

Received 26 April 2022, accepted 4 June 2022, date of publication 22 June 2022, date of current version 30 June 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3185243

# Pre-Trained Word Embedding and Language Model Improve Multimodal Machine Translation: A Case Study in Multi30K

TOSHO HIRASAWA<sup>1</sup>, MASAHIRO KANEKO<sup>1</sup>, AIZHAN IMANKULOVA,  
AND MAMORU KOMACHI<sup>1</sup>

Graduate School of System Design, Tokyo Metropolitan University, Hino, Tokyo 191-0065, Japan

Corresponding author: Tosho Hirasawa (hirasawa-tosho@ed.tmu.ac.jp)

**ABSTRACT** Multimodal machine translation (MMT) is an attractive application of neural machine translation (NMT) that is commonly incorporated with image information. However, the MMT models proposed thus far have only comparable or slightly better performance than their text-only counterparts. One potential cause of this infeasibility is a lack of large-scale data. Most previous studies mitigate this limitation by employing large-scale textual parallel corpora, which are more accessible than multimodal parallel corpora, in various ways. However, these corpora are still available on only a limited scale in low-resource language pairs or domains. In this study, we leveraged monolingual (or multimodal monolingual) corpora, which are available at scale in most languages and domains, to improve MMT models. Our approach follows that of previous unimodal works that use monolingual corpora to train the word embedding or language model and incorporate them into NMT systems. While these methods demonstrated the advantage of using pre-trained representations, there is still room for MMT models to improve. To this end, our system employs debiasing procedures for the word embedding and multimodal extension of the language model (visual-language model, VLM) to make better use of the pre-trained knowledge in the MMT task. The results of evaluations conducted on the de facto MMT dataset for the English–German translation indicate that the improvement obtained using well-tailored word embedding and VLM is approximately +1.84 BLEU and +1.63 BLEU, respectively. The evaluation on multiple language pairs reveals their adoptability across the languages. Beyond the success of our system, we also conducted an extensive analysis on VLM manipulation and showed promising areas for developing better MMT models by exploiting VLM; some benefits brought by either modality are missing, and MMT with VLM generates less fluent translations. Our code is available at <https://github.com/toshohirasawa/mmt-with-monolingual-data>.

**INDEX TERMS** Multimodal machine translation, natural language processing, neural machine translation.

## I. INTRODUCTION

In multimodal machine translation (MMT), a target sentence is translated from a source sentence together with related nonlinguistic information, such as images. Since the development of a multimodal parallel corpus, namely, Multi30K [1], most research in this area has focused on incorporating static images into encoder–decoder neural machine translation (NMT) systems. In an image-guided machine translation task, the multimodal NMT models are expected to disambiguate lexical ambiguity in the source and target languages

The associate editor coordinating the review of this manuscript and approving it for publication was Arianna Dulizia<sup>1</sup>.

or correct inaccurate expressions in the source sentence [2]. These models should also resolve language phenomena that exist only in the target languages. The success of MMT has encouraged its application in translating subtitles of movies, utterances in conferences, and descriptions of paintings.

Following the proposal of the encoder–decoder NMT model [3], large-scale parallel corpora have been built to train better NMT systems. For news translation, an English–Czech NMT system acquired 170k data and achieved quality on par with that of human translators [4]. More recently, [5] employed over 200M data to train an English–German and English–Japanese NMT model that won first place in an international competition [6]. However, the Multi30K [1],

a well-established corpus for image-guided machine translation tasks, is annotated based on an image-captioning dataset (Flickr30K [7]), and comprises 30k tuples of image, English sentence, and German sentence. Compared to the datasets used for news translation or other text-only translation tasks, the Multi30K dataset is considered a low-resource dataset. Owing to a lack of available large-scale datasets, training high-quality MMT models is challenging, and has resulted in only “modest” improvement being achieved from using image information [8]. To address this problem, previous studies employed large-scale textual parallel corpora. [8] trained a text-only NMT model on a textual parallel corpus (OpenSubtitles [9]) and translated a multimodal monolingual corpus (MS-COCO [10]) to augment the data for training MMT models. More recently, [11] also employed the same strategy to augment the training data. They trained a visual translation language model (VTLM)—an extension of the translation language model (TLM) [12]—and transferred the weight of the VTLM to initialize MMT models. Despite the success of these approaches, the limitations on the use of large-scale “parallel” corpora circumscribe their availability in low-resource languages.

In this study, we focus on manipulating knowledge obtained from either textual or multimodal monolingual corpora, which is more feasible for the low-resource domain. Specifically, we prove the usefulness of the pre-trained word embedding and language model for MMT models. Although these approaches were developed for text-only MT, their application toward MMT has room for improvement and is worthy of consideration.

The main contributions of this study are as follows:

- 1) We propose two approaches to incorporate a monolingual (or multimodal monolingual) corpus into MMT models; one uses a pre-trained word embedding using a debiasing procedure and monolingual-corpus-based subword tokenization; the other uses VLM.
- 2) We demonstrate that both approaches achieve substantial improvement over their baselines; in particular, the MMT model fused with VLM consistently outperforms its text-only counterparts across a range of target languages.

## A. WORD EMBEDDING

Pre-trained word embedding is considered an important part of neural network models in many natural language processing (NLP) tasks. In the context of NMT, pre-trained word embedding has proved useful in low-resource domains [13], in which FastText [14] embedding is used to initialize the encoder and decoder of the NMT model. They also indicate that the improvement achieved using the pre-trained word embedding decreases as the training data increases. Reference [15] introduced an MMT model with embedding prediction that provided substantial performance improvement.

However, many studies have proven that the vectors in a pre-trained word embedding distribute unevenly in a narrow

conical subspace rather than evenly in the entire space. This highly localized geometry of pre-trained word embeddings harms the isotropy of word embedding and consequently their advantage in downstream tasks. In these embedding spaces, some words appear frequently in the nearest neighbors of other words [16], [17]. This is called the hubness problem in the general machine learning domain [18], and it impairs the utility of pre-trained word embedding. To address this problem, several post-process debiasing methods have been proposed with respect to different bias scopes, such as local bias [19] and global bias [20]. Recently, Kaneko and Bollegala [21] proposed a debiasing method that uses an autoencoder. Extending [22], which was proposed to debias pre-trained word embeddings prior to integrating them into MMT models for English–German and English–French translations, we examined the latest debiasing techniques in more language pairs.

Moreover, despite the application of subword-level tokenization over NLP tasks, its impact on pre-trained word embeddings and downstream tasks has been less quantified. Specifically, reusing subwords learned during pre-training in downstream tasks is a well-established strategy in recent emergent language models [23]. Thus, we hypothesize that reusing subwords also benefits tasks using pre-trained word embeddings. To this end, we conducted experiments in which the training data for pre-training word embeddings were tokenized at either the word or subword level.

Our main findings on the use of pre-trained word embeddings are as follows:

- 1) Integrating pre-trained word embeddings into MMT models improves the translation performance, and applying a debiasing method further boosts the gains among a wide range of language pairs and MMT models.
- 2) Models that reuse the subwords of pre-trained word embeddings consistently outperform their counterpart, showing the best translation performance when debiasing is applied.

## B. LANGUAGE MODEL

The pre-trained language model (LM) improves the performance of a target model for different NLP tasks, such as text summarization [24], grammatical error correction [25], and machine translation [26]. The bidirectional encoder representations from Transformers (BERT) [23] comprises the encoder of the transformer [27] architecture, and it learns the general LM through pre-training on large-scale corpora. This pre-trained BERT is then fine-tuned on downstream tasks [23] or serves as a feature extractor [24]–[26]. Inspired by pre-trained LMs, many pre-trained LMs of vision–language modalities have been proposed. For example, the LXMERT [28] model incorporates object-level visual features into a BERT-like architecture, and it achieves state-of-the-art results in visual question-answering tasks.

Several researchers have incorporated VLM into MMT models. Reference [11] proposed to use a pre-trained MT

model to translate the captions of images and train a VLM on the pseudo multimodal parallel data, followed by fine-tuning on an MMT dataset. [29] pre-trained an MMT-specific visual feature extractor and incorporated the extracted visual feature into MMT models along with the BERT feature. Those approaches achieved substantial improvement, however, they required at least one expensive parallel corpus or high computational resource, which is unaffordable for training a model in a low-resource language or domain.

To address this problem, we propose a multimodal transformer model fused with a general-purpose pre-trained LXMERT system that utilizes the multimodal feature beyond its visual and textual features. The proposed model utilizes both textual and visual knowledge from the multimodal feature to improve the translation performance, which is unattainable solely with textual modality. The results obtained indicate the following:

- 1) The LXMERT-fused model improves the translation quality, especially under a limited language context.
- 2) An extensive analysis of our model indicates that a model fused with both LXMERT and BERT features further benefits the translation quality and still has room to fully exploit LXMERT and BERT features.

The remainder of this paper is organized as follows. Section II briefly discusses related work. Sections III and IV describe conventional MMT models with a pre-trained word embedding and the proposed MMT model fused with a visual–language LM, respectively. Sections V and VI respectively describe the relevant experiments conducted and analyze the results obtained in detail. Finally, concluding remarks are presented and future work is outlined.

