

# TMU Japanese-English Multimodal Machine Translation System for WAT 2020

Hiroto Tamura    Tosho Hirasawa    Masahiro Kaneko    Mamoru Komachi

Tokyo Metropolitan University

6-6 Asahigaoka, Hino, Tokyo 191-0065, Japan

{tamura-hiroto, hirasawa-tosho, kaneko-masahiro}@ed.tmu.ac.jp  
komachi@tmu.ac.jp

## Abstract

We introduce our TMU system submitted to the Japanese↔English Multimodal Task (constrained) for WAT 2020 (Nakazawa et al., 2020). This task aims to improve translation performance with the help of another modality (images) associated with the input sentences. In a multimodal translation task, the dataset is, by its nature, a low-resource one. Our method used herein augments the data by generating noisy translations and adding noise to existing training images. Subsequently, we pretrain a translation model on the augmented noisy data, and then fine-tune it on the clean data. We also examine the probabilistic dropping of either the textual or visual context vector in the decoder. This aims to regularize the network to make use of both features while training. The experimental results indicate that translation performance can be improved using our method of textual data augmentation with noising on the target side and probabilistic dropping of either context vector.

## 1 Introduction

In recent years, neural machine translation (NMT) has become the standard machine translation system owing to its high performance (Sutskever et al., 2014; Bahdanau et al., 2015; Luong et al., 2015). However, NMT requires considerable parallel corpora for training; thus, it does not perform well in situations where low-resource data are present. To address this issue, Sennrich et al. (2016a) proposed back-translation that generates pseudo-parallel data by translating monolingual data in the target language.

Multimodal machine translation (MMT) is a task whose purpose is to generate better translations with information from other modalities (such as images) related to the source sentences (Specia et al., 2016). Owing to the nature of MMT, which requires image information paired with sentences,

the size of the available data is relatively small compared to that of text-only data. To overcome this issue, in this study, we augment training texts and images using several methods without external data. In our experiments, we pretrain the  $MMT_{decinit}$  model (see Subsection 3.2) on the augmented training data and fine-tune it on the original training data to improve translation performance.

Furthermore, to effectively utilize the features of both images and texts, we introduce the dropnet method (Zhu et al., 2020) into MMT models. It is expected to regularize the network training by probabilistically dropping one of the context vectors (textual or visual context vector) in the decoder. To the best of our knowledge, this is the first attempt incorporating the dropnet method into MMT with a recurrent neural network (RNN).

Our main findings herein are as follows:

- Textual data augmentation are better than visual data augmentation for MMT.
- Placing noise on the target side of the augmented data is effective in improving translation performance in the English→Japanese direction.
- The use of the dropnet method leads to improvements in translation performance.

## 2 Related Work

Several approaches to MMT have been proposed in the recent studies. Caglayan et al. (2016) and Calixto et al. (2017) proposed the doubly-attentive model wherein the encoder is a bi-directional gated recurrent unit (BiGRU) (Cho et al., 2014) that processes only the source sequence, and the decoder is a conditional GRU (CGRU)<sup>1</sup> that simultaneously

<sup>1</sup><https://github.com/nyu-dl/dl4mt-tutorial/blob/master/docs/cgru.pdf>

pays attention to the source sequence and the spatial visual feature. Calixto and Liu (2017) also used global visual features to initialize either the encoder or the decoder of the attention-based NMT. In the WMT 17 Shared Task on MMT, the models using global features were shown to be better than those using spatial features (Caglayan et al., 2017a). Additionally, Grönroos et al. (2018) adapted the Transformer (Vaswani et al., 2017) model to a multimodal setting and proposed concatenating the regional visual features encoded as a pseudo-word embedding to the word embeddings of the source sentence. According to them, however, the improvement achieved by incorporating visual information is modest, and they observed that external parallel data can significantly improve the performance. In contrast, we augmented training data without external data.

With respect to research on data augmentation in NMT, in addition to the method mentioned in Section 1, Fadaee et al. (2017) generated synthetic sentence pairs containing low-frequency words by leveraging the language models trained on large monolingual corpora. Under simulated low-resource settings, their results showed that translations using this augmentation approach have more low-frequency words than those not using this approach, leading to improved performance. Edunov et al. (2018) investigated back-translation at a large scale for generating useful synthetic source sentences using several approaches. They obtained back-translated data via sampling and noisy beam outputs and added them to parallel corpora. They found that the above methods outperform the ones that generate synthetic sentences based on argmax inference (e.g., beam or greedy search), except in low-resource settings.

### 3 Model

#### 3.1 NMT Model

Our baseline NMT (Caglayan et al., 2017a) is an attentive encoder-decoder model, wherein the encoder is BiGRU, and the decoder is CGRU. Thus, our MMT models are based on RNNs.

To generate synthetic data via textual data augmentation methods (see Subsection 4.1), we used the Transformer (Vaswani et al., 2017) model.

#### 3.2 MMT Model with Decoder Initialization

This MMT model initializes the hidden state of the decoder of our baseline NMT with global visual

features (Caglayan et al., 2017a). This model’s architecture is used for our baseline MMT as well as MMT models using augmented data. We denote this model as  $\text{MMT}_{\text{decinit}}$ .

#### 3.3 MMT Model with Double Attention

In our MMT model with a double attention mechanism, the decoder part of our baseline NMT model is extended to be multimodal (Caglayan et al., 2017a). While decoding, this model individually pays attention to the source sentence and the image to obtain the textual and visual context vectors. Subsequently, it combines both context vectors to obtain the multimodal context vector. In our experiments, we also adopt a hierarchical attention mechanism to combine each context vector (Libovický and Helcl, 2017). At each decoding step  $i$ , this attention combination projects each context vector into a common space (Equation 1) and computes another distribution with the projected context vectors (Equation 2). Then, we obtain the multimodal context vector by calculating the weighted average corresponding to each context vector (Equation 3).

$$e_i^{(k)} = v_b^\top \tanh(W_b s_i + U_b^{(k)} c_i^{(k)}), \quad (1)$$

$$\alpha_i^{(k)} = \frac{\exp(e_i^{(k)})}{\sum_{n=1}^2 \exp(e_i^{(n)})}, \quad (2)$$

$$c_{i,\text{multimodal}} = \sum_{k=1}^2 \alpha_i^{(k)} U_c^{(k)} c_i^{(k)} \quad (3)$$

where  $c_i^{(k)}$  is the  $k$ -th context vector (visual and textual),  $s_i$  is the decoder hidden state,  $v_b$  and  $W_b$  are trainable parameters, and  $U_b^{(k)}$  and  $U_c^{(k)}$  are encoder-specific matrices.

We denote this model as  $\text{MMT}_{\text{datt}}$ . In contrast to the  $\text{MMT}_{\text{decinit}}$  models, we do not conduct experiments for  $\text{MMT}_{\text{datt}}$  models using augmented data<sup>2</sup>. We incorporate the dropout method (see Subsection 3.3) into only this model for regularization owing to its architecture that combines each context vector.

**Dropnet method** Zhu et al. (2020) studied incorporating BERT (Devlin et al., 2019) into the

<sup>2</sup>We conducted preliminary experiments for both  $\text{MMT}_{\text{decinit}}$  and  $\text{MMT}_{\text{datt}}$  models using augmented data, although our dataset used in those experiments had the different splits of the dataset given by WAT 2020. As a result, the baseline  $\text{MMT}_{\text{datt}}$  model is inferior to the  $\text{MMT}_{\text{decinit}}$  model; moreover, the baseline  $\text{MMT}_{\text{datt}}$  model outperformed the ones pretrained on the augmented data. We therefore do not train  $\text{MMT}_{\text{datt}}$  models using augmented data.

$$\begin{aligned}
c_{i,\text{multimodal}} = & \mathbb{I}(r_i < \frac{p_{\text{net}}}{2}) \cdot \alpha_i^{(1)} U_c^{(1)} c_i^{(1)} + \mathbb{I}(r_i > 1 - \frac{p_{\text{net}}}{2}) \cdot \alpha_i^{(2)} U_c^{(2)} c_i^{(2)} \\
& + \mathbb{I}(\frac{p_{\text{net}}}{2} \leq r_i \leq 1 - \frac{p_{\text{net}}}{2}) \cdot \sum_{k=1}^2 \alpha_i^{(k)} U_c^{(k)} c_i^{(k)}
\end{aligned} \tag{4}$$

Transformer (Vaswani et al., 2017) model in the translation task. They proposed the dropout method that intends to regularize the network training to fully utilize the features output from BERT and the conventional encoder. Specifically, at any layer  $l$  in the encoder during training, with probability  $p_{\text{net}}/2$ , the output from either BERT or the  $(l-1)$ -th layer of the encoder is selected for computing the  $l$ -th layer’s output, and with probability  $(1-p_{\text{net}})$ , both outputs are used for computing the  $l$ -th layer’s output, where the dropout rate  $p_{\text{net}} \in [0, 1]$ .

Similar to the above method, we attempt regularization by probabilistically dropping either the textual or the visual context vector. Our  $\text{MMT}_{\text{datt}}$  model combines each context vector to obtain the multimodal context vector in the decoder. Therefore, we adapt the dropout method to the decoder. At each decoding step  $i$  while training, with probability  $p_{\text{net}}/2$ , either the textual context vector or the visual context vector is used for computing the multimodal context vector; with probability  $(1-p_{\text{net}})$ , both context vectors are used for the multimodal context vector (Equation 4). In Equation 4,  $\mathbb{I}(\cdot)$  is the indicator function,  $r_i$  is a random variable uniformly sampled from  $[0, 1]$ ,  $c_i^{(k)}$  is the  $k$ -th context vector.

## 4 Data Augmentation Method

### 4.1 Textual Data Augmentation

**Sampling** This method samples the hypothesis from the output distribution at each decoding step to generate synthetic parallel data (Edunov et al., 2018).

**Random noising** This method was originally used to generate synthetic ungrammatical sentences in the grammatical error correction task (Xie et al., 2018). They penalized every hypothesis on the beam by adding noise  $r\beta$  to its hypothesis’ score, where  $r$  is drawn from the uniform distribution on the interval  $[0, 1]$  during the beam search procedure, and  $\beta$  controls the noise intensity. If  $\beta$  is sufficiently large, this method is similar to the method that randomly shuffles the ranks of the hypotheses

according to their scores.

### 4.2 Visual Data Augmentation

We incorporate several image data augmentation methods based on computer vision tasks (Shorten and Khoshgoftaar, 2019). According to Luke and Geoff (2018), they augmented images in several ways in the image recognition task and demonstrated that the cropping method achieved the highest accuracy, followed by rotation. We likewise choose cropping (center cropping and random cropping) and rotation methods from the above results.

**Center cropping** This method crops a center patch ( $256 \times 256$  size) of each image.

**Random cropping** This method randomly selects a patch ( $256 \times 256$  size) of each image and crops it.

**Rotation** This method rotates the images right or left on an axis between  $-20^\circ$  and  $20^\circ$  randomly. This range is useful for digit recognition tasks such as MNIST (Shorten and Khoshgoftaar, 2019).

## 5 Experimental Setup

### 5.1 Data

For training and validation, we use the Flickr30k Entities Japanese dataset<sup>3</sup> for Japanese sentences, the Flickr30k Entities dataset<sup>4</sup> for English sentences, and the Flickr30k dataset<sup>5</sup> for images. For test data, we use both Japanese and English sentences provided by WAT 2020, and their associated images are in the Flickr30k dataset. The Japanese training data size is originally 59,566 sentences, but four sentences are missing; thus, we use 59,562 sentences (both Japanese and English) for training. We use Moses (Koehn et al., 2007) scripts to lowercase, normalize, and tokenize English sentences,

<sup>3</sup><https://github.com/nlab-mpg/Flickr30kEnt-JP>

<sup>4</sup>[https://github.com/BryanPlummer/flickr30k\\_entities](https://github.com/BryanPlummer/flickr30k_entities)

<sup>5</sup><http://shannon.cs.illinois.edu/DenotationGraph/>

and tokenized Japanese sentences using MeCab<sup>6</sup> with the IPA dictionary. The evaluation metric used is BLEU calculated by *multibleu.perl*<sup>7</sup> of Moses. We use the word-level vocabularies of 9,546 items for English and 11,235 items for Japanese. We also used byte pair encoding (BPE: Sennrich et al., 2016b) for vocabularies, but the word-level method demonstrated better results than BPE; therefore, we decide to use word-level vocabularies.

To augment training texts, we train a text-only Transformer model on the original training texts. Subsequently, we translate the original English/Japanese training texts into Japanese/English texts as additional training texts using the trained model. We train the models with three different seeds and translate with the trained model having the highest score on the dev set among the three trained models. We generate noisy training sentences using the sampling method and the random noising method with each value  $\beta = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 15, 20$  (if  $\beta = 0$ , the models simply translate English or Japanese sentences without noise).

For images, we augment images using Albumentations<sup>8</sup>, which is a library for image data augmentation. We extract the visual features from a pretrained CNN model, ResNet-50 (He et al., 2016). The size of the spatial features extracted from the *res4f\_relu* layer is  $14 \times 14 \times 1024$ , and the global features extracted from the *pool5* layer are 2,048-dimensional features.

## 5.2 Model

We conduct our experiments with the toolkit *nmtpytorch* version 4.0.0<sup>9</sup> (Caglayan et al., 2017b), except during the textual data augmentation step. The encoder and decoder GRUs have 320 hidden dimensions, and word embeddings are 200 dimensions. We use the Adam (Kingma and Ba, 2015) optimizer with a learning rate of  $4e-4$ , and a batch size of 64 for training and 32 for evaluation. We adopt the early stopping for training if the BLEU score of the dev set does not improve for ten epochs. The beam size is 12, and the total gradient

<sup>6</sup><https://taku910.github.io/mecab/>

<sup>7</sup><https://github.com/moses-smt/mosesdecoder/blob/master/scripts/generic/multi-bleu.perl>

<sup>8</sup><https://github.com/albumentations-team/albumentations>

<sup>9</sup><https://github.com/lium-lst/nmtpytorch/tree/v4.0.0>

norm is clipped to 1. We set the dropout probability  $p_{\text{net}}$  to 0.3. For the English→Japanese (En→Ja) direction, dropout (Srivastava et al., 2014) rates applied to source embeddings, source annotations, and pre-softmax activations are (0.4, 0.4, 0.6) for our NMT model, (0.3, 0.4, 0.5) for our MMT<sub>decinit</sub> model, and (0.4, 0.4, 0.4) for our MMT<sub>datt</sub> model, respectively. For the Japanese→English (Ja→En) direction, we set the dropout rates (0.4, 0.3, 0.4) for our NMT model, (0.5, 0.3, 0.4) for our MMT<sub>decinit</sub> model, and (0.4, 0.3, 0.3) for our MMT<sub>datt</sub> model. We use same dropout rates set on each model when both pretraining and fine-tuning.

**Data augmentation** We train a text-only Transformer model on the original training texts using the *fairseq*<sup>10</sup> toolkit for generating noisy texts. Both the encoder and the decoder have six blocks, and the input and output embeddings of the decoder are shared. The word embedding size and the hidden size is 512 dimensions. We optimize the models with Adam ( $\beta_1 = 0.9, \beta_2 = 0.98$ ). The learning rate is  $4e-4$  and the maximum number of tokens for each mini-batch is 4,096. The beam size is 5, and the total gradient norm is clipped to 5.0. We set the dropout rate of 0.1 for both directions.

We repeated each experiment with three different seeds to calculate the BLEU score by averaging three scores on the dev set in order to select the best model for textual augmentation and fine-tuning as well as to ensemble the models.

## 5.3 Training

### 5.3.1 w/o augmented data

Our baseline NMT, baseline MMT<sub>decinit</sub> and MMT<sub>datt</sub> models are trained on the original training data.

### 5.3.2 w/ augmented data

For the MMT<sub>decinit</sub> models using augmented data, we first pretrain our MMT<sub>decinit</sub> model on augmented data obtained via the above-mentioned method (see Section 4). We pretrain three models with each different seed and select the best pretrained model. Thereafter, we fine-tune the selected pretrained model on clean data.

**Augmented texts** Usually, when using augmented texts, training data comprise noisy texts on the source side and clean texts on the target side. In addition, we experimented with noisy texts on

<sup>10</sup><https://github.com/pytorch/fairseq>

<b>Model (En→Ja)</b>	<b>dev</b>
Baseline NMT	40.34
Baseline MMT <sub>decinit</sub>	40.50
<b>Textual data augmentation</b>	
MMT <sub>decinit</sub> w/ sampling (target)	40.33
MMT <sub>decinit</sub> w/ random noising ( $\beta = 5$ ; source)	40.68
MMT <sub>decinit</sub> w/ random noising ( $\beta = 5$ ; target)	40.69
MMT <sub>decinit</sub> w/ mix ( $\beta = 1, 5, 6$ ; target)	<b>40.75</b>
<b>Visual data augmentation</b>	
MMT <sub>decinit</sub> w/ center cropping	40.50
MMT <sub>decinit</sub> w/ random cropping	40.59
MMT <sub>decinit</sub> w/ rotation	40.54

Table 1: En→Ja results: BLEU scores on the dev set. “MMT<sub>decinit</sub> w/ sampling (target)” denotes the model is pre-trained on the noisy texts generated via sampling on the target side, and fine-tuned on the clean data. “MMT<sub>decinit</sub> w/ random noising ( $\beta = 5$ ; source)” denotes the model is pretrained on the noisy texts generated with random noising ( $\beta = 5$ ) on the source side, and fine-tuned on the clean data. The model “MMT<sub>decinit</sub> w/ mix ( $\beta = 1, 5, 6$ ; target)”, is pretrained on the data mixed with each text generated with random noising method with  $\beta = 1$ ,  $\beta = 5$  and  $\beta = 6$  each on the target side. The model “MMT<sub>decinit</sub> w/ center cropping” is pretrained on augmented images which are center cropped, and fine-tuned on clean data.

<b>Model (Ja→En)</b>	<b>dev</b>
Baseline NMT	<b>42.86</b>
Baseline MMT <sub>decinit</sub>	42.81
<b>Textual data augmentation</b>	
MMT <sub>decinit</sub> w/ sampling (target)	42.73
MMT <sub>decinit</sub> w/ random noising ( $\beta = 2$ ; source)	42.38
MMT <sub>decinit</sub> w/ random noising ( $\beta = 4$ ; target)	42.84
MMT <sub>decinit</sub> w/ mix ( $\beta = 4$ , sampling; target)	42.55

Table 2: Ja→En results: BLEU scores on the dev set. The model “MMT<sub>decinit</sub> w/ mix ( $\beta = 4$ , sampling; target)” is pretrained on the mixed data of the noisy texts with random noising ( $\beta = 4$ ) and the one with sampling both on the target side.

the target side and clean texts on the source side during pretraining. We thus aim to utilize the visual feature more by smoothing attention to texts during decoding. Furthermore, we combine the noisy texts generated by different textual data augmentation methods for pretraining. For example, if we combine each generated data with a random noising method with  $\beta = 1$  and  $\beta = 2$ , the mixed textual data comprise 119,124 sentences.

**Augmented images** In the case of using augmented images for pretraining, we use clean texts with associated augmented images which are center cropped, random cropped or rotated. When fine-tuning, both clean texts and clean images are used.

## 6 Results and Analysis

**En→Ja translation** Table 1 shows the BLEU scores on the dev set for the English→Japanese direction. We found that using noisy data with the random noising method is effective for the En-Ja direction. For the “random noising” models on the target side, the model of random noising method with  $\beta = 5$  has achieved the highest score among the ones with other data augmentation methods, followed by  $\beta = 1$  and  $\beta = 6$ . Therefore, we chose  $\beta = 1$ ,  $\beta = 5$ , and  $\beta = 6$  for the “mix” model. Our MMT<sub>decinit</sub> model outperforms the baseline NMT by 0.16 BLEU points. MMT<sub>decinit</sub> models pretrained on augmented textual data gain more than 0.18 points compared to the baseline

Model (En→Ja)	dev	test	
		BLEU	RIBES
Baseline MMT <sub>decinit</sub>	40.41	43.30	0.8639
MMT <sub>datt</sub> w/ dropout	39.96	42.65	0.8657
MMT <sub>decinit</sub> w/ random noising ( $\beta = 1, 5, 6$ ; target)	40.42	44.12	0.8648
Ensemble (3 baseline MMT <sub>decinit</sub> models)	40.80	43.99	0.8684
Ensemble (3 MMT <sub>datt</sub> models w/ dropout)	41.30	43.78	<b>0.8715</b>
Ensemble (top 6 models)	<b>41.40</b>	<b>44.57</b>	0.8699

Table 3: En→Ja published results: BLEU and RIBES scores on the dev and test set. The tokenizer is MeCab. “Ensemble (top 6 models)” is ensembled of the six models (“MMT<sub>decinit</sub> w/ random noising ( $\beta = 1$ ; target)”, “MMT<sub>decinit</sub> w/ random noising ( $\beta = 0$ ; target)”, “Baseline MMT<sub>decinit</sub>”, “MMT<sub>decinit</sub> w/ random noising ( $\beta = 10$ ; target)”, “MMT<sub>decinit</sub> w/ random cropping” and “MMT<sub>decinit</sub> w/ random noising ( $\beta = 6$ ; target)”).

Model (Ja→En)	dev	test	
		BLEU	RIBES
Baseline MMT <sub>decinit</sub>	42.73	46.19	0.8951
MMT <sub>datt</sub> w/ dropout	42.70	46.26	0.8959
Ensemble (3 baseline MMT <sub>decinit</sub> models)	44.03	<b>48.38</b>	0.8996
Ensemble (3 MMT <sub>datt</sub> models w/ dropout)	44.33	48.33	<b>0.9007</b>
Ensemble (top 6 models)	<b>44.44</b>	47.86	0.9000

Table 4: Ja→En published results: BLEU and RIBES scores on the dev and test set. The tokenizer is Moses. “Ensemble (top 6 models)” ensembled of the six models (“Baseline MMT<sub>decinit</sub>”, “MMT<sub>decinit</sub> w/ random noising ( $\beta = 4$ ; target)”, “MMT<sub>decinit</sub> w/ random noising ( $\beta = 1$ ; target)”, “MMT<sub>decinit</sub> w/ sampling (target)”, “MMT<sub>decinit</sub> w/ random noising ( $\beta = 4$ ; target)” and “MMT<sub>decinit</sub> w/ random noising ( $\beta = 8$ ; target)”)

MMT<sub>decinit</sub>, except for “sampling (target)”. “random cropping” has the highest score among the three models using augmented visual data, but its score exceeds that of the baseline MMT<sub>decinit</sub> by only 0.09 points. Therefore, we did not conduct experiments using augmented visual data for the Ja→En direction.

We show our published results for the En→Ja direction. In Table 3, the single model has the highest score among its three models with each different seed. The model “Ensemble (top 6 models)” is an ensemble of the top six models in terms of the BLEU metric among all trained single models. It outperforms the other models in terms of the BLEU metric, but its RIBES scores are lower than those of “Ensemble (3 MMT<sub>datt</sub> models w/ dropout)”. Moreover, comparing “Baseline MMT<sub>decinit</sub>” and “MMT<sub>datt</sub> w/ dropout”, the RIBES score of the latter is higher than the former but not the BLEU metric.

Additionally, we show the translation examples of several models in the appendix, focusing on the quality of “Ensemble (top 6 models)”. A good

translation of “Ensemble (top 6 models)” is in Table 7, and a poor translation of it is in Table 8.

**Ja→En translation** Table 2 shows the BLEU scores on the dev set for the Ja→En direction. We found that the noisy data did not have a positive effect on translation performance. We chose  $\beta = 2$  because the model pretrained on noisy data on the source side with the random noising method with  $\beta = 2$  is the best. Likewise, we chose  $\beta = 4$  of the random noising method, which achieves the best score among the models pretrained with the noisy data on the target side, followed by sampling. Hence, we used the noisy data (both on the target side) with random noising ( $\beta = 4$ ) and sampling for pretraining the model “mix”. Contrary to the En→Ja results, our baseline NMT model is 0.05 points higher than that of the baseline MMT<sub>decinit</sub>. “random noising ( $\beta = 4$ ; target)” gains 0.03 points over the baseline MMT<sub>decinit</sub>; however, it still does not reach the baseline NMT score.

We present our published results for the Ja→En direction in Table 4. “Ensemble (top 6 models)”

Model	En→Ja		Ja→En	
	BLEU	RIBES	BLEU	RIBES
Baseline MMT <sub>decinit</sub>	<b>40.50</b>	0.8190	<b>42.81</b>	<b>0.8741</b>
MMT <sub>datt</sub> w/o dropnet	39.89	0.8193	42.16	0.8726
MMT <sub>datt</sub> w/ dropnet	40.39	<b>0.8206</b>	42.72	0.8730

Table 5: BLEU and RIBES scores on the dev set for comparing the effect of the models with or without dropnet method. Each score is the averaged score of three models with different seeds.

	En→Ja	Ja→En
source	40.21	42.01
target	<b>40.59</b>	<b>42.50</b>

Table 6: BLEU scores on the dev set for each direction; a comparison between placing noise on the source side and target side when pretraining. “source” is the score that averages each score of thirteen models with different  $\beta$  pretrained on the noisy data on the source side.

does not achieve the best score in terms of both the BLEU and RIBES metrics. “Ensemble (3 baseline MMT<sub>decinit</sub> models)” surpasses the other models in terms of the BLEU metric, and “Ensemble (3 MMT<sub>datt</sub> models w/ dropnet)” is the best in terms of the RIBES metric.

We show the translation examples of several models in the appendix, focusing on the quality of “Ensemble (top 6 models)”. There are a good translation of “Ensemble (top 6 models)” in Table 9 and a poor translation of it in Table 10.

**Dropnet** Table 5 shows the results of MMT models with and without the dropnet method. Although the models trained both with and without the dropnet method have lower BLEU scores than the baseline MMT<sub>decinit</sub>, the model with dropnet gains 0.5 BLEU points for the En→Ja direction and 0.56 BLEU points for the Ja→En direction than the one without dropnet. For the RIBES metric, the model with dropnet achieves the best RIBES score in the En→Ja direction, but the baseline MMT<sub>decinit</sub> is the best model for the Ja→En direction. Moreover, the differences between the RIBES scores of each model are marginal; however, the models with dropnet are better than those without dropnet. These demonstrate that incorporating the dropnet method into MMT<sub>datt</sub> models can help improve the translation performance.

**Source vs target noising** We investigated how the score is affected by whether the noise data for

pretraining is on the source or the target side. As indicated in Table 6, we pretrain the models on the noisy data generated with thirteen different  $\beta$  values (i.e.,  $\beta = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 15, 20$ ) and calculate the average of these thirteen models scores. This table shows that using the noisy data on the target side is better than on the source side for both directions while pretraining. Although it is unclear why the noisy data on the target side is effective, we infer this setting is useful under low-resource or MMT situations.

## 7 Conclusion

We introduced several data augmentation methods to solve the low-resource problem in MMT. These methods have been shown to be useful for the En→Ja direction, but not for the Ja→En direction. The random noising method positively affects translation performance compared to other data augmentation methods. Furthermore, it is notable that adding noise on the target side is more effective than on the source side for textual data augmentation. We also adapted the dropnet method for the regularization of double attention mechanism for MMT. This method is effective compared to not using dropnet.

For future work, we investigate why the noisy data on the target side is effective. We also explore other textual and visual data augmentation methods and mixing visual augmented data for pretraining; moreover, we research whether combining textual and visual augmented data improves the performance or not. Furthermore, we study how the MMT models with the dropnet method work.

## References

- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. [Neural Machine Translation by Jointly Learning to Align and Translate](#). In *ICLR*.
- Ozan Caglayan, Walid Aransa, Adrien Bardet, Mer-

- cedes García-Martínez, Fethi Bougares, Loïc Barrault, Marc Masana, Luis Herranz, and Joost van de Weijer. 2017a. [LIUM-CVC Submissions for WMT17 Multimodal Translation Task](#). In *WMT*, pages 432–439.
- Ozan Caglayan, Loïc Barrault, and Fethi Bougares. 2016. [Multimodal Attention for Neural Machine Translation](#). *CoRR*, abs/1609.03976.
- Ozan Caglayan, Mercedes García-Martínez, Adrien Bardet, Walid Aransa, Fethi Bougares, and Loïc Barrault. 2017b. [NMTPY: A Flexible Toolkit for Advanced Neural Machine Translation Systems](#). *Journal of Prague Bull. Math. Linguistics*, 109:15–28.
- Iacer Calixto and Qun Liu. 2017. [Incorporating Global Visual Features into Attention-based Neural Machine Translation](#). In *EMNLP*, pages 992–1003.
- Iacer Calixto, Qun Liu, and Nick Campbell. 2017. [Doubly-Attentive Decoder for Multi-modal Neural Machine Translation](#). In *ACL*, pages 1913–1924.
- Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. [On the Properties of Neural Machine Translation: Encoder–Decoder Approaches](#). In *SSST*, pages 103–111.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding](#). In *NAACL*, pages 4171–4186.
- Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. [Understanding Back-Translation at Scale](#). In *EMNLP*, pages 489–500.
- Marzieh Fadaee, Arianna Bisazza, and Christof Monz. 2017. [Data Augmentation for Low-Resource Neural Machine Translation](#). In *ACL*, pages 567–573.
- Stig-Arne Grönroos, Benoit Huet, Mikko Kurimo, Jorma Laaksonen, Bernard Merialdo, Phu Pham, Mats Sjöberg, Umut Sulubacak, Jörg Tiedemann, Raphael Troncy, and Raúl Vázquez. 2018. [The MeMAD Submission to the WMT18 Multimodal Translation Task](#). In *WMT*, pages 603–611.
- K. He, X. Zhang, S. Ren, and J. Sun. 2016. [Deep Residual Learning for Image Recognition](#). In *CVPR*, pages 770–778.
- Diederik P. Kingma and Jimmy Ba. 2015. [Adam: A Method for Stochastic Optimization](#). In *ICLR*.
- 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 *ACL*, pages 177–180.
- Jindřich Libovický and Jindřich Helcl. 2017. [Attention Strategies for Multi-Source Sequence-to-Sequence Learning](#). In *ACL*, pages 196–202.
- Taylor Luke and Nitschke Geoff. 2018. [Improving Deep Learning with Generic Data Augmentation](#). In *IEEE SSCI*, pages 1542–1547.
- Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. [Effective Approaches to Attention-based Neural Machine Translation](#). In *EMNLP*, pages 1412–1421.
- Toshiaki Nakazawa, Hideki Nakayama, Chenchen Ding, Raj Dabre, Hideya Mino, Isao Goto, Win Pa Pa, Anoop Kunchukuttan, Shantipriya Parida, Ondřej Bojar, and Sadao Kurohashi. 2020. [Overview of the 7th Workshop on Asian Translation](#). In *WAT*.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. [Improving Neural Machine Translation Models with Monolingual Data](#). In *ACL*, pages 86–96.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. [Neural Machine Translation of Rare Words with Subword Units](#). In *ACL*, pages 1715–1725.
- Connor Shorten and T. Khoshgoftaar. 2019. [A survey on Image Data Augmentation for Deep Learning](#). *Journal of Big Data*, 6:1–48.
- Lucia Specia, Stella Frank, Khalil Sima'an, and Desmond Elliott. 2016. [A Shared Task on Multimodal Machine Translation and Crosslingual Image Description](#). In *WMT*, pages 543–553.
- Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. [Dropout: A Simple Way to Prevent Neural Networks from Overfitting](#). *Journal of Machine Learning Research*, 15(56):1929–1958.
- Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. [Sequence to Sequence Learning with Neural Networks](#). In *NeurIPS*, pages 3104–3112.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, and Jakó Uszkoreit. 2017. [Attention is All you Need](#). In *NeurIPS*, pages 5998–6008.
- Ziang Xie, Guillaume Genthial, Stanley Xie, Andrew Ng, and Dan Jurafsky. 2018. [Noising and Denoising Natural Language: Diverse Backtranslation for Grammar Correction](#). In *NAACL*, pages 619–628.
- Jinhua Zhu, Yingce Xia, Lijun Wu, Di He, Tao Qin, Wengang Zhou, Houqiang Li, and Tieyan Liu. 2020. [Incorporating BERT into Neural Machine Translation](#). In *ICLR*.



## A Translation Examples

We show several translation examples focusing on “Ensemble (top 6 models)” below.

Image	
Source	a man with blue sleeveless shirt and sunglasses paints a face while on his knees .
Baseline NMT	青い袖なしのシャツを着てサングラスをかけた男性が、膝の上に顔を塗っている。
Baseline MMT <sub>decinit</sub>	青い袖なしシャツとサングラスを身につけた男性は、膝の上に顔を描いている。
MMT <sub>datt</sub> w/ dropout	青い袖なしのシャツを着てサングラスをかけた男性が膝の上に顔を描いている。
MMT <sub>decinit</sub> w/ random noising ( $\beta = 1, 5, 6$ ; target)	青い袖なしシャツを着てサングラスをかけた男性が、顔を膝に塗っている。
Ensemble (3 baseline MMT <sub>decinit</sub> models)	青い袖なしのシャツを着てサングラスをかけた男性が、膝の上に顔を塗っている。
Ensemble (3 MMT <sub>datt</sub> models w/ dropout)	青い袖なしのシャツを着てサングラスをかけた男性が、膝の上に顔を塗っている。
Ensemble (top 6 models)	青い袖なしのシャツを着てサングラスをかけた男性が、膝をついて顔を描いている。
Reference	青い袖無しシャツにサングラスの男性が、ひざまづいて顔を描いている。

Table 7: A **good** translation example of “Ensemble (top 6 models)” and other models’ examples on the dev set for the En→Ja direction.

Image	
Source	a older man in a orange wrap looks into the camera on the sidewalk of a city .
Baseline NMT	オレンジ色の布を着た年配の男性が街の歩道でカメラを覗き込んでいる。
Baseline MMT <sub>decinit</sub>	オレンジ色の布を身につけた年配の男性が街の歩道でカメラを覗き込んでいる。
MMT <sub>datt</sub> w/ dropout	オレンジ色のラップトップを着た年配の男性が、街の歩道でカメラを覗き込んでいる。
MMT <sub>decinit</sub> w/ random noising ( $\beta = 1, 5, 6$ ; target)	街の歩道で、オレンジ色の布を身に着けた年配の男性がカメラを覗き込んでいる。
Ensemble (3 baseline MMT <sub>decinit</sub> models)	オレンジ色のラップトップを着た年配の男性が、街の歩道でカメラを覗き込んでいる。
Ensemble (3 MMT <sub>datt</sub> models w/ dropout)	オレンジ色の布を着た年配の男性が、街の歩道でカメラを覗き込んでいる。
Ensemble (top 6 models)	オレンジ色のラップトップを着た年配の男性が、街の歩道でカメラを覗き込んでいる。
Reference	オレンジ色の布をまとった年配の男性が街の歩道でカメラの中を見る。

Table 8: A **poor** translation example of “Ensemble (top 6 models)” and other models’ examples on the dev set for the En→Ja direction.


Image	
Source	<p>プロパンバーベキューの隣で、小さなスツールに腰掛けながら、何かを食べている女性。</p>
Baseline NMT	<p>a woman sitting on a small stool eating something while sitting on a small stool next to a propane barbecue .</p>
Baseline MMT <sub>decinit</sub>	<p>a woman sitting on a small stool sitting on a small stool eating something .</p>
MMT <sub>datt</sub> w/ dropnet	<p>a woman sitting on a small stool eating something next to a propane barbecue .</p>
Ensemble (3 baseline MMT <sub>decinit</sub> models)	<p>a woman eating something while sitting on a small stool next to a charcoal barbecue .</p>
Ensemble (3 MMT <sub>datt</sub> models w/ dropnet)	<p>a woman eating something while sitting on a small stool next to a charcoal barbecue .</p>
Ensemble (top 6 models)	<p>a woman eating something while sitting on a small stool next to a propane barbecue .</p>
Reference	<p>a woman eating some food while sitting on a tiny stool next to a propane barbecue .</p>

Table 9: A **good** translation example of “Ensemble (top 6 models)” and other models’ examples on the dev set for the Ja→En direction.


Image	
Source	<p>アジアの少年のグループがバーベキューで肉が焼けるのを待っている。</p>
Baseline NMT	<p>a group of asian boys wait for meat and meat at a barbecue .</p>
Baseline MMT <sub>decinit</sub>	<p>a group of asian boys wait for meat on a barbecue .</p>
MMT <sub>datt</sub> w/ dropnet	<p>a group of asian boys wait for meat on a barbecue .</p>
Ensemble (3 baseline MMT <sub>decinit</sub> models)	<p>a group of asian boys are waiting for meat on a barbecue .</p>
Ensemble (3 MMT <sub>datt</sub> models w/ dropnet)	<p>a group of asian boys are waiting for meat on a barbecue .</p>
Ensemble (top 6 models)	<p>a group of asian boys wait at a barbecue at a barbecue .</p>
Reference	<p>group of asian boys wait for meat to cook over barbecue .</p>

Table 10: A **poor** translation example of “Ensemble (top 6 models)” and other models’ examples on the dev set for the Ja→En direction.