, the result.

Our paper contributes to the neural machine translation research community in three ways.

- We present the first neural machine translation model to introduce a latent variable inferred from image and text information. We also present the first translation task with which one uses a parallel corpus and images in training, while using a source corpus in translating.
- Our translation model can generate more accurate translation by training with images, especially for short sentences.
- We present how the translation of source is changed by adding image information compared to VNMT which does not use image information.

2 Background

Our model is the extension of Variational Neural Machine Translation (VNMT) (Zhang et al., 2016). Our model is also viewed as one of the multimodal translation models. In our model, VAE is used to introduce a latent variable. We describe the background of our model in this section.

2.1 Variational Neural Machine Translation

The VNMT translation model introduces a latent variable. This model's architecture shown in Figure 1 excludes the arrow from $\boldsymbol{\pi}$. This model involves three parts: encoder, inferrer, and decoder. In the encoder, both the source and target are encoded by bidirectional-Recurrent Neural Networks (bidirectional-RNN) and a semantic representation is generated. In the inferrer, a latent variable $\mathbf{z}$ is modeled from a semantic representation by introducing VAE. In the decoder, a latent variable $\mathbf{z}$ is integrated in the Gated Recurrent Unit (GRU) decoder; also, a translation is generated.

Our model is followed by architecture, except that the image is also encoded to obtain a latent variable $\mathbf{z}$.

2.2 Multimodal Translation

Multimodal Translation is the task with which one might one can use a parallel corpus and images. The first papers to study multimodal translation are Elliott et al. (2015) and Hitschler & Riezler (2016). It was selected as a shared task in Workshop of Machine Translation 2016 (WMT16[1]). Although several studies have been conducted (Caglayan et al., 2016; Huang et al., 2016; Calixto et al., 2016; Libovický et al., 2016; Rodríguez Guasch & Costa-jussà, 2016; Shah et al., 2016), they do not show great improvement, especially in neural machine translation (Specia et al., 2016). Here, we introduce end-to-end neural network translation models like our model.

Caglayan et al. (2016) integrate an image into an NMT decoder. They simply put source context vectors and image feature vectors extracted from ResNet-50's 'res4f_relu' layer (He et al., 2016) into the decoder called multimodal conditional GRU. They demonstrate that their method does not surpass the text-only baseline: NMT with attention.

Huang et al. (2016) integrate an image into a head of source words sequence. They extract prominent objects from the image by Region-based Convolutional Neural Networks (R-CNN) (Girshick, 2015). Objects are then converted to feature vectors by VGG-19 (Simonyan & Zisserman, 2014) and are put into a head of source words sequence. They demonstrate that object extraction by R-CNN contributes greatly to the improvement. This model achieved the highest METEOR score in NMT-based models in WMT16, which we compare to our model in the experiment. We designate this model as CMU.

Caglayan et al. (2016) argue that their proposed model did not achieve improvement because they failed to benefit from both text and images. We assume that they failed to integrate text and images because they simply put images and text into neural machine translation despite huge gap exists between image information and text information. Our model, however, presents the possibility of benefitting from images and text because text and images are projected to their common semantic space so that the gap of images and text would be filled.

2.3 Variational Auto Encoder

VAE was proposed in an earlier report of the literature Kingma et al. (2014); Rezende et al. (2014). Given an observed variable $\mathbf{x}$, VAE introduces a continuous latent variable $\mathbf{z}$, with the assumption that $\mathbf{x}$ is generated from $\mathbf{z}$. VAE incorporates $p_\theta(\mathbf{x}|\mathbf{z})$ and $q_\phi(\mathbf{z}|\mathbf{x})$ into an end-to-end neural network. The lower bound is shown below.

$$\mathcal{L}_{\text{VAE}} = -\mathrm{D}_{\text{KL}}\left[q_\phi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z})\right] + \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x}|\mathbf{z})\right] \leq \log p_\theta(\mathbf{x}) \tag{1}$$

3 Neural Machine Translation With Latent Semantic Of Image And Text

We propose a neural machine translation model which explicitly has a latent variable containing an underlying semantic extracted from both text and image. This model can be seen as an extension of VNMT by adding image information.

Our model can be drawn as a graphical model in Figure 3. Its lower bound is

$$\mathcal{L} = -\mathrm{D}_{\text{KL}}\left[q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y},\boldsymbol{\pi})||p_\theta(\mathbf{z}|\mathbf{x})\right] + \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y},\boldsymbol{\pi})}\left[\log p_\theta(\mathbf{y}|\mathbf{z},\mathbf{x})\right], \tag{2}$$

where $\mathbf{x}, \mathbf{y}, \boldsymbol{\pi}, \mathbf{z}$ respectively denote the source, target, image and latent variable, and p_θ and q_ϕ respectively denote the prior distribution and the approximate posterior distribution. It is noteworthy in Eq. (2) that we want to model $p(\mathbf{z}|\mathbf{x},\mathbf{y},\boldsymbol{\pi})$, which is intractable. Therefore we model $q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y},\boldsymbol{\pi})$

[1] http://www.statmt.org/wmt16/

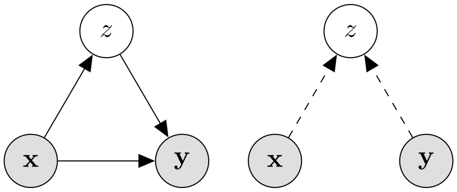


Figure 2: VNMT

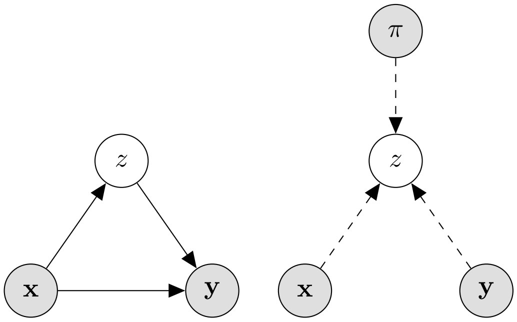


Figure 3: Our model

instead, and also model prior $p_\theta(\mathbf{z}|\mathbf{x})$ so that we can generate a translation from the source in testing. Derivation of the formula is presented in the appendix.

We model all distributions in Eq. (2) by neural networks. Our model architecture is divisible into three parts: 1) encoder, 2) inferrer, and 3) decoder.

3.1 Encoder

In the encoder, the semantic representation $\mathbf{h_e}$ is obtained from the image, source, and target. We propose several methods to encode an image. We show how these methods affect the translation result in the Experiment section. This representation is used in the inferrer. This section links to the green part of Figure 1.

3.1.1 Text encoding

The source and target are encoded in the same way as Bahdanau et al. (2015). The source is converted to a sequence of 1-of-k vector and is embedded to d_{emb} dimensions. We designate it as the source sequence. Then, a source sequence is put into bidirectional RNN. Representation $\mathbf{h}_i$ is obtained by concatenating $\overrightarrow{\mathbf{h}}_i$ and $\overleftarrow{\mathbf{h}}_i$: $\overrightarrow{\mathbf{h}}_i = \mathrm{RNN}(\overrightarrow{\mathbf{h}}_{i-1}, E_{w_i})$, $\overleftarrow{\mathbf{h}}_i = \mathrm{RNN}(\overleftarrow{\mathbf{h}}_{i+1}, E_{w_i})$, $\mathbf{h}_i = [\overrightarrow{\mathbf{h}}_i; \overleftarrow{\mathbf{h}}_i]$, where E_{w_i} is the embedded word in a source sentence, $\mathbf{h}_i \in \mathbb{R}^{d_h}$, and $\overrightarrow{\mathbf{h}}_i, \overleftarrow{\mathbf{h}}_i \in \mathbb{R}^{\frac{d_h}{2}}$. It is conducted through $i = 0$ to $i = T_f$, where T_f is the sequence length. GRU is implemented in bidirectional RNN so that it can attain long-term dependence. Finally, we conduct mean-pooling to $\mathbf{h}_i$ and obtain the source representation vector as $\mathbf{h}_f = \frac{1}{T_f}\sum_i^{T_f} \mathbf{h}_i$. The exact same process is applied to target to obtain target representation $\mathbf{h}_g$.

3.1.2 Image encoding and semantic representation

We use Convolutional Neural Networks (CNN) to extract feature vectors from images. We propose several ways of extracting image features.

Global (G) The image feature vector is extracted from the image using a CNN. With this method, we use a feature vector in the certain layer as $\boldsymbol{\pi}$. Then $\boldsymbol{\pi}$ is encoded to the image representation vector $\mathbf{h}_\pi$ simply by affine transformation as

$$\mathbf{h}_\pi = W_\pi \boldsymbol{\pi} + b_\pi \quad \text{where } W_\pi \in \mathbb{R}^{d_\pi \times d_{fc7}}\ ,\ b_\pi \in \mathbb{R}^{d_\pi}. \tag{3}$$

Global and Objects (G+O) First we extract some prominent objects from images in some way. Then, we obtain fc7 image feature vectors $\boldsymbol{\pi}$ from the original image and extracted objects using a CNN. Therefore $\boldsymbol{\pi}$ takes a variable length. We handle $\boldsymbol{\pi}$ in two ways: average and RNN encoder.

In average (**G+O-AVG**), we first obtain intermediate image representation vector $\mathbf{h}'_\pi$ by affine transformation in Eq. (3). Then, the average of $\mathbf{h}'_\pi$ becomes the image representation vector: $\mathbf{h}_\pi = \frac{\sum_i^l \mathbf{h}'_{\pi_i}}{l}$, where l is the length of $\mathbf{h}'_\pi$.

In RNN encoder (**G+O-RNN**), we first obtain $\mathbf{h}'_\pi$ by affine transformation in Eq. (3). Then, we encode $\mathbf{h}'_\pi$ in the same way as we encode text in Section 3.1.1 to obtain $\mathbf{h}_\pi$.

Global and Objects into source and target (G+O-TXT) Thereby, we first obtain $\mathbf{h}'_\pi$ by affine transformation in Eq. (3). Then, we put sequential vector $\mathbf{h}'_\pi$ into the head of the source sequence and target sequence. In this case, we set d_π to be the same dimension as d_{emb}. In fact, the source sequence including $\mathbf{h}'_\pi$ is only used to model $q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y}, \boldsymbol{\pi})$. Context vector $\mathbf{c}$ (Eq. (15)) and $p_\theta(\mathbf{z}|\mathbf{x})$ are computed by a source sequence that does not include $\mathbf{h}'_\pi$. We encode the source sequence including $\mathbf{h}'_\pi$ as Section 3.1.1 to obtain $\mathbf{h}_f$ and $\mathbf{h}_g$. In this case, $\mathbf{h}_\pi$ is not obtained. Image information is contained in $\mathbf{h}_f$ and $\mathbf{h}_g$.

All representation vectors $\mathbf{h}_f$, $\mathbf{h}_g$ and $\mathbf{h}_\pi$ are concatenated to obtain a semantic representation vector as $\mathbf{h}_e = [\mathbf{h}_f; \mathbf{h}_g; \mathbf{h}_\pi]$, where $\mathbf{h}_e \in \mathbb{R}^{d_e = 2\times d_h + d_\pi}$ (in G+O-TXT: $\mathbf{h}_e = [\mathbf{h}_f; \mathbf{h}_g]$, where $\mathbf{h}_e \in \mathbb{R}^{d_e = 2\times d_h}$). It is an input of the multimodal variational neural inferrer.

3.2 Inferrer

We model the posterior $q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y}, \boldsymbol{\pi})$ using a neural network and also the prior $p_\theta(\mathbf{z}|\mathbf{x})$ by neural network. This section links to the black and grey part of Figure 1.

3.2.1 Neural Posterior Approximator

Modeling the true posterior $p_\theta(\mathbf{z}|\mathbf{x}, \mathbf{y}, \boldsymbol{\pi})$ is usually intractable. Therefore, we consider modeling of an approximate posterior $q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y}, \boldsymbol{\pi})$ by introducing VAE. We assume that the posterior $q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y}, \boldsymbol{\pi})$ has the following form:

$$q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y}, \boldsymbol{\pi}) = \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}(\mathbf{x}, \mathbf{y}, \boldsymbol{\pi}), \boldsymbol{\sigma}(\mathbf{x}, \mathbf{y}, \boldsymbol{\pi})^2\mathbf{I}). \tag{4}$$

The mean $\boldsymbol{\mu}$ and standard deviation $\boldsymbol{\sigma}$ of the approximate posterior are the outputs of neural networks.

Starting from the variational neural encoder, a semantic representation vector $\mathbf{h}_e$ is projected to latent semantic space as

$$\mathbf{h}_z = g(W_z^{(1)}\mathbf{h}_e + \mathbf{b}_z^{(1)}), \tag{5}$$

where $W_z^{(1)} \in \mathbb{R}^{d_z \times (d_e)}$ $\mathbf{b}_z^{(1)} \in \mathbb{R}^{d_z}$. $g(\cdot)$ is an element-wise activation function, which we set as $\tanh(\cdot)$. Gaussian parameters of Eq. (4) are obtained through linear regression as

$$\boldsymbol{\mu} = W_\mu \mathbf{h}_z + \mathbf{b}_\mu, \log \boldsymbol{\sigma}^2 = W_\sigma \mathbf{h}_z + \mathbf{b}_\sigma, \tag{6}$$

where $\boldsymbol{\mu}, \log \boldsymbol{\sigma}^2 \in \mathbb{R}^{d_z}$.

3.2.2 Neural Prior Model

We model the prior distribution $p_\theta(\mathbf{z}|\mathbf{x})$ as follows:

$$p_\theta(\mathbf{z}|\mathbf{x}) \quad = \quad \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}'(\mathbf{x}), \boldsymbol{\sigma}'(\mathbf{x})^2\mathbf{I}). \tag{7}$$

$\boldsymbol{\mu}'$ and $\boldsymbol{\sigma}'$ are generated in the same way as that presented in Section 3.2.1, except for the absence of $\mathbf{y}$ and $\boldsymbol{\pi}$ as inputs. Because of the absence of representation vectors, the dimensions of weight in equation (5) for prior model are $W_z^{'(1)} \in \mathbb{R}^{d_z \times d_h}$, $\mathbf{b}_z^{'(1)} \in \mathbb{R}^{d_z}$. We use a reparameterization trick to obtain a representation of latent variable $\mathbf{z}$: $\mathbf{h}'_z = \boldsymbol{\mu} + \boldsymbol{\sigma}\boldsymbol{\epsilon}$, $\boldsymbol{\epsilon} \sim \mathcal{N}(0, I)$. During translation, $\mathbf{h}'_z$ is set as the mean of $p_\theta(\mathbf{z}|\mathbf{x})$. Then, $\mathbf{h}'_z$ is projected onto the target space as

$$\mathbf{h}'_e = g(W_z^{(2)}\mathbf{h}'_z + \mathbf{b}_z^{(2)}) \quad \text{where } \mathbf{h}'_e \in \mathbb{R}^{d_e}. \tag{8}$$

$\mathbf{h}'_e$ is then integrated into the neural machine translation's decoder.

3.3 DECODER

This section links to the orange part of Figure 1. Given the source sentence $\mathbf{x}$ and the latent variable $\mathbf{z}$, decoder defines the probability over translation $\mathbf{y}$ as

$$p(\mathbf{y}|\mathbf{z},\mathbf{x}) = \prod_{j=1}^{T} p(\mathbf{y}_j|\mathbf{y}_{<j},\mathbf{z},\mathbf{x}). \tag{9}$$

How we define the probability over translation $\mathbf{y}$ is fundamentally the same as VNMT, except for using conditional GRU instead of GRU. Conditional GRU involves two GRUs and an attention mechanism. We integrate a latent variable $\mathbf{z}$ into the second GRU. We describe it in the appendix.

3.4 MODEL TRAINING

Monte Carlo sampling method is used to approximate the expectation over the posterior Eq. (2), $\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y},\boldsymbol{\pi})} \approx \frac{1}{L}\sum_{l=1}^{L}\log p_\theta(\mathbf{y}|\mathbf{x},\mathbf{h}_z^{(l)})$, where L is the number of samplings. The training objective is defined as

$$\mathcal{L}(\theta,\phi) = -\mathrm{D}_{\mathrm{KL}}\left[q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y},\boldsymbol{\pi})||p_\theta(\mathbf{z}|\mathbf{x})\right] + \frac{1}{L}\sum_{l=1}^{L}\sum_{j=1}^{T}\log p_\theta(\mathbf{y}_j|\mathbf{y}_{<j},\mathbf{x},\mathbf{h}_z^{(l)}), \tag{10}$$

where $\mathbf{h}_z = \boldsymbol{\mu} + \boldsymbol{\sigma}\cdot\boldsymbol{\epsilon}$, $\boldsymbol{\epsilon} \sim \mathcal{N}(0, I)$. The first term, KL divergence, can be computed analytically and is differentiable because both distributions are Gaussian. The second term is also differentiable. We set L as 1. Overall, the objective $\mathcal{L}$ is differentiable. Therefore, we can optimize the parameter θ and variational parameter ϕ using gradient ascent techniques.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

We used Multi30k (Elliott et al., 2016) as the dataset. Multi30k have an English description and a German description for each corresponding image. We handle 29,000 pairs as training data, 1,014 pairs as validation data, and 1,000 pairs as test data.

Before training, punctuation normalization and lowercase are applied to both English and German sentences by Moses (Koehn et al., 2007) scripts[2]. Compound-word splitting is conducted only to German sentences using Sennrich et al. (2016)[3]. Then we tokenize sentences[2] and use them as training data. We produce vocabulary dictionaries from training data. The vocabulary becomes 10,211 words for English and 13,180 words for German after compound-word splitting.

Image features are extracted using VGG-19 CNN (Simonyan & Zisserman, 2014). We use 4096-dimensional fc7 features. To extract the object's region, we use Fast R-CNN (Girshick, 2015). Fast R-CNN is trained on ImageNet and MSCOCO dataset [4].

All weights are initialized by $\mathcal{N}(0, 0.01\mathbf{I})$. We use the adadelta algorithm as an optimization method. The hyperparameters used in the experiment are presented in the Appendix. All models are trained with early stopping. When training, VNMT is fine-tuned by NMT model and our models are fine-tuned using VNMT. When translating, we use beam-search. The beam-size is set as 12. Before evaluation, we restore split words to the original state and de-tokenize[2] generated sentences.

We implemented proposed models based on *dl4mt*[5]. Actually, *dl4mt* is fundamentally the same model as Bahdanau et al. (2015), except that its decoder employs conditional GRU[6]. We implemented VNMT also with conditional GRU so small difference exists between our implementation

[2] https://github.com/moses-smt/mosesdecoder/blob/master/scripts/tokenizer/{normalize-punctuation, lowercase, tokenizer, detokenizer}.perl

[3] https://github.com/rsennrich/subword-nmt

[4] https://github.com/rbgirshick/fast-rcnn/tree/coco

[5] https://github.com/nyu-dl/dl4mt-tutorial

[6] The architecture is described at https://github.com/nyu-dl/dl4mt-tutorial/blob/master/docs/cgru.pdf

and originally proposed VNMT which employs normal GRU as a decoder. We evaluated results based on METEOR and BLUE using MultEval [7].

4.2 Result

Table 1 presents experiment results. It shows that our models outperforms the baseline in both METEOR and BLEU. Figure 4 shows the plot of METEOR score of baselines and our models models in validation. Figure 5 shows the plot of METEOR score and the source sentence length.

Table 1: Evaluation Result on Multi30k dataset (English–German). The scores in parentheses are computed with '-norm' parameter. NMT is *dl4mt*'s NMT (in the *session3* directory). The score of the CMU is from (Huang et al., 2016).

		METEOR ↑		BLEU ↑	
		val	test	val	test
	NMT	51.5 (55.8)	50.5 (54.9)	35.8	33.1
	VNMT	52.2 (56.3)	51.1 (55.3)	**37.0**	34.9
	CMU	- (-)	- (54.1)	-	-
Our Model	G	50.6 (54.8)	**52.4 (56.0)**	34.5	**36.5**
	G+O-AVG	51.8 (55.8)	51.8 (55.8)	35.7	35.8
	G+O-RNN	51.8 (56.1)	51.0 (55.4)	35.9	34.9
	G+O-TXT	**52.6 (56.8)**	51.7 (56.0)	36.6	35.1

4.3 Quantitative Analysis

Table 1 shows that G scores the best in proposed models. In G, we simply put the feature of the original image. Actually, proposed model does not benefit from R-CNN, presumably because we can not handle sequences of image features very well. For example, G+O-AVG uses the average of multiple image features, but it only makes the original image information unnecessarily confusing.

Figure 4 shows that G and G+O-AVG outperforms VNMT almost every time, but all model scores increase suddenly in the 17,000 iteration validation. We have no explanation for this behavior. Figure 4 also shows that G and G+O-AVG scores fluctuate more moderately than others. We state that G and G+O-AVG gain stability by adding image information. When one observes the difference between the test score and the validation score for each model, baseline scores decrease more than proposed model scores. Especially, the G score increases in the test, simply because proposed models produce a better METEOR score on average, as shown in Figure 4.

Figure 5 shows that G and G+O-AVG make more improvements on baselines in short sentences than in long sentences, presumably because $q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y}, \boldsymbol{\pi})$ can model $\mathbf{z}$ well when a sentence is short. Image features always have the same dimension, but underlying semantics of the image and text differ. We infer that when the sentence is short, image feature representation can afford to approximate the underlying semantic, but when a sentence is long, image feature representation can not approximate the underlying semantic.

Multi30k easily becomes overfitted, as shown in Figure 8 and 9 in the appendix. This is presumably because 1) Multi30k is the descriptions of image, making the sentences short and simple, and 2) Multi30k has 29,000 sentences, which could be insufficient. In the appendix, we show how the parameter setting affects the score. One can see that decay-c has a strong effect. Huang et al. (2016) states that their proposed model outperforms the baseline (NMT), but we do not have that observation. It can be assumed that their baseline parameters are not well tuned.

4.4 Qualitative Analysis

We presented the top 30 sentences, which make the largest METEOR score difference between G and VNMT, to native speakers of German and get the overall comments. They were not informed of

[7] https://github.com/jhclark/multeval, we use meteor1.5 instead of meteor1.4, which is the default of MultEval.

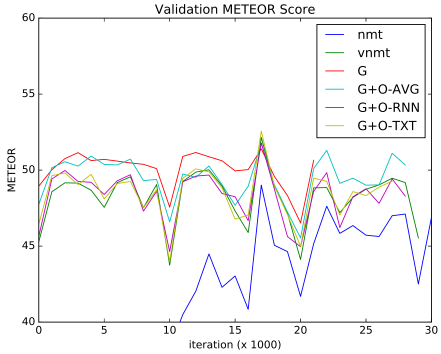


Figure 4: METEOR score to the validation data which are calculated for each 1000 iterations.

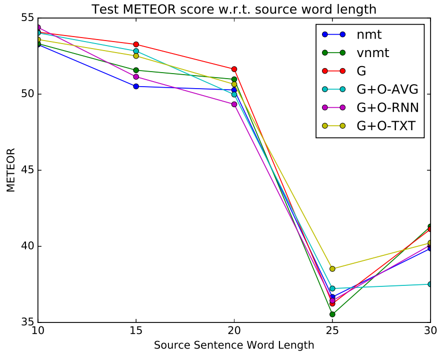
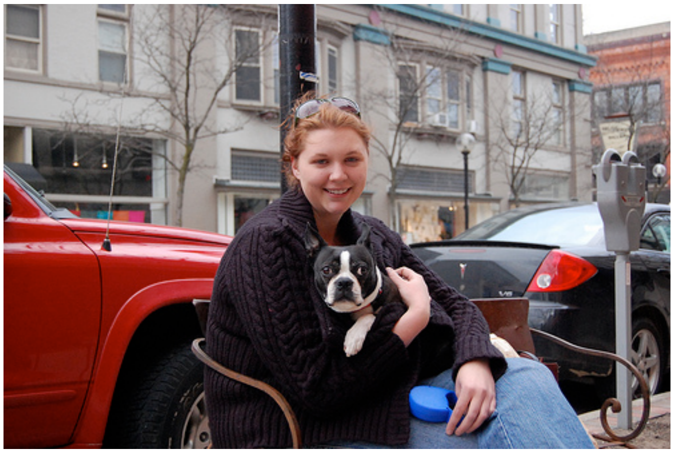


Figure 5: METEOR score on different groups of the source sentence length.

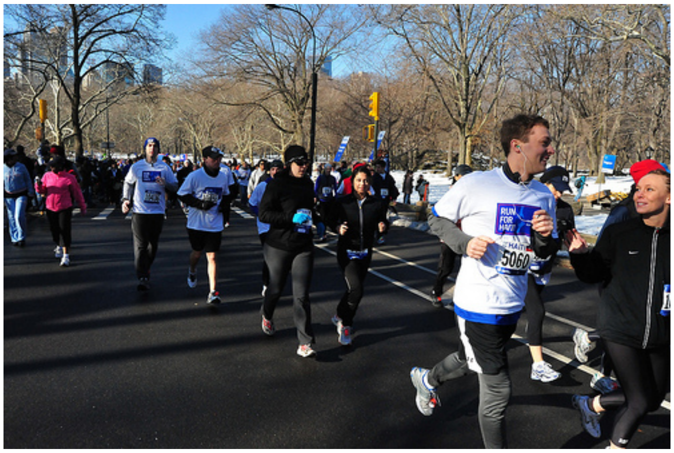
our model training with image in addition to text. These comments are summarized into two general remarks. One is that G translates the meaning of the source material more accurately than VNMT. The other is that our model has more grammatical errors as prepositions' mistakes or missing verbs compared to VNMT. We assume these two remarks are reasonable because G is trained with images which mainly have a representation of noun rather than verb, therefore can capture the meaning of materials in sentence.

Figure 6 presents the translation results and the corresponding image which G translates more accurately than VNMT in METEOR. Figure 7 presents the translation results and the corresponding image which G translates less accurately than VNMT in METEOR. Again, we note that our model does not use image during translating. In Figure 6, G translates ”a white and black dog” correctly while VNMT translates it incorrectly implying ”a white dog and a black dog”. We assume that G correctly translates the source because G captures the meaning of material in the source. In Figure 7, G incorrectly translates the source. Its translation result is missing the preposition meaning ”at”, which is hardly represented in image.We present more translation examples in appendix.



Source	a woman holding a white and black dog.
Truth	eine frau hält einen weiß-schwarzen hund.
VNMT	eine frau hält einen weißen und schwarzen hund.
Our Model (G)	eine frau hält einen weiß-schwarzen hund.

Figure 6: Translation 1



Source	a group of people running a marathon in the winter.
Truth	eine gruppe von menschen läuft bei einem marathon im winter.
VNMT	eine gruppe von menschen läuft bei einem marathon im winter.
Our Model (G)	eine gruppe leute läuft einen marathon im winter an.

Figure 7: Translation 2

5 Conclusion

As described herein, we proposed the neural machine translation model that explicitly has a latent variable that includes underlying semantics extracted from both text and images. Our model outperforms the baseline in both METEOR and BLEU scores. Experiments and analysis present that our model can generate more accurate translation for short sentences. In qualitative analysis, we present that our model can translate nouns accurately while our model make grammatical errors.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *ICLR*, 2015.

Lawrence W. Barsalou. Perceptual symbol Systems. *Behavioral and Brain Sciences*, 22:577–609, 1999.

Ozan Caglayan, Walid Aransa, Yaxing Wang, Marc Masana, Mercedes García-Martínez, Fethi Bougares, Loïc Barrault, and Joost van de Weijer. Does Multimodality Help Human and Machine for Translation and Image Captioning? In *WMT*, 2016.

Iacer Calixto, Desmond Elliott, and Stella Frank. DCU-UvA Multimodal MT System Report. In *Proceedings of the First Conference on Machine Translation*, pp. 634–638. Association for Computational Linguistics, 2016.

Michael Denkowski and Alon Lavie. Meteor Universal: Language Specific Translation Evaluation for Any Target Language. In *Proceedings of the EACL 2014 Workshop on Statistical Machine Translation*, 2014.

D. Elliott, S. Frank, and E. Hasler. Multilingual Image Description with Neural Sequence Models. *ArXiv e-prints*, 2015.

Desmond Elliott, Stella Frank, Khalil Sima'an, and Lucia Specia. Multi30K: Multilingual English-German Image Descriptions. *CoRR*, abs/1605.00459, 2016.

Ross Girshick. Fast R-CNN. In *ICCV*, 2015.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *CVPR*, 2016.

Julian Hitschler and Stefan Riezler. Multimodal Pivots for Image Caption Translation. *arXiv preprint arXiv:1601.03916*, 2016.

Po-Yao Huang, Frederick Liu, Sz-Rung Shiang, Jean Oh, and Chris Dyer. Attention-based Multimodal Neural Machine Translation. In *WMT*, 2016.

Diederik P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised Learning with Deep Generative Models. In *NIPS*, 2014.

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. Moses: Open Source Toolkit for Statistical Machine Translation. In *ACL*, 2007.

Jindřich Libovický, Jindřich Helcl, Marek Tlustý, Ondřej Bojar, and Pavel Pecina. CUNI System for WMT16 Automatic Post-Editing and Multimodal Translation Tasks. In *Proceedings of the First Conference on Machine Translation*, pp. 646–654. Association for Computational Linguistics, 2016.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: A Method for Automatic Evaluation of Machine Translation. In *ACL*, 2002.

Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative Adversarial Text to Image Synthesis. In *ICML*, 2016.

Danilo J. Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic Backpropagation and Approximate Inference in Deep Generative Models. In *ICML*, 2014.

Sergio Rodríguez Guasch and Marta R. Costa-jussà. WMT 2016 Multimodal Translation System Description based on Bidirectional Recurrent Neural Networks with Double-Embeddings. In *Proceedings of the First Conference on Machine Translation*, pp. 655–659. Association for Computational Linguistics, 2016.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural Machine Translation of Rare Words with Subword Units. In *ACL*, 2016.

Kashif Shah, Josiah Wang, and Lucia Specia. SHEF-Multimodal: Grounding Machine Translation on Images. In *Proceedings of the First Conference on Machine Translation*, pp. 660–665. Association for Computational Linguistics, 2016.

Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR*, abs/1409.1556, 2014.

Lucia Specia, Stella Frank, Khalil Sima ’ an, and Desmond Elliott. A shared Task on Multimodal Machine Translation and Crosslingual Image Description. In *Proceedings of the First Conference on Machine Translation, Berlin, Germany. Association for Computational Linguistics*, 2016.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to Sequence Learning with Neural Networks. In *NIPS*, 2014.

Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. Modeling Coverage for Neural Machine Translation. In *ACL*, 2016.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard S Zemel, and Yoshua Bengio. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In *CVPR*, 2015.

Biao Zhang, Deyi Xiong, and Jinsong Su. Variational Neural Machine Translation. In *EMNLP*, 2016.

A Derivation of Lower bounds

The lower bound of our model can be derived as follows:

$$\begin{aligned} p(\mathbf{y}|\mathbf{x}) &= \int p(\mathbf{y},\mathbf{z}|\mathbf{x})d\mathbf{z} \\ &= \int p(\mathbf{z}|\mathbf{x})p(\mathbf{y}|\mathbf{z},\mathbf{x})d\mathbf{z} \end{aligned}$$

$$\begin{aligned} \log p(\mathbf{y}|\mathbf{x}) &= \log \int q(\mathbf{z}|\mathbf{x},\mathbf{y},\boldsymbol{\pi})\frac{p(\mathbf{z}|\mathbf{x})p(\mathbf{y}|\mathbf{z},\mathbf{x})}{q(\mathbf{z}|\mathbf{x},\mathbf{y},\boldsymbol{\pi})}d\mathbf{z} \\ &\geq \int q(\mathbf{z}|\mathbf{x},\mathbf{y},\boldsymbol{\pi})\log\frac{p(\mathbf{z}|\mathbf{x})p(\mathbf{y}|\mathbf{z},\mathbf{x})}{q(\mathbf{z}|\mathbf{x},\mathbf{y},\boldsymbol{\pi})}d\mathbf{z} \\ &= \int q(\mathbf{z}|\mathbf{x},\mathbf{y},\boldsymbol{\pi})\left(\log\frac{p(\mathbf{z}|\mathbf{x})}{q(\mathbf{z}|\mathbf{x},\mathbf{y})}+\log p(\mathbf{y}|\mathbf{z},\mathbf{x})\right)d\mathbf{z} \\ &= -\mathrm{D}_{\mathrm{KL}}\left[q(\mathbf{z}|\mathbf{x},\mathbf{y},\boldsymbol{\pi})||p(\mathbf{z}|\mathbf{x})\right]+\mathbb{E}_{q(\mathbf{z}|\mathbf{x},\mathbf{y},\boldsymbol{\pi})}\left[\log p(\mathbf{y}|\mathbf{z},\mathbf{x})\right] \\ &= \mathcal{L} \end{aligned}$$

B Conditional GRU

Conditional GRU is implemented in *dl4mt*. Caglayan et al. (2016) extends Conditional GRU to make it capable of receiving image information as input. The first GRU computes intermediate representation s'_j as

$$\mathbf{s}'_j = (1-\mathbf{o}'_j)\odot\underline{\mathbf{s}}'_j + \mathbf{o}'_j\odot\mathbf{s}_{j-1} \tag{11}$$

$$\underline{\mathbf{s}}'_j = \tanh(W'E\left[\mathbf{y}_{j-1}\right]+\mathbf{r}'_j\odot(U'\mathbf{s}_{j-1})) \tag{12}$$

$$\mathbf{r}'_j = \sigma(W'_rE\left[\mathbf{y}_{j-1}\right]+U'_r\mathbf{s}_{j-1}) \tag{13}$$

$$\mathbf{o}'_j = \sigma(W'_oE\left[\mathbf{y}_{j-1}\right]+U'_o\mathbf{s}_{j-1}) \tag{14}$$

where $E\in\mathbb{R}^{d_{emb}\times d_t}$ signifies the target word embedding, $\underline{\mathbf{s}}'_j\in\mathbb{R}^{d_h}$ denotes the hidden state, $\mathbf{r}'_j\in\mathbb{R}^{d_h}$ and $\mathbf{o}'_j\in\mathbb{R}^{d_h}$ respectively represent the reset and update gate activations. d_t stands for the dimension of target; the unique number of target words. $[W',W'_r,W'_o]\in\mathbb{R}^{d_h\times d_{emb}}$, $[U',U'_r,U'_o]\in\mathbb{R}^{d_h\times d_h}$ are the parameters to be learned.

Context vector $\mathbf{c}_j$ is obtained as

$$\mathbf{c}_j = \tanh\left(\sum_{i=1}^{T_f}\alpha_{ij}\mathbf{h}_i\right) \tag{15}$$

$$\alpha_{ij} = \frac{\exp(\mathbf{e}_{ij})}{\sum_{k=1}^{T_f}\exp(e_{kj})} \tag{16}$$

$$\mathbf{e}_{ij} = U_{att}\tanh(W_{catt}\mathbf{h}_i + W_{att}\mathbf{s}'_j) \tag{17}$$

where $[U_{att},W_{catt},W_{att}]\in\mathbb{R}^{d_h\times d_h}$ are the parameters to be learned.

The second GRU computes $\mathbf{s}_j$ from $\mathbf{s}'_j$, $\mathbf{c}_j$ and $\mathbf{h}'_e$ as

$$\mathbf{s}_j = (1-\mathbf{o}'_j)\odot\underline{\mathbf{s}}_j + \mathbf{o}_j\odot\mathbf{s}'_j \tag{18}$$

$$\underline{\mathbf{s}}_j = \tanh(W\mathbf{c}_j + \mathbf{r}_j\odot(U\mathbf{s}'_j) + V\mathbf{h}'_e) \tag{19}$$

$$\mathbf{r}_j = \sigma(W_r\mathbf{c}_j + U_r\mathbf{s}'_j + V_r\mathbf{h}'_e) \tag{20}$$

$$\mathbf{o}_j = \sigma(W_o\mathbf{c}_j + U_o\mathbf{s}'_j + V_o\mathbf{h}'_e) \tag{21}$$

where $\underline{\mathbf{s}}_j\in\mathbb{R}^{d_h}$ stands for the hidden state, $\mathbf{r}_j\in\mathbb{R}^{d_h}$ and $\mathbf{o}_j\in\mathbb{R}^{d_h}$ are the reset and update gate activations. $[W,W_r,W_o]\in\mathbb{R}^{d_h\times d_h}$, $[U,U_r,U_o]\in\mathbb{R}^{d_h\times d_h}$, $[V,V_r,V_o]\in\mathbb{R}^{d_h\times d_z}$ are the parameters to be learned. We introduce $\mathbf{h}'_e$ obtained from a latent variable here so that a latent variable can affect the representation $\mathbf{s}_j$ through GRU units.

Finally, the probability of y is computed as

$$\mathbf{u}_j = L_u \tanh(E\left[\mathbf{y}_{j-1}\right] + L_s\mathbf{s}_j + L_x\mathbf{c}_j) \tag{22}$$

$$P(\mathbf{y}_j|\mathbf{y}_{j-1}, \mathbf{s}_j, \mathbf{c}_j) = \mathrm{Softmax}(\mathbf{u}_j) \tag{23}$$

where $L_u \in \mathbb{R}^{d_t \times d_{emb}}$, $L_s \in \mathbb{R}^{d_{emb} \times d_h}$ and $L_c \in \mathbb{R}^{d_{emb} \times d_h}$ are the parameters to be learned.

C Training Detail

C.1 Hyperparameters

Table 2 presents parameters that we use in the experiments.

Table 2: Hyperparameters. The name is the variable name of *dl4mt* except for *dimv* and *dim_pic*, which are the dimension of the latent variables and image embeddings. We set *dim* (number of LSTM unit size) and *dim_word* (dimensions of word embeddings) 256, *batchsize* 32, *maxlen* (max output length) 50 and *lr* (learning rate) 1.0 for all models. *decay-c* is weights on L2 regularization.

		dimv	*dim_pic*	*decay-c*
	NMT	-	256	0.001
	VNMT	256	256	0.0005
Our Model	G	256	512	0.001
	G+O-AVG	256	256	0.0005
	G+O-RNN	256	256	0.0005
	G+O-TXT	256	256	0.0005

We found that Multi30k dataset is easy to overfit. Figure 8 and Figure 9 present training cost and validation METEOR score graph of the two experimental settings of the NMT model. Table 3 presents the hyperparameters which were used in the experiments. Large *decay-c* ans small *batchsize* give the better METEOR scores in the end. Training is stopped if there is no validation cost improvements over the last 10 validations.

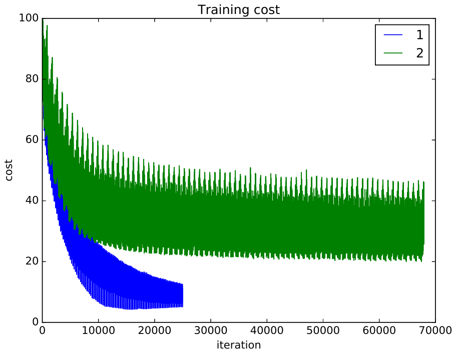


Figure 8: NMT Training Cost

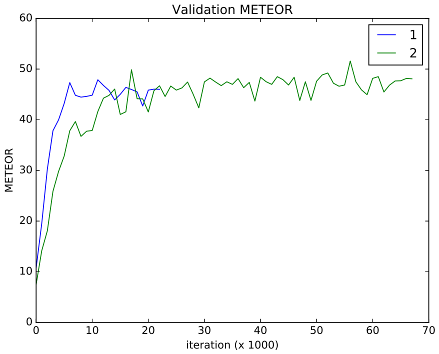


Figure 9: NMT Validation METEOR score

Table 3: Hyperparameters using the experiments in the Figure 8 and 9

	dim	*dim_word*	*lr*	*decay-c*	*maxlen*	*batchsie*
1	256	256	1.0	0.0005	30	128
2	256	256	1.0	0.001	50	32

Figure 10 presents the English word length histogram of the Multi30k test dataset. Most sentences in the Multi30k are less than 20 words. We assume that this is one of the reasons why Multi30k is easy to overfit.

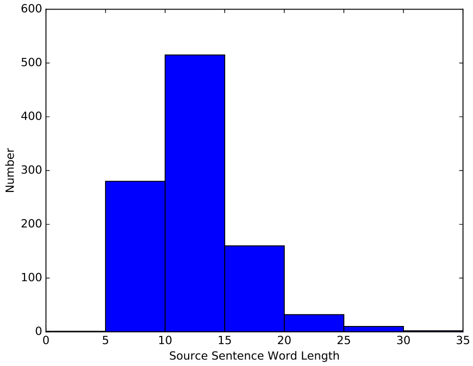
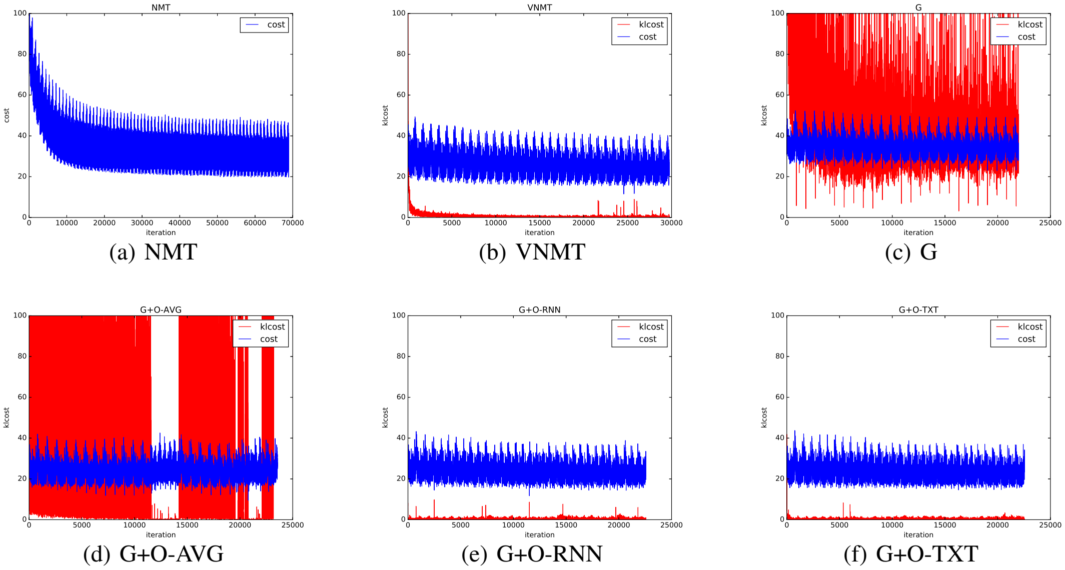


Figure 10: Word Length Histogram of the Multi30k Test Dataset

C.2 COST GRAPH

Figure 11 and 12 present the training cost and validation cost graph of each models. Please note that VNMT fine-tuned NMT, and other models fine-tuned VNMT.



(a) NMT

(b) VNMT

(c) G

(d) G+O-AVG

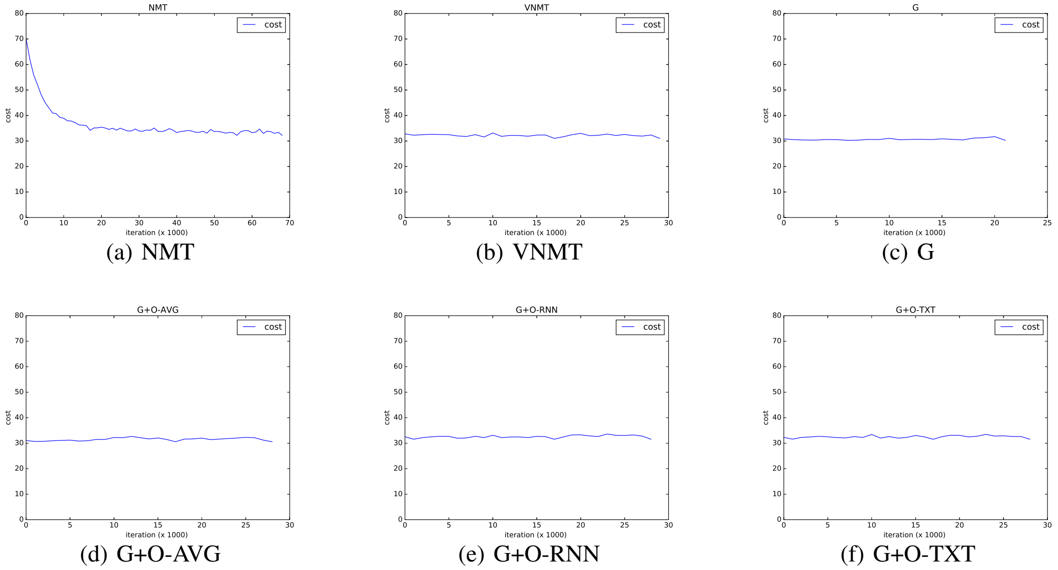
(e) G+O-RNN

(f) G+O-TXT

Figure 11: Training cost

C.3 TRANSLATION EXAMPLES

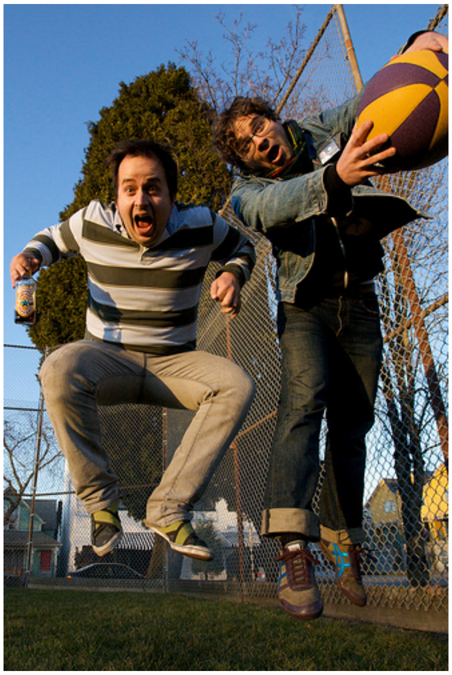
We present some selected translations from VNMT and our proposed model (G). As of translation 3 to 5 our model give the better METEOR scores than VNMT and as of translation 6 to 8 VNMT give the better METEOR scores than our models.



(a) NMT (b) VNMT (c) G

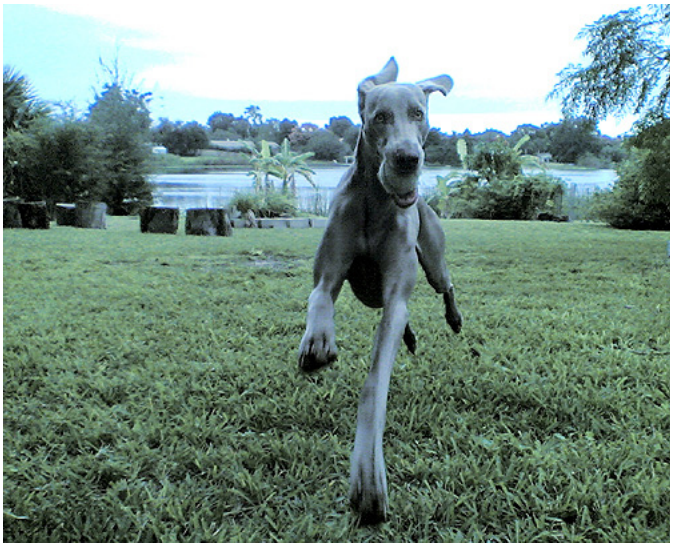
(d) G+O-AVG (e) G+O-RNN (f) G+O-TXT

Figure 12: Validation cost



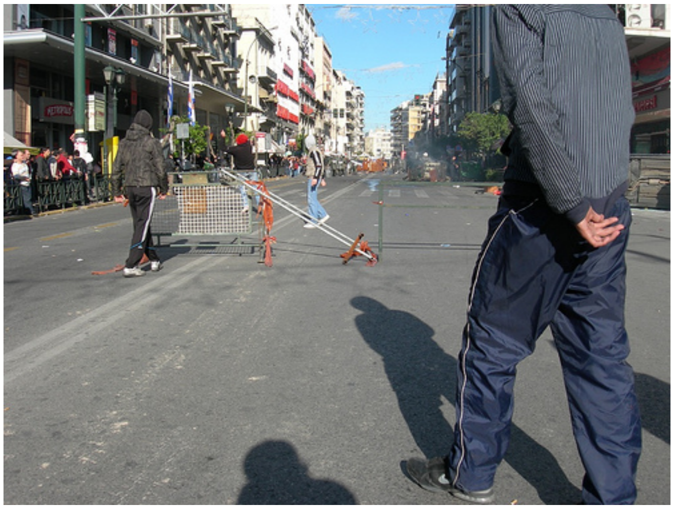
Source	two boys inside a fence jump in the air while holding a basketball.
Truth	zwei jungen innerhalb eines zaunes springen in die luft und halten dabei einen basketball.
VNMT	zwei jungen in einem zaun springen in die luft, während sie einen basketball hält.
Our Model (G)	zwei jungen in einem zaun springen in die luft und halten dabei einen basketball.

Figure 13: Translation 3



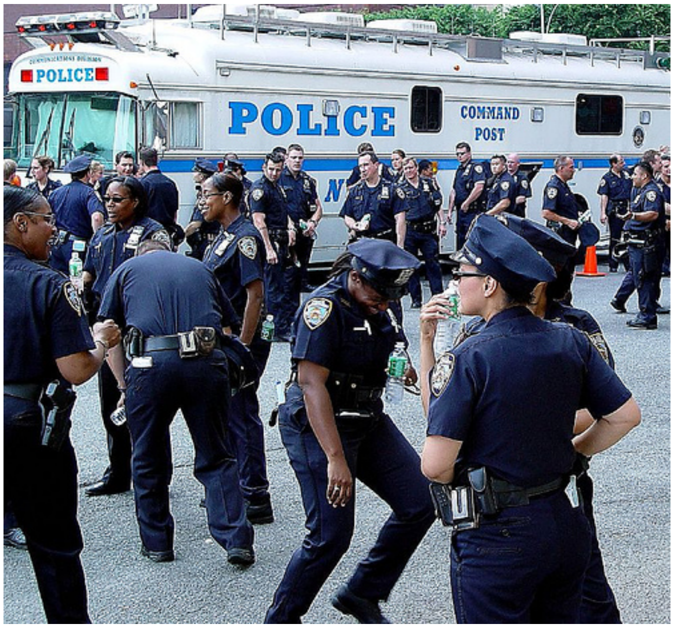
Source	a dog runs through the grass towards the camera.
Truth	ein hund rennt durch das gras auf die kamera zu.
VNMT	ein hund rennt durch das gras in die kamera.
Our Model (G)	ein hund rennt durch das gras auf die kamera zu.

Figure 14: Translation 4



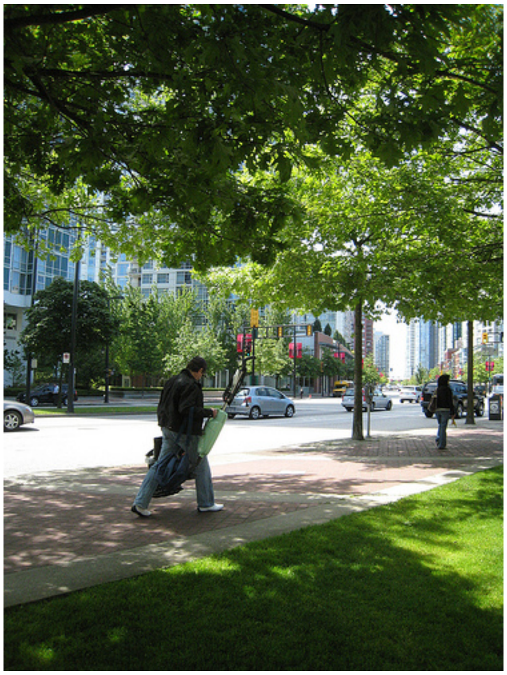
Source	a couple of men walking on a public city street.
Truth	einige männer gehen auf einer öffentlichen straße in der stadt.
VNMT	ein paar männer gehen auf einer öffentlichen stadtstraße.
Our Model (G)	ein paar männer gehen auf einer öffentlichen straße in der stadt.

Figure 15: Translation 5



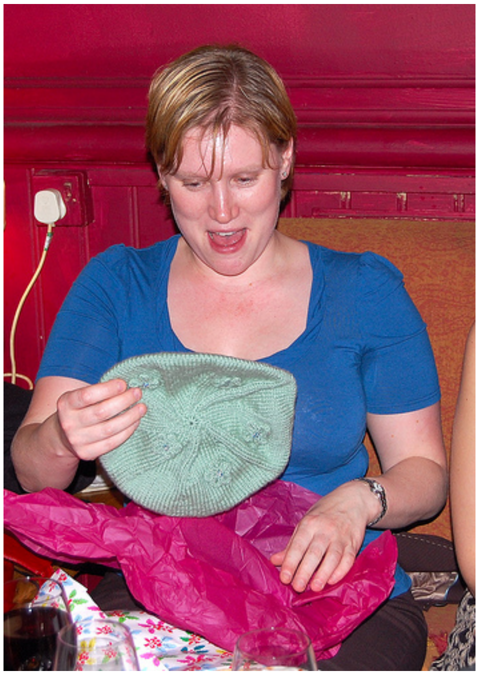
Source	a bunch of police officers are standing outside a bus.
Truth	eine gruppe von polizisten steht vor einem bus.
VNMT	eine gruppe von polizisten steht vor einem bus.
Our Model (G)	mehrere polizisten stehen vor einem bus.

Figure 16: Translation 6



Source	a man is walking down the sidewalk next to a street.
Truth	ein mann geht neben einer straße den gehweg entlang.
VNMT	ein mann geht neben einer straße den bürgersteig entlang.
Our Model (G)	ein mann geht auf dem bürgersteig an einer straße.

Figure 17: Translation 7



Source	a blond-haired woman wearing a blue shirt unwraps a hat.
Truth	eine blonde frau in einem blauen t-shirt packt eine mütze aus.
VNMT	eine blonde frau in einem blauen t-shirt wirft einen hut.
Our Model (G)	eine blonde frau trägt ein blaues hemd und einen hut.

Figure 18: Translation 8