## II. RELATED WORK

### A. MMT WITH DATA AUGMENTATION

Data augmentation is widely studied in MMT owing to the lack of available large-scale corpora. Most previous studies adopted textual parallel corpora as external data for Multi30K [1], a well-established multimodal parallel corpus. Reference [8] proposed to pre-train an NMT model on a large-scale textual parallel corpus (OpenSubtitles corpus [9]) and translated a multimodal monolingual corpus (MS-COCO [10]) to augment the data for training MMT models. Although their system trained using the augmented training data achieved state-of-the-art results in both English–French and English–German translation, the improvement made by images is moderate or even negative. Reference [30] combined four different pseudo-parallel and parallel corpora as additional data to train MMT models: back-translation of Multi30K Task 2 data and in-domain monolingual corpora of target language, in-domain data extracted from general domain parallel corpora, and general domain parallel data. Recently, [11] proposed a VTLM to aggregate the recent development of TLM and VLM. As training VTLM requires multimodal parallel data, they employed a publicly available NMT model trained on multiple large-scale parallel corpora. Reference [29] acquired

approximately six million sets of image-caption data that they used to pre-train an object recognition model, followed by fine-tuning on Multi30K data concurrently with training an MMT model.

In most studies on data augmentation in the MMT domain, either large-scale out-of-domain textual parallel corpora [8], [30] or pre-trained NMT models [11] of good quality for back-translation [31] is mandatory. Some studies require a large computational resource [29], which is considered overabundance to train an MMT model. With this constraint, the approaches cannot be adopted in low-resource domains. Thus, we explore the use of either textual or multimodal monolingual data, which tend to be available.

### B. MONOLINGUAL CORPORA AUGMENTED NMT

Knowledge learned from monolingual corpora has been widely exploited in NMT. Using word embedding or a language model is a common method to incorporate monolingual corpora in NMT systems.

Reference [13] made the first attempt to utilize pre-trained word embedding in low-resource NMT. They revealed that using a pre-trained word embedding to initialize embedding layers in an MT system improves the translation quality. Further, they stated that their approach is more effective for more similar language pairs. Recently, [32] proposed a search-based NMT model that predicts the embedding of the output word, rather than the distribution over the vocabulary. Their approach achieves not only faster training convergence without decreasing translation performance, but also a more accurate translation of rare words. This technique has been extended to an MMT model by [15], with resulting improvement in translation quality.

BERT [23] and its derivations [33], [34] employ a Transformer [27] architecture in a self-supervised learning manner and have achieved new state-of-the-art results on several natural language processing tasks. Moreover, several studies that leverage BERT for NMT have been published. Previous studies have revealed that simply initializing a transformer encoder using pre-trained BERT parameters does not improve translation quality [26], [35]. To address this problem, [35] proposed two-stage curriculum learning, in which model parameters initialized using pre-trained BERT are frozen until convergence in the first stage; all model parameters are then fine-tuned in the second stage. Meanwhile, [26] proposed a BERT-fused model that incorporates the representations from pre-trained BERT into a transformer model by feeding it into all the encoder and decoder layers. In both studies, models with BERT features outperformed naive transformer models to a substantial degree. Table 1 shows the dataset used to train each model.

Recently, many studies on self-supervised learning for vision–language tasks have been conducted. Similar to BERT, the models in these studies are first pre-trained on a large-scale text-image dataset, and then they are fine-tuned on downstream tasks. Table 2 shows the performance for the downstream tasks of each model that has publicly

TABLE 1. Datasets used to train NMT/MMT models.

Model	Multi30k		General-domain parallel data	Monolingual data	
	Bi-text	Image		Text	Image
NMT [3, 27]	✓				
BERT-fused NMT [26]	✓			✓	
MMT	✓	✓			
MMT w/ back-translation [8, 30]	✓	✓	✓	✓	✓
VTLM [11]	✓	✓	✓	✓	✓
MMT w/ embedding prediction ([15], Section. III)	✓	✓		✓	✓
LXMERT-fused MMT (Section. IV)	✓	✓		✓	✓

TABLE 2. Comparison of pre-trained LM in vision–language modalities. “Images” and “Sentences” denote the size of the data for pre-training LMs. “VQA” shows the overall accuracy on the test–dev split in VQA v2 [36].

Model	Images	Sentences	VQA
LXMERT [28]	180K	9.2M	72.42
ViLBERT [37]	3.1M	3.1M	70.55
Unified VLP [38]	3.1M	3.1M	70.50
VisualBERT [39]	328K	2.5M	70.80
VL-BERT [40]	3.1M	3.1M	70.50
UNITER [41]	4.2M	9.5M	72.27

available pre-trained models. Inspired by the progress in vision–language LM, we explore a Transformer-based MMT model incorporating vision–language LMs.

### III. MULTIMODAL MACHINE TRANSLATION WITH PRE-TRAINED WORD EMBEDDING

In this section, we present our proposal for exploiting pre-trained word embedding for conventional MMT models. Although pre-trained word embedding has been widely used in NMT tasks after the emergence of [13], there is still room to improve their effectiveness by alleviating the following problems: (a) Learning word embedding of good quality is challenging for rare words. (b) Some words frequently appear in the nearest neighbors of other words irrespective of their similarity. To this end, our proposed approach comprises five steps:

- 1) Tokenizing monolingual data at either the word or subword level
- 2) Pre-training word embedding using a model
- 3) Applying the debiasing process to remove hubness from pre-trained word embedding
- 4) Initializing the embedding of MMT encoder and decoder to the pre-trained word embedding
- 5) Training the MMT model on the Multi30K dataset

To show that our approach is invariant with the architecture of MMT models, we employed four different MMT models: decoder initialization [42] (“DECINIT”), IMAGINATION [43] (“IMAG+”), hierarchical-attention NMT [44] (“HA-NMT”), and visual attention grounding NMT [45] (“VAG-NMT”).

### A. CONVENTIONAL MMT MODELS

Conventional MMT models are based on the attentive recurrent neural network. All of these models handle machine translation as a sequence-to-sequence learning problem in which a neural model is trained to translate a source sentence of  $N$ -tokens  $x = \{x_1, x_2, \dots, x_N\}$  into a target sentence of  $M$ -tokens  $y = \{y_1, y_2, \dots, y_M\}$  along with the global visual feature  $v_g$  and/or the local visual feature  $v_l$ .

The underlying text-only NMT model of all MMT models comprises a bidirectional gated recurrent unit (GRU) [46] encoder and a unidirectional GRU decoder. The encoder first maps the source sentence  $x$  into the encoder hidden state  $\mathbf{h} = \mathbf{h}_0, \dots, \mathbf{h}_N$ , which is a concatenation of outputs from the forward GRU encoder and the backward GRU encoder.

Thereafter, the decoder computes a hidden state proposal  $s_j$  for each time step  $j \in [1, M]$ :

$$s_j = \text{GRU}(\hat{s}_{j-1}, e_{\text{dec}}(\hat{y}_{j-1})) \quad (1)$$

where  $\hat{s}_{j-1}$  is the previous hidden state and  $e_{\text{dec}}(\hat{y}_{j-1})$  is the embedding for the previous output word  $\hat{y}_{j-1}$ . The initial state  $\hat{s}_0$  is set to a zero vector.

The textual context vector is computed using an attention mechanism, given  $s_j$  as the query and  $\mathbf{h}_i$  as the key and value. Technically, in each time step  $j$  while decoding, a feed-forward layer is used to calculate a normalized soft alignment  $\alpha_{j,i}$  with each source hidden state  $\mathbf{h}_i$ , and the textual context vector  $\mathbf{c}_j^t$  is computed as the weighted sum of the source hidden states:

$$z_{j,i}^t = \mathbf{v}_t \tanh(\mathbf{U}_\alpha^t s_j + \mathbf{W}_\alpha^t \mathbf{h}_i) \quad (2)$$

$$\alpha_{j,i}^t = \frac{\exp(z_{j,i}^t)}{\sum_{k=1}^N \exp(z_{j,k}^t)} \quad (3)$$

$$\mathbf{c}_j^t = \sum_{i=1}^N \alpha_{j,i}^t \mathbf{h}_i \quad (4)$$

where  $\mathbf{v}_t$ ,  $\mathbf{U}_\alpha^t$ , and  $\mathbf{W}_\alpha^t$  are model parameters.

The decoder employs another GRU unit to compute the final hidden state  $\hat{s}_j$  from the hidden state proposal  $s_j$ , textual context  $\mathbf{c}_j^t$ , and visual context  $\mathbf{c}_j^v$ :

$$\hat{s}_j = \text{GRU}(s_j, \mathbf{c}_j^t) \quad (5)$$

The system output at time-step  $j$  is obtained using the current hidden state, previous word embedding, textual context, and visual context:

$$\mathbf{o}_j = \tanh(\mathbf{L}^s \hat{\mathbf{s}}_j + \mathbf{L}^w \mathbf{e}_{\text{dec}}(\hat{y}_{j-1}) + \mathbf{L}^t \mathbf{c}_j^t) \quad (6)$$

$$p(w|\hat{y}_{<j}) = \text{softmax}(\mathbf{o}_j) \quad (7)$$

$$\hat{y}_j = \underset{w \in \mathcal{V}}{\text{argmax}} \{p(w|\hat{y}_{<j})\} \quad (8)$$

where  $\mathbf{L}^s$ ,  $\mathbf{L}^w$ , and  $\mathbf{L}^t$  are model parameters.

### 1) DECODER INITIALIZATION

This MMT model uses a projected global visual feature as the initial decoder state  $\hat{\mathbf{s}}_0$ , rather than a zero vector:

$$\hat{\mathbf{s}}_0 = \sigma(\mathbf{W}_0 \mathbf{v}_g + \mathbf{b}_0) \quad (9)$$

where  $\mathbf{W}_0$  and  $\mathbf{b}_0$  are model parameters.

### 2) HIERARCHICAL-ATTENTION NMT

Hierarchical-attention NMT [44] incorporates the local visual feature in an attentive manner. The model first computes the textual context vector  $\mathbf{c}_j^t$  and the visual context vector  $\mathbf{c}_j^v$  using two individual attention units, as described by (2)–(4). The concatenation of two context vectors  $\{\mathbf{c}_j^t, \mathbf{c}_j^v\}$  is then fed into another attention as the key and value, where the hidden state proposal  $\mathbf{s}_j$  is used as the query. The second GRU in (5) takes the obtained multimodal context vector, rather than the textual context vector  $\mathbf{c}_j^t$ .

### 3) IMAGINATION

IMAGINATION [43] is a multitask model that jointly learns machine translation and visual latent space models. The MT model is the vanilla NMT model that does not involve any visual features; this model does not use images during inferring. The latent space model is a feed-forward model that calculates the average vector over the hidden states  $\mathbf{h}_i$  in the encoder and maps it to the final vector  $\hat{\mathbf{v}}$  in the latent space:

$$\hat{\mathbf{v}}_g = \tanh(\mathbf{W}_v \cdot \frac{1}{N} \sum_i^N \mathbf{h}_i), \quad (10)$$

where  $\mathbf{W}_v$  is a model parameter.

We employ the max-margin loss to train the latent space model to ensure that the model maps the encoder hidden states closer to the global visual feature:

$$\sum_{v' \neq v} \max\{0, \alpha - d(\hat{\mathbf{v}}_g, \mathbf{v}_g) + d(\hat{\mathbf{v}}_g, \mathbf{v}'_g)\}, \quad (11)$$

where  $d$  is a cosine similarity function and  $\alpha$  is the margin.<sup>1</sup> The final loss is the sum of the losses for MT and latent space learning.

<sup>1</sup>We use  $\alpha = 0.1$  in our experiment.

### 4) VISUAL ATTENTION GROUNDING NMT

Visual attention grounding NMT (VAG-NMT) [45] is another multitask model comprising an MMT model and a latent space model.

The model first computes the sentence representation  $\mathbf{t}$  from the global visual feature  $\mathbf{v}_g$  and encoder hidden states  $\mathbf{h}$ :

$$z_i = \tanh(\mathbf{W}_v \mathbf{v}_g) \cdot \tanh(\mathbf{W}_h \mathbf{h}_i) \quad (12)$$

$$\beta_i = \frac{\exp(z_i)}{\sum_{k=1}^N \exp(z_k)} \quad (13)$$

$$\mathbf{t} = \sum_{i=1}^N \beta_i \mathbf{h}_i \quad (14)$$

Thereafter, VAG-NMT utilizes the sentence representation to initialize the decoder hidden state:

$$\hat{\mathbf{s}}_0 = \tanh(\mathbf{W}_{\text{init}}(\rho \mathbf{t} + (1 - \rho) \frac{1}{N} \sum_i^N \mathbf{h}_i)) \quad (15)$$

where  $\mathbf{W}_{\text{init}}$  is a model parameter and  $\rho$  is a hyperparameter for determining the text representation ratio in the decoder initial state.<sup>2</sup>

Further, the model learns to map the sentence representation  $\mathbf{t}$  and the global visual feature  $\mathbf{v}_g$  closer in a latent space:

$$\mathbf{t}_{\text{emb}} = \tanh(\mathbf{W}_t \mathbf{t} + \mathbf{b}_t) \quad (16)$$

$$\mathbf{v}_{\text{emb}} = \tanh(\mathbf{W}_v \mathbf{v}_g + \mathbf{b}_v) \quad (17)$$

The loss for latent space learning is the max-margin loss with negative sampling:

$$\sum_p \sum_k \max\{0, \gamma - d(\mathbf{v}_{g,p}, \mathbf{t}_p) + d(\mathbf{v}_{g,p}, \mathbf{t}_{k \neq p})\} + \sum_k \sum_p \max\{0, \gamma - d(\mathbf{t}_k, \mathbf{v}_{g,k}) + d(\mathbf{t}_k, \mathbf{v}_{g,k \neq p})\} \quad (18)$$

where  $d$  is the cosine similarity function;  $k$  and  $p$  are the indexes for sentences and images, respectively;  $\mathbf{t}_{k \neq p}$  are the negative samples for which all examples in the same batch with the target example are selected; and  $\gamma$  is the margin that adjusts the sparseness of each item in the latent space.<sup>3</sup>

The final loss is the weighted sum of losses for MT and latent space learning.

### B. TOKENIZATION

The distribution of word occurrence follows Zipf's law and is long-tailed, where a few common tokens dominate and most tokens appear several times. Consequently, the obtained word embedding for tokens lying on the long tail may have poor quality. Improving the word embedding for rare words intuitively requires relaxing the slope of the distribution. To this end, we propose learning the word embedding from sentences that are tokenized at the subword level. Technically, we employ Byte Pair Encoding [47] (BPE) to tokenize words into subwords.

<sup>2</sup>We use  $\rho = 0.5$  in our experiment.

<sup>3</sup>We use  $\gamma = 0.1$  in our experiment.



For training the MMT models, we adopt the same subword vocabulary used in pre-training word embedding to tokenize Multi30K sentences.

### C. DEBIASING PRE-TRAINING WORD EMBEDDING

The geometry of pre-trained word embeddings is anisotropic, and the word representations are distributed locally in a conical subspace. This geometry causes problems such as hubness [16], [17] and undermines the usefulness of pre-trained word embeddings. To address this problem, we employ three different debiasing methods as follows: localized centering [19] (“LC”), All-but-the-Top [20] (“AbtT”), and an approach using an autoencoder [21] (“AE”).

#### 1) LOCALIZED CENTERING

Localized centering shifts each word based on its local bias. The local centroid for each word  $x$  is computed and subtracted from the original word  $x$  to obtain the new embedding  $\hat{x}$ :

$$c_k(x) = \frac{1}{k} \sum_{x' \in k\text{NN}(x)} x' \quad (19)$$

$$\hat{x} = x - c_k(x), \quad (20)$$

where  $k$  is a hyperparameter called the local segment size<sup>4</sup> and  $k\text{NN}(x)$  returns the  $k$ -nearest neighbors of the word  $x$ .

#### 2) ALL-BUT-THE-TOP

All-but-the-Top uses the global bias of the entire vocabulary to shift the embedding of each word. The All-but-the-Top algorithm comprises three steps: subtract the centroid of all words from each word  $x$ , compute the PCA components for the centered space, and subtract the top  $n$  PCA components from each centered word to obtain the final word  $\hat{x}$ :

$$x' = x - \frac{1}{|\mathcal{V}|} \sum_{w \in \mathcal{V}} w \quad (21)$$

$$u_1, u_2, \dots, u_D = \text{PCA}(x' \in \mathcal{V}) \quad (22)$$

$$\hat{x} = x' - \sum_{i=1}^D (u_i^\top x') u_i, \quad (23)$$

where  $D$  is a hyperparameter used to determine how many principal components of the pre-trained word embedding are ignored.<sup>5</sup>

#### 3) AUTOENCODER

Reference [21] showed that applying centering and PCA is the same as applying an autoencoder. Following their approach, we first train an autoencoder upon a pre-trained word embedding using the  $L_2$  reconstruction loss. Subsequently, we extract the hidden state of each word embedding in the autoencoder as the revised word embedding. The centering and PCA effects of the autoencoder result in the obtained word embeddings being both debiased and isotropic.

<sup>4</sup>We use  $k = 10$  in our experiment.

<sup>5</sup>We use  $D = 3$  in our experiment.

## IV. MULTIMODAL MACHINE TRANSLATION WITH LM

In this study, we use a variant of the BERT-fused model [26], namely, the LXMERT-fused model, in which the LXMERT [28] system is used as the feature extractor instead of the BERT system. As the performance of the LXMERT system is the best among available pre-trained systems (Table 2), we employ LXMERT as our feature extractor.<sup>6</sup>

This model tackles MMT as a sequence-to-sequence learning problem, in which a neural network model is trained to translate a source sentence of  $N$ -tokens  $\mathbf{x} = \{x_1, \dots, x_N\}$  and the corresponding image into the target sentence of  $M$ -tokens  $\mathbf{y} = \{y_1, \dots, y_M\}$ .

### A. LXMERT

The LXMERT model is a pre-trained VLM that represents cross-modal connections, as well as each modality. It takes both a sentence  $\mathbf{x}$  and an image as its inputs and generates two different features: language output  $\mathbf{H}_L$  and vision output  $\mathbf{H}_R$ .

More specifically, LXMERT first encodes each modality using individual single-modality encoders, and then it encodes cross-modality connections using another individual encoder with a cross-modality attention mechanism. The LXMERT model is pre-trained with four tasks: masked cross-modality LM, masked object prediction (feature vector regression and detected-label classification), cross-modality matching, and image question answering.

#### 1) SINGLE-MODALITY LANGUAGE ENCODER

The single-modality language encoder is a BERT-like encoder. It is composed of a stack of nine layers, with each layer containing a self-attention sub-layer and a feed-forward sub-layer. Two special tokens, CLS and SEP, are appended before and after the input sentence words, respectively. The resulting input sequence  $\{\text{CLS}, x_1, \dots, x_N, \text{SEP}\}$  is fed into the encoder to generate the language-modality representation  $\mathbf{h}^0$ .

#### 2) SINGLE-MODALITY VISION ENCODER

The single-modality vision encoder is also a BERT-like encoder. However, it is composed of a stack of five layers and takes visual features as inputs. Rather than using a raw image, LXMERT uses a bag-of-objects representation of  $K$  objects  $\{(p_1, f_1), \dots, (p_K, f_K)\}$  that a Faster R-CNN [48] system detects from the image. Here,  $p_j$  is the position feature, and  $f_j$  is the region-of-interest (RoI) feature for  $j \in [1, K]$ . The position-aware embedding  $o_i$  for each object is derived from the position and RoI features:

$$\hat{p}_j = \text{LayerNorm}(W_P p_j + b_P) \quad (24)$$

$$\hat{f}_j = \text{LayerNorm}(W_F f_j + b_F) \quad (25)$$

$$o_j = (\hat{p}_j + \hat{f}_j) / 2 \quad (26)$$

where  $\text{LayerNorm}$  is the layer-normalization function and  $W_F$ ,  $b_F$ ,  $W_P$ , and  $b_P$  are model parameters.

<sup>6</sup>We plan to release our implementation upon publication.

The position-aware embeddings  $\mathbf{o} = \{o_1, \dots, o_K\}$  are fed into the single-modality encoder to deliver vision-modality representation  $\mathbf{v}^0$ .

### 3) CROSS-MODALITY ENCODER

The cross-modality encoder is composed of a stack of five identical layers. Each cross-modality layer consists of two unidirectional cross-attention, two self-attention, and two feed-forward sub-layers. In the  $k$ -th layer, two unidirectional cross-attention sub-layers are first applied—one from language to vision and the other from vision to language. The query and context vectors are the outputs of the  $(k - 1)$ -th layer:

$$\hat{h}_i^k = \text{Head}_{h \rightarrow v}^k(h_i^{k-1}, \mathbf{v}^{k-1}, \mathbf{v}^{k-1}) \quad (27)$$

$$\hat{v}_j^k = \text{Head}_{v \rightarrow h}^k(v_j^{k-1}, \mathbf{h}^{k-1}, \mathbf{h}^{k-1}) \quad (28)$$

where  $\text{Head}_{h \rightarrow v}^k$  and  $\text{Head}_{v \rightarrow h}^k$  are two different multi-headed attention modules.

Subsequently, we apply the self-attention sub-layers to the output of the cross-attention sub-layers, which is followed by the feed-forward sub-layers to obtain the  $k$ -th layer outputs  $\mathbf{h}^k$  and  $\mathbf{v}^k$ .

### 4) OUTPUT REPRESENTATIONS

The corresponding outputs of the last cross-modality encoder are denoted  $\mathbf{H}_L$  for language and  $\mathbf{H}_R$  for objects. Technically, we use  $\mathbf{h}^5$  as  $\mathbf{H}_L$  and  $\mathbf{v}^5$  as  $\mathbf{H}_R$ .

## B. LXMERT-FUSED MODEL

The LXMERT-fused model takes the LXMERT representations as the embedding of the images. In addition, we use a concatenation of  $\mathbf{H}_L$  and  $\mathbf{H}_R$  as the LXMERT representations  $\mathbf{H}_{LR} = [\mathbf{H}_L \text{ and } \mathbf{H}_R]$  to ensure that the model can exploit both language-to-vision and vision-to-language cross-modality information.

### 1) ENCODER

The encoder is composed of a stack comprising one embedding layer and six encoder layers. Each encoder layer contains one fusion sub-layer and one feed-forward sub-layer. A residual connection is applied around each of the two sub-layers, and then layer normalization proceeds.

The encoder first projects tokens in a sentence  $\mathbf{x} = \{x_1, \dots, x_N\}$  to vectors via the embedding layer, followed by a tanh activation. It then injects positional encoding into the input embedding and applies layer normalization to obtain the position-aware word embeddings  $\mathbf{H}_E^0 = \{H_{E,1}^0, \dots, H_{E,N}^0\}$ :

$$H_{E,i}^0 = \text{LayerNorm}(\tanh(e_{\text{enc}}(x_i)) + \text{PE}(i)) \quad (29)$$

where  $i \in [1, N]$  denotes each position in a source sentence,  $e_{\text{enc}}(x_i)$  is the embedding representation for a word  $x_i$ , and  $\text{PE}(i)$  is the positional embedding for a position  $i$ .

In the  $l$ -th encoder layer, the fusion sub-layer is first applied, where two context vectors are computed using two

different attention modules: self-attention and encoder-to-lxmert attention. The fusion sub-layer then interpolates two context vectors and obtains the final context vector  $\tilde{\mathbf{H}}_E^l$ :

$$\tilde{H}_{E,i}^l = \lambda_E \text{Head}_{E \rightarrow E}^l(H_{E,i}^{l-1}, \mathbf{H}_E^{l-1}, \mathbf{H}_E^{l-1}) + \lambda_{LR} \text{Head}_{E \rightarrow LR}^l(H_{E,i}^{l-1}, \mathbf{H}_{LR}, \mathbf{H}_{LR}) \quad (30)$$

where  $\lambda_E$  and  $\lambda_{LR}$  are interpolation coefficients<sup>7</sup> and  $\lambda_E + \lambda_{LR} = 1$  and  $\text{Head}_{E \rightarrow E}^l$  and  $\text{Head}_{E \rightarrow LR}^l$  are multi-head attention modules with different parameters. Dropnet [26] is applied for all fusion sub-layers.

The context vectors are then processed by the position-wise feed-forward sub-layers, and the output of the  $l$ -th layer  $\mathbf{H}_E^l$  is derived:

$$H_{E,i}^l = \text{ReLU}(\tilde{H}_{E,i}^l W_1^l + b_1^l) W_2^l + b_2^l \quad (31)$$

where  $W_1^l$ ,  $W_2^l$ ,  $b_1^l$ , and  $b_2^l$  are the model parameters, and  $\text{ReLU}$  is a  $\text{ReLU}$  activation.

### 2) DECODER

The decoder is also composed of a stack comprising one embedding layer and six decoder layers. Each decoder layer contains one self-attention sub-layer, one fusion sub-layer, and one feed-forward sub-layer. The residual connection and layer normalization are applied between sub-layers.

In each position  $t$ , while decoding, the decoder first computes the position-aware word embeddings  $\mathbf{H}_{D,<t}^0 = \{H_{D,1}^0, \dots, H_{D,t-1}^0\}$  from the predicted tokens  $\hat{\mathbf{y}}_{<t} = \{\hat{y}_1, \dots, \hat{y}_{t-1}\}$ :

$$H_{D,j}^0 = \text{LayerNorm}(\tanh(e_{\text{dec}}(\hat{y}_j)) + \text{PE}(j)) \quad (32)$$

where  $j \in [1, t - 1]$  denotes each position in the predicted tokens, and  $e_{\text{dec}}(\hat{y}_j)$  is the embedding representation for a word  $\hat{y}_j$ .

In the  $l$ -th decoder layer, the output of the  $l - 1$ -th layer is fed to a self-attention module  $\text{Head}_{D \rightarrow D}$  to generate the intermediate representation  $\hat{H}_{D,j}^l$ :

$$\hat{H}_{D,j}^l = \text{Head}_{D \rightarrow D}^l(H_{D,j}^{l-1}, \mathbf{H}_{D,<t}^{l-1}, \mathbf{H}_{D,<t}^{l-1}) \quad (33)$$

The fusion layer in the decoder works similar to that in the encoder; however, it uses decoder-to-encoder attention (rather than self-attention) to generate the final representation  $\tilde{H}_{D,j}^l$ :

$$\tilde{H}_{D,j}^l = \rho_E \text{Head}_{D \rightarrow E}^l(\hat{H}_{D,j}^l, \mathbf{H}_{D,<t}^{l-1}, \mathbf{H}_{D,<t}^{l-1}) + \rho_{LR} \text{Head}_{D \rightarrow LR}^l(\hat{H}_{D,j}^l, \mathbf{H}_{LR}, \mathbf{H}_{LR}) \quad (34)$$

where  $\rho_E$  and  $\rho_{LR}$  are the interpolation coefficients<sup>8</sup> and  $\rho_E + \rho_{LR} = 1$ . Further,  $\text{Head}_{D \rightarrow E}^l$  and  $\text{Head}_{D \rightarrow LR}^l$  are multi-head attention modules with different parameters. Dropnet is also applied for all fusion sub-layers in the decoder.

<sup>7</sup>We use 0.5 for both  $\lambda_E$  and  $\lambda_{LR}$  in our experiments.

<sup>8</sup>We use 0.5 for both  $\rho_E$  and  $\rho_{LR}$  in our experiments.

The context vectors are then processed by the position-wise feed-forward sub-layers, and the output of the  $l$ -th layer  $\mathbf{H}_E^l$  is derived:

$$H_{D,j}^l = \text{ReLU}(\tilde{H}_{D,j}^l W_3^l + b_3^l) W_4^l + b_4^l \quad (35)$$

where  $W_3^l$ ,  $W_4^l$ ,  $b_3^l$ , and  $b_4^l$  are the model parameters.

The output of the last decoder layer  $\mathbf{H}_D^6$  is fed into the projection layer to generate the output distribution  $p(\hat{y}_t | \hat{y}_{<t})$ :

$$p(\hat{y}_t | \hat{y}_{<t}) = \text{softmax}(H_{D,t-1}^6 W_5 + b_5) \quad (36)$$

where  $W_5$  and  $b_5$  are the model parameters. In particular,  $W_5$  is a projection matrix that maps the decoder state into vocabulary space.

### C. TRAINING

A preliminary study, reported by [26], has indicated that training LXMERT-fused models from scratch does not lead to a good model performance, which is also while using the BERT feature.

To address this problem, we employ a two-step procedure to train LXMERT-fused models. We first train an LXMERT-fused model with  $\rho_E = 0$ , where only the language part of the training data is included. After the model has converged, we then set  $\rho_E$  for a specific value to fine-tune the model on both language and vision data. During fine-tuning, the learning rate and batch size are set to smaller values than those in the first step.

## V. EXPERIMENT

### A. WORD EMBEDDING

In our experiments, we used three different well-established word embedding models: word2vec [49], GloVe [50], and FastText [14]. The publicly available pre-trained word embeddings use different training corpora; however, we trained the word embeddings of different models using an identical monolingual corpus for fair comparison.

#### 1) TRAINING CORPUS

We downloaded Wikidump<sup>9</sup> for English, German, French, and Czech and extracted the article pages. All the extracted sentences were preprocessed by lower-casing, tokenizing, and normalizing the punctuation using a Moses script.<sup>10</sup>

For the subword-level experiments, we used Byte Pair Encoding to split words into subwords. We used subword-nmt<sup>11</sup> to process the sentences. The number of merge operations was 30,000, and the vocabulary threshold was set to zero. Table 3 shows the statistics of the preprocessed Wikipedia corpus for each language.

We applied each debiasing method to the obtained word embeddings with the same options as in its original paper.

<sup>9</sup><https://dumps.wikimedia.org/>. We used the July 20, 2020 version for English, German, and French and the December 20, 2020 version for Czech and Japanese.

<sup>10</sup>We used a script from Multi30K to preprocess the sentences. <https://github.com/multi30k/dataset/blob/master/scripts/task1-tokenize.sh>

<sup>11</sup><https://github.com/rsennrich/subword-nmt>

TABLE 3. Statistics of wikidump corpus for each language.

Language	Lines	Word		BPE	
		Tokens	Types	Tokens	Types
English	45.5M	2,590M	7.9M	2,917M	57.0K
German	18.6M	936M	8.5M	1,173M	44.7K
French	16.7M	820M	3.7M	931M	47.7K
Czech	4.3M	144M	2.8M	191M	39.0K
Japanese	10.1M	602M	2.6M	656M	59.5K

#### 2) TRAINING SETTING

All word embeddings were trained on a dimension of 300. The specific options for training were as follows (default values were used for other options).

We trained the word2vec model<sup>12</sup> using the CBOW algorithm (with window size of 10, negative sampling of 10, and minimum count of 10), the GloVe model<sup>13</sup> (with window size of 10 and minimum count of 10), and the FastText model<sup>14</sup> using the CBOW algorithm (with maximum character n-gram of 5, window size of 5, and negative sampling of 10).

#### 3) UNKNOWN WORDS

Unknown words are of two types: words that are a part of a pre-trained word embedding but are not included in a vocabulary (Out-Of-Vocabulary (OOV) words) and words that are a part of a vocabulary but are not included in pre-trained word embedding (OOV words for embedding). OOV words for embedding only exist when using word-level embedding (word2vec and glove); the embedding of such words in FastText are calculated as the mean embedding of character n-grams consisting of the word.

The embeddings for both types of OOV words were calculated as the average embedding over the words that were a part of the pre-trained word embedding, but were not included in the vocabularies, and they were updated individually during training.

### B. MODEL

Tables 4 and 5 show the hyperparameters of the conventional and LXMERT-fused MMT models in our experiments, respectively. Note that each conventional MMT model has an encoder size of 320; therefore, the size of bidirectional GRU is 640. All models were implemented using the nmttorch toolkit v4.0.0 [51].

#### 1) GLOBAL AND LOCAL VISUAL FEATURE

We encoded each image using pre-trained ResNet-50 [52] and selected the hidden state in the res4f layer of 1024D as its global visual feature, and that in the pool5 layer of 2048D as its local visual feature.

<sup>12</sup><https://github.com/tmikolov/word2vec>.

<sup>13</sup><https://github.com/stanfordnlp/GloVe>.

<sup>14</sup><https://github.com/facebookresearch/fastText>.



**TABLE 4.** Hyperparameters of conventional MMT models.

Hyperparameter	Value
Embedding size	300
Encoder size	320
Encoder layers	2
Decoder size	320
Shared Space (IMAGINATION)	2048
Shared Space (VAG-NMT)	512
Optimizer	Adam
Batch size	64 sentences
Learning rate	0.0004
Gradient Clipping	1.0
L <sub>2</sub> regularization	0.00001
Dropout (embedding)	0.4
Dropout (context)	0.5
Dropout (output)	0.5
Maximum source length	100
Maximum output length	100
Beam size	12

**TABLE 5.** Hyperparameters of the LXMERT-fused MMT model.

Hyperparameter	Value
Embedding size	512
Encoder size	512
Encoder layers	6
Decoder size	512
Decoder layers	6
Optimizer	Adam in [27]
Batch size (pre-train)	4,000 tokens
Batch size (fine-tune)	1,000 tokens
Learning rate (pre-train)	0.044
Learning rate (fine-tune)	0.01
L <sub>2</sub> regularization	Disabled
Dropout	0.3
Beam size	5

## 2) LXMERT FEATURE

We used a publicly available LXMERT model<sup>15</sup> in our experiment. We first employed the alternative pre-trained Faster R-CNN model<sup>16</sup> to encode all images in the Multi30K dataset and selected 36 RoI features of the 2048-dimension and 36 positional features of the four-dimension for each image in the same manner as [28]. Finally, the pre-trained LXMERT model processed the selected visual features and the corresponding source sentence to obtain LXMERT features of the 768-dimension.

## 3) OTHER FEATURES

We also examined two types of features with LXMERT-fused models, in which the feature is fed into the model (rather than the LXMERT feature). The BERT feature (“BERT”) denotes the output of bert-base-uncased provided by PyTorch-Transformers.<sup>17</sup>

The all-inclusive feature (“All-inclusive”) is a concatenation of “LXMERT” and “BERT” features.

<sup>15</sup>[http://nlp.cs.unc.edu/data/model\\_LXRT.pth](http://nlp.cs.unc.edu/data/model_LXRT.pth)

<sup>16</sup><https://github.com/airsplay/lxmert#alternative-dataset-and-features-download-links>

<sup>17</sup><https://github.com/huggingface/pytorch-transformers>

**TABLE 6.** Number of tokens and types for each language in the Multi30K training set.

Language	Word		Subword	
	Tokens	Types	Tokens	Types
English	380,214	9,794	403,575	8,476
German	365,536	18,043	438,681	10,005
French	416,428	11,050	452,760	9,165
Czech	297,896	22,236	399,214	11,438

**TABLE 7.** Corpus-level BLEU scores of the 2016 test set for English–German translation. “English” and “German” show the tokenization strategies. The bold values are higher than the value of the Word–Word tokenization strategy.

English	German	BiGRU	Transformer
Word	Word	38.53	38.83
Word	BPE	<b>38.78</b>	38.82
BPE	Word	38.49	38.61
BPE	BPE	<b>39.33</b>	38.65

## C. MULTI30K DATASET

We used the Multi30K [1] dataset for all translation directions and the 2017 test set [53] for English–German and English–French translations. The training, validation, 2016 test, and 2017 test sets have 29,000, 1,014, 1,000, and 1,000 instances, respectively. We selected English as the source language and German/French/Czech as the target languages. All sentences in English/German/French/Czech were preprocessed by lower-casing, tokenizing, and normalizing the punctuation using the same scripts described in V-A.

For the subword-level experiments, we applied BPE using the subwords obtained from Wikipedia; consequently, no OOV tokens appeared in the training and other sets. Table 7 shows the results of the preliminary experiments. Considering the results, we decided to perform our experiments only on the BiGRU-based MMT models and omit BPE-to-word translation.

We also evaluated the LXMERT-fused models on a degraded version of Multi30K (2016<sub>N</sub>), where the first noun of each noun phrase is masked. Note that the models for the 2016<sub>N</sub> test set were trained on the degraded version of the training set.

## D. EVALUATION

We used BLEU [54] and METEOR [55] as our evaluation metrics. BLEU evaluates the hard matches on unigrams, bigrams, trigrams, and 4-grams between the system output and the reference. METEOR is a BLEU-like metric that employs WordNet [56] to relax the hard alignment between the prediction and reference, which allows the metric to take more account of semantics. Note that METEOR is only available for German, French, and Czech; we did not evaluate Japanese translation using METEOR. We trained each model three times using different seeds and averaged the scores.

**TABLE 8.** Corpus-level BLEU / METEOR on the 2016 test set for English–German translation using different tokenization strategies, pre-trained word embeddings, and MMT models. “+ debias” shows the best score of the three models using different debiasing methods. The underlined scores are higher than the score of randomly initialized models. The bold score is the best score of each model. “†” indicates the statistical significance of the improvement over randomly initialized models.

Word Embedding	Text-only		DECINIT		IMAG+		HA-NMT		VAG-NMT		
	BLEU	METEOR	BLEU	METEOR	BLEU	METEOR	BLEU	METEOR	BLEU	METEOR	
Word–Word	Random	38.53	57.77	38.38	57.73	38.09	57.41	37.79	57.31	37.80	57.11
	word2vec	<u>39.20</u>	<u>58.18</u>	<u>38.85</u>	<u>58.17</u> †	<u>38.32</u>	<u>58.11</u> †	<u>38.61</u>	<u>57.79</u> †	<u>38.14</u>	<u>57.17</u>
	GloVe	38.02	<u>57.87</u>	<u>38.41</u>	<u>57.88</u>	<u>38.48</u>	<u>57.82</u>	37.51	57.28	<u>37.84</u>	<u>57.38</u>
	FastText	26.55	47.94	26.11	47.32	26.33	47.77	33.87	54.26	37.64	56.99
	word2vec + debias	38.51	57.62	<u>39.20</u> †	<u>58.17</u> †	<u>39.13</u> †	<u>58.05</u> †	<u>38.52</u>	<u>57.82</u> †	<u>38.12</u>	<u>57.47</u>
	GloVe + debias	<u>38.77</u>	<u>57.98</u>	<u>38.61</u>	<u>57.89</u> †	<u>39.13</u> †	<u>58.08</u>	<u>38.50</u> †	<u>57.69</u> †	<u>38.07</u>	<u>57.43</u>
	FastText + debias	<u>38.63</u>	<u>57.86</u>	<u>38.71</u>	<u>57.97</u> †	<u>39.10</u> †	<u>58.03</u>	<u>38.34</u>	<u>57.75</u> †	<u>38.20</u>	<u>57.36</u>
Word–BPE	Random	38.78	57.98	39.12	58.16	39.10	57.91	38.07	57.48	39.06	57.78
	word2vec	<u>39.22</u>	<u>58.21</u>	<u>39.37</u>	<u>57.92</u>	<u>39.73</u>	<u>58.56</u> †	<u>39.04</u> †	<u>57.90</u>	<b>39.24</b>	<u>57.81</u>
	GloVe	<u>39.58</u>	<u>58.45</u>	<u>39.33</u>	<u>58.29</u>	<u>39.33</u>	<u>58.21</u>	<u>38.30</u>	<u>57.62</u>	38.67	57.56
	FastText	7.81	24.61	7.49	24.36	6.74	23.65	18.05	36.60	38.51	56.90
	word2vec + debias	<u>39.85</u>	<u>58.46</u>	<u>39.60</u>	<u>58.30</u>	<u>40.07</u> †	<u>58.56</u> †	<u>39.10</u>	<u>57.96</u>	38.77	<u>57.83</u>
	GloVe + debias	<u>39.74</u>	<u>58.41</u>	<b>39.87</b>	<b>58.61</b>	<u>39.97</u> †	<u>58.53</u> †	<u>39.07</u> †	<u>57.94</u> †	<u>39.15</u>	<b>57.95</b>
	FastText + debias	<u>39.67</u>	<u>58.35</u>	<u>39.68</u>	<u>58.48</u>	<b>40.10</b> †	<u>58.55</u> †	<u>39.17</u> †	<u>58.02</u> †	<u>39.18</u>	<u>57.85</u>
BPE–BPE	Random	39.33	57.70	39.73	58.21	39.22	57.79	38.78	57.50	38.60	57.42
	word2vec	<u>39.90</u> †	<u>58.42</u> †	39.54	<u>58.27</u>	<u>39.38</u>	<u>58.06</u>	<u>39.17</u>	<b>58.05</b>	<u>38.95</u>	<u>57.91</u>
	GloVe	<u>39.54</u>	<u>57.79</u>	38.92	57.83	<u>39.23</u>	57.70	38.30	57.29	38.49	57.19
	FastText	10.52	28.39	7.53	25.54	7.84	27.58	35.78	55.10	38.30	56.38
	word2vec + debias	<u>39.68</u> †	<u>58.23</u> †	39.67	<u>58.34</u>	<u>39.79</u>	<u>58.32</u>	<u>39.01</u>	<u>57.96</u>	<u>38.79</u>	<u>57.83</u>
	GloVe + debias	<b>39.96</b> †	<b>58.53</b> †	39.44	58.20	<b>40.10</b>	<u>58.43</u>	<u>39.12</u>	<u>57.97</u>	<u>38.97</u>	<u>57.74</u>
	FastText + debias	<u>39.76</u> †	<u>58.25</u> †	39.66	<u>58.26</u>	<u>39.76</u>	<u>58.37</u>	<b>39.41</b>	<u>58.05</u>	<u>38.95</u>	<u>57.85</u>

## E. RESULTS

### 1) MMT WITH PRE-TRAINED WORD EMBEDDING

Table 8 shows the BLEU and METEOR scores across the “text-only” model and MMT models for English–German translation. First, we observe that applying the Wikipedia-based BPE to both English sentences and German translation results in substantial improvement (+1.01 BLEU on average) for all models. Note that applying BPE to English sentences also boosts the model performance, which is contrary to the report by [57] that Multi30K-based BPE to source sentences is not beneficial. Second, debiasing pre-trained word embedding further improves the model performance. Given the use of BPE on both sides, models using debiased word embedding have a higher BLEU score than their counterparts that use vanilla word embedding.

We observed a slightly different trend for other translation pairs. Table 9 shows the BLEU scores of three models for English–French, English–Czech, and English–Japanese translation. Using debiased word embedding still results in improvements over randomly initialized models. However, Wikipedia-based BPE no longer benefits model performance (−0.37, +0.01, and −0.52 BLEU for English–French, English–Czech, and English–Japanese on average, respectively).

### 2) LXMERT-FUSED MMT

We trained all the models three times with different seeds and averaged the scores.

Table 10 shows the BLEU and METEOR scores across three test sets in Multi30K. Adding to the text-only

**TABLE 9.** Corpus-level BLEU on the 2016 test set for English–French, English–Czech, and English–Japanese translation. “+ debias” shows the best score of the three models using different debiasing methods. The bold score is the best score of each model. “†” indicates the statistical significance of the improvement over randomly initialized models.

Tokenization	Word Embedding	Text-only	DECINIT	IMAG+	
English–French					
Word–Word	Random	<b>61.87</b>	61.22	61.51	
	Random	61.33	60.82	61.34	
	word2vec + debias	61.38	61.28	61.72	
	GloVe + debias	61.35	61.48	61.61	
BPE–BPE	FastText + debias	61.58	<b>61.72</b> †	<b>61.77</b>	
	English–Czech				
	Word–Word	Random	32.43	31.91	32.07
		Random	32.02	32.09	32.32
word2vec + debias		<b>32.97</b> †	32.81†	32.82	
GloVe + debias		32.81†	33.17†	<b>32.88</b>	
BPE–BPE	FastText + debias	32.88†	<b>33.19</b> †	32.60	
	English–Japanese				
	Word–Word	Random	40.72	<b>40.65</b>	40.61
		Random	39.95	40.20	40.27
word2vec + debias		<b>40.73</b> †	40.49	<b>40.74</b>	
GloVe + debias		40.67†	40.42†	40.66	
BPE–BPE	FastText + debias	40.66†	40.58†	40.51	

Transformer and LXMERT-fused Transformer, we also conducted experiments on models fused with the BERT feature (“BERT”), ResNet-50 local visual feature (“ResNet-50”), and RoI feature (“Faster R-CNN”). We observed that the MMT models incorporating BERT (or all-inclusive feature)

**TABLE 10. Quantitative comparison of Transformer-based MMT models on the 2016, 2017, and 2016<sub>N</sub> test sets; corpus-level BLEU/METEOR of the Transformer-based MMT models on the 2016, 2017, and 2016<sub>N</sub> test sets. The underlined scores are higher than the score of the text-only Transformer model (“None”). The bold score is the best score of each language. “†” and “\*” indicate the statistical significance of the improvement or deterioration over the text-only Transformer model (“None”) and the BERT-fused model (“BERT”), respectively.**

Feature Type	2016		2017		2016 <sub>N</sub>		
	BLEU	METEOR	BLEU	METEOR	BLEU	METEOR	
English–German	None	38.83	56.96	30.93	50.51	25.61	45.45
	BERT	39.46	<b>58.02</b> †	<b>31.19</b>	<b>51.50</b> †	26.77†	46.71†
	ResNet-50	<u>39.30</u>	<u>57.62</u>	<u>30.98</u> †*	<u>51.07</u>	<u>26.99</u> †	<u>46.89</u> †
	Faster R-CNN	<u>39.49</u>	<u>57.73</u> †	<u>31.05</u>	<u>51.19</u>	<u>28.13</u> †	<u>48.05</u> †*
	LXMERT	<b>39.72</b>	<u>57.76</u> †	30.99	50.99	<u>27.58</u> †*	<u>48.02</u> †*
	All-inclusive	<u>39.18</u>	<u>57.73</u> †	30.80	<u>51.15</u> †	<b>28.39</b> †*	<b>48.35</b> †*
English–French	None	61.31	75.54	53.62	69.69	42.90	59.96
	BERT	<u>61.94</u> †	<u>76.09</u> †	<u>54.26</u>	<u>70.28</u>	<u>43.71</u> †	<u>60.74</u> †
	ResNet-50	<u>61.89</u>	<u>76.04</u> †	<u>54.21</u>	<u>70.24</u>	<u>44.79</u> †*	<u>61.80</u> †*
	Faster R-CNN	<u>61.63</u>	<u>75.92</u>	<u>53.62</u>	<u>69.79</u>	<u>46.55</u> †*	<u>63.26</u> †*
	LXMERT	<u>61.84</u>	<u>76.10</u> †	<u>53.84</u>	<u>69.96</u>	<u>46.82</u> †*	<u>63.60</u> †*
	All-inclusive	<b>62.40</b> †	<b>76.44</b> †	<b>54.41</b>	<b>70.47</b> †	<b>46.96</b> †*	<b>63.89</b> †*
English–Czech	None	31.25	29.99	-	-	21.42	23.74
	BERT	<b>32.41</b> †	<b>30.55</b> †	-	-	<u>22.25</u> †	<u>24.33</u> †
	ResNet-50	<u>31.71</u>	<u>30.18</u>	-	-	<u>22.86</u> †*	<u>24.69</u> †
	Faster R-CNN	<u>31.90</u>	<u>30.26</u>	-	-	<u>23.65</u> †*	<u>25.24</u> †*
	LXMERT	<u>31.98</u> *	<u>30.37</u>	-	-	<b>23.76</b> †*	<b>25.36</b> †*
	All-inclusive	<u>32.16</u> *	<u>30.38</u> †	-	-	<u>23.20</u> †*	<u>25.18</u> †*
English–Japanese	None	39.32	-	-	-	31.96	-
	BERT	<b>41.04</b> †	-	-	-	<u>32.78</u> †	-
	ResNet-50	<u>40.48</u> †	-	-	-	<u>33.00</u> †	-
	Faster R-CNN	<u>40.08</u> *	-	-	-	<u>33.21</u> †	-
	LXMERT	<u>40.28</u> †	-	-	-	<b>33.90</b> †*	-
	All-inclusive	<u>40.72</u> †	-	-	-	<u>33.58</u> †*	-

outperformed other models on the 2016 and 2017 test sets. This suggests that, when the input sentence is complete, the textual modality is more important than the visual modality.

However, in the 2016<sub>N</sub> test set, the benefit of using the BERT feature is less than those in the 2016 and 2017 test sets. This suggests that, while the textual context is limited, visual features profit more than the textual feature. We can further observe the significant improvement resulting from using most visual feature; the LXMERT feature profits more than most of the other visual features (ResNet-50 and Faster R-CNN). Moreover, the model achieves the best score along with the all-inclusive feature in many translation directions, which is a concatenation of BERT and LXMERT features. We may conclude that the LXMERT-fused MMT model is not only capable of utilizing visual features but is also good at working with both strong textual LM and visual-language LM.

Furthermore, these properties are consistent among translation directions, which is different from what we observed when using pre-trained word embedding. In all translation directions, the models fused with BERT, LXMERT, or the all-inclusive feature perform the best. We provide a detailed model comparison for English–German translation in Section VI-C.

## VI. DISCUSSION

In this section, we first examine the effectiveness of each debiasing method. Subsequently, we conduct an extensive quantitative analysis of the LXMERT-fused model.

### A. DEBIASING METHOD

Table 11 reports the average BLEU and METEOR scores of each model using different word embeddings and debiasing methods over different tokenization strategies. We can observe that All-but-the-Top (“AbtT”) achieves the best scores for nine out of 15 combinations of MMT models and word embedding. This is followed by localized centering (“LC”), which achieves the best scores for two. Conversely, autoencoder (“AE”) seems less capable with pre-trained embedding in the MMT scenario.

More interestingly, whereas the debiasing procedures only improve the benefit on six out of 10 benchmarks for word2vec, GloVe and FastText benefit on all benchmarks. This difference may be caused by how each embedding learns the global property of the training corpus. In contrast to word2vec, which learns to predict local context words from each word, GloVe learns based on the global co-occurrence matrix of the training data. FastText comprises each word embedding from its subword embeddings, which results in the generalized embedding rather than the

**TABLE 11.** Detailed comparison of pre-trained word embeddings and debiasing methods across conventional MMT models for English–German translation. The score in boldface is the best score among the vanilla and debiased embeddings for each embedding.

Emb.	Debias	Text-only		DECINIT		IMAG+		HA-NMT		VAG-NMT	
		BLEU	METEOR	BLEU	METEOR	BLEU	METEOR	BLEU	METEOR	BLEU	METEOR
Random	-	38.88	57.81	39.08	58.03	38.81	57.70	38.21	57.43	38.49	57.44
word2vec	Vanilla	<b>39.44</b>	<b>58.27</b>	39.25	58.12	39.14	<b>58.24</b>	<b>38.94</b>	<b>57.91</b>	<b>38.78</b>	57.63
	LC	39.13	58.00	38.98	57.95	39.18	58.05	38.70	57.72	38.27	57.48
	AbtT	39.24	58.04	<b>39.39</b>	<b>58.23</b>	39.28	58.19	38.88	57.91	38.50	<b>57.71</b>
	AE	39.19	57.98	39.33	58.20	<b>39.57</b>	58.24	38.48	57.64	38.55	57.52
GloVe	Vanilla	39.05	58.04	38.89	58.00	39.02	57.91	38.04	57.40	38.33	57.38
	LC	39.17	58.12	39.06	58.06	39.44	58.21	38.73	57.78	38.48	57.37
	AbtT	<b>39.49</b>	<b>58.30</b>	<b>39.30</b>	<b>58.17</b>	<b>39.57</b>	<b>58.24</b>	<b>38.76</b>	<b>57.82</b>	<b>38.55</b>	<b>57.71</b>
	AE	39.42	58.21	39.12	58.10	39.16	58.07	38.69	57.73	38.25	57.41
FastText	Vanilla	14.96	33.65	13.71	32.40	13.64	33.00	29.23	48.65	38.15	56.76
	LC	<b>39.25</b>	<b>58.13</b>	38.92	57.78	39.28	58.05	38.74	57.79	<b>38.67</b>	<b>57.64</b>
	AbtT	39.23	58.12	<b>39.29</b>	<b>58.22</b>	<b>39.53</b>	<b>58.27</b>	<b>38.87</b>	<b>57.88</b>	38.67	57.61
	AE	39.16	57.97	39.16	58.09	39.23	58.17	38.76	57.68	38.39	57.42

**TABLE 12.** BLEU (“B”) and METEOR (“M”) scores with various available features extracted from LXMERT for English–German translation. “(language)” and “(visual)” show the results of models that use only the language and vision part of each feature, respectively. The underlined scores are higher than the score of the text-only Transformer model (“None”). The score in boldface the best score for each test set.

Feature Type	2016		2017		2016 <sub>N</sub>	
	B	M	B	M	B	M
None	38.83	56.96	30.93	50.51	25.61	45.45
1. Objects	<u>39.49</u>	<u>57.73</u>	<u>31.05</u>	<u>51.19</u>	<u>28.13</u>	<u>48.05</u>
2. Embedding (language) (visual)	<u>39.48</u>	<u>57.71</u>	<u>31.04</u>	<u>51.11</u>	<u>28.14</u>	<u>48.02</u>
	38.78	57.52	30.95	51.48	26.48	46.47
	<u>39.20</u>	<u>57.40</u>	<u>31.08</u>	<u>51.00</u>	<u>28.15</u>	<u>47.90</u>
3. Single-M (language) (visual)	<u>39.12</u>	<u>57.48</u>	30.71	<u>50.91</u>	<b>28.63</b>	<b>48.73</b>
	<u>39.25</u>	<u>57.72</u>	<u>31.18</u>	<u>51.41</u>	<u>26.19</u>	<u>46.61</u>
	<u>39.56</u>	<u>57.75</u>	<u>31.12</u>	<u>51.21</u>	<u>27.65</u>	<u>47.71</u>
4. Cross-M (language) (visual)	<b>39.72</b>	57.76	30.99	<u>50.99</u>	<u>27.58</u>	<u>48.02</u>
	<u>39.02</u>	<b>57.78</b>	<b>31.37</b>	<b>51.70</b>	<u>27.73</u>	<u>47.63</u>
	<u>39.00</u>	57.68	<u>30.99</u>	<u>51.31</u>	<u>28.04</u>	<u>48.12</u>

localized embedding. Consequently, GloVe and FastText learn the global property of the training corpus more than word2vec does, which makes GloVe and FastText more capable with the debiasing method based on the global bias of the entire vocabulary. This result is consistent with the report by [21], which stated that word2vec with the autoencoder-based debiasing procedure is less capable than GloVe and FastText on word disambiguation tasks.

## B. FEATURE ABLATION

Selecting the appropriate feature is essential for leveraging the visual information for NMT. To reveal which part of the LXMERT feature contributes the most, we conducted experiments with various features extracted from LXMERT: (1) Object-level visual features as defined in (26) (Objects); (2) features before single-modality encoders (Embedding); (3) output of single-modality encoders (Single-M); and (4) output of cross-modality encoders (Cross-M).

Table 12 reports the results of ablation experiments conducted on the 2016, 2017, and 2016<sub>N</sub> test sets. Although the

model exploiting the cross-modal feature (“Cross-M”) is not the best model w.r.t. most of the test sets, it achieves almost the best performance. Interestingly, the best feature for any test sets is either a cross-modality feature or a concatenated single-modality feature. This suggests that the multimodal feature is more feasible for the model than single-modality features. Moreover, we need to select features from different layers to make the model best fit with different test sets. This would be caused by the pre-training tasks of LXMERT that are not optimized for NMT models and would present deceptive information in the LXMERT representations. The observation also suggests that selecting appropriate pre-training tasks will further boost translation quality.

## C. ALL-INCLUSIVE FEATURE

A key finding of this study is that the fuse-based model can utilize both LXMERT and BERT features in degraded scenarios. Table 13 shows the statistics of sentence sets that benefit from either LXMERT, BERT, or all-inclusive features.

Evidently, the largest contribution is made by 214 sentences (2016 test set) and 166 sentences (2016<sub>N</sub> test set) that are improved by all features. In the 2016<sub>N</sub> test set, the difference in the improvement made by the LXMERT feature and the all-inclusive feature is substantial (+0.83 BLEU). However, almost no additional improvement (+0.05 BLEU) is made by the all-inclusive features for Multi30K. Based on these results, we can conclude that the fuse-based model can utilize both LXMERT and BERT features when the input sentences are incomplete.

Moreover, by using the all-inclusive features, our model improved 47 samples (2016 test set) and 46 samples (2016<sub>N</sub> test set) that are not improved by using either the LXMERT or BERT features. These samples validate the assertion that the fused-based model with the LXMERT feature is capable of not only selectively using the better features from the BERT features or LXMERT features but also extracting novel information that is imperceptible in the underlying features.



**TABLE 13.** Statistics of the sentence subsets in the 2016 test set for English–German translation that benefit from the features with up arrow (green) and do not benefit from the features with down arrow (red). “L,” “B,” and “A” denote the models with the LXMERT feature, BERT feature, and all-inclusive feature, respectively. “Samples” shows the number of samples in each set. “Avg. ΔBLEU” shows the gain (or loss) of each feature from the text-only baseline.

Improved by			Samples	2016			Samples	2016 <sub>N</sub>		
L	B	A		Avg. ΔBLEU (L,B,A)				Avg. ΔBLEU (L,B,A)		
↑	↓	↓	48	+6.49	-4.04	-4.28	42	+8.13	-4.68	-3.40
↓	↑	↓	68	-4.78	+7.06	-4.54	52	-3.71	+8.29	-4.18
↓	↓	↑	47	-3.42	-3.20	+9.60	46	-4.51	-4.71	+8.11
↑	↑	↓	33	+10.17	+10.80	-2.37	34	+7.69	+9.38	-3.14
↑	↓	↑	36	+12.48	-4.48	+10.48	83	+15.59	-3.99	+14.71
↓	↑	↑	54	-2.09	+7.77	+8.10	38	-3.13	+10.30	+9.96
↑	↑	↑	214	+11.39	+10.02	+11.44	166	+12.59	+11.41	+13.42

**TABLE 14.** Each cell contains the ratio of wins and losses between the model in the row against the model in the column on the 2016 test set (top) and its image-demanding subset (bottom) for English–German translation.

	BERT	LXMERT
All-inclusive	<b>39.1</b> –36.8	<b>39.0</b> –36.9
LXMERT	<b>38.4</b> –36.8	-
All-inclusive	<b>41.3</b> –36.0	38.0– <b>40.0</b>
LXMERT	<b>44.7</b> –38.0	-

However, our model failed to improve 33 samples (2016 test set) and 34 samples (2016<sub>N</sub> test set) that were improved by using a single feature, but not by the all-inclusive feature. This demonstrates that the model failed to utilize two promising features for some samples. Further, the pairwise comparison (Table 14) of the models supports this idea, where the LXMERT feature contributes more than BERT and all-inclusive features in sentences that need images for translation.<sup>18</sup> Therefore, the model still has room for further improvements, especially in exploiting multiple features simultaneously without losing the individual benefits from composing features. We will explore this issue in future work.

**D. VISUAL AWARENESS**

To determine whether the LXMERT-fused model is aware of visual context, we performed adversarial evaluation [58] on the 2016 and 2016<sub>N</sub> test sets. In adversarial evaluation, we measure how a system performs when it is presented with the correct text data and either the correct image data (congruent) or incorrect image data (incongruent). To this end, we reversed the order of 1,000 images in each test set to obtain incongruent text–image data pairs. As we assumed that the input sentences are congruent, the incongruent LXMERT features were extracted from congruent sentences, giving incongruent images.

<sup>18</sup>Some translations in the 2016 test set were modified during the post-edit process with the presence of images, indicating that images are mandatory for these samples. We determined post-edited sentences by extracting sentences in WMT17 that differ from those in WMT16, obtaining 150 samples.

**TABLE 15.** BLEU scores on the 2016 test set in the incongruent setting for English–German translation. Subscripts are the difference to testing with congruent images.

Model	2016	2016 <sub>N</sub>
LXMERT	39.01 (-0.03)	24.48 (-3.38)
All-inclusive	39.35 (-0.00)	24.78 (-2.92)

Table 15 shows the corpus-level BLEU scores for each model in the adversarial evaluation. A large difference is observed between the congruent and incongruent settings in the 2016<sub>N</sub> test set, but almost no difference in the original Multi30K. This observation is consistent with the assertion made in [59], claiming that the source text in Multi30K is sufficient to perform the translation and prevents the visual features from affecting the model.

**E. HUMAN EVALUATION**


To investigate the characteristics of our models for human users, we asked human judges to rank the systems from best to worse for each source text. The 2016 test set, which consists of 1,000 input sentences, serves as evaluation data. For each input sentence, we sampled an output for each model from three translations generated by three trained systems. Ties were allowed, as multiple systems may generate the same translation for an input sentence. Finally, we turned absolute ranks into pairwise comparison of the two selected systems.

**TABLE 16.** Pairwise comparison by human judges on the 2016 test set for English–German translation. Each cell contains the ratio of wins and losses of the model in the row against the model in the column.

	Text-only	BERT	LXMERT
BERT	17.0–8.6	-	-
LXMERT	12.9–10.5	8.3–13.3	-
All-inclusive	13.9–12.0	7.0–13.2	8.6–9.9

Table 16 shows pairwise comparison of the four systems. While our proposed method still outperforms the text-only baseline, ties dominate human judges across all pairs. This result suggests that not only visual features but also the linguistic knowledge brings only a moderate improvement.

TABLE 17. Examples of English–German translation in the 2016 test set.

	Source	a young boy is standing next to a sand sculpture of a pyramid .
	Reference	ein kleiner junge steht neben einer pyramide aus sand .
	Text-only	ein kleiner junge steht neben einer sandskulptur .
	BERT	ein kleiner junge steht neben einer sandskulptur eines UNK .
	Faster R-CNN	ein kleiner junge steht neben einer sandskulptur .
	LXMERT	ein kleiner junge steht neben einer skulptur von einer pyramide .

Furthermore, we observed a small yet remarkable gap between LXMERT-based and BERT-fused models, which contradicts the fact that the LXMERT-fused model has a higher BLEU score than the BERT-fused model. We consider the perplexity of each model’s account for this gap; the perplexity<sup>19</sup> of BERT-fused model (8.59) is slightly lower than that of LXMERT-fused model (8.96). As BERT is pre-trained on larger data than those used for LXMERT, the BERT-fused model might generate more fluent translations than the LXMERT-fused model.

#### F. TRANSLATION EXAMPLES

Table 17 shows the English–German translation generated with different features. In this sample, the word “pyramid” is not translated by the text-only model and models with either BERT or RoI features. However, the LXMERT feature successfully guides the model to generate the German translation word “pyramide.” This sample demonstrates a good interaction between language and vision modalities. Specifically, the LXMERT feature guides the model to construct the sentence structure by leveraging the language modality, which is also observed when using the BERT feature, and then completes uncertain words by leveraging the vision modality.

#### VII. CONCLUSION

In this paper, we introduced two approaches to incorporate a monolingual corpus to improve MMT models. We showed that pre-trained word embeddings improve the translation performance along with the debiasing procedure and/or monolingual-corpus-based subword tokenization. Pre-trained VLMs are also proven to boost the translation quality. The results on multiple language pairs support the usefulness of monolingual data. Compared to the approaches based on parallel corpus, our proposed approach requires less-expensive annotations and is, therefore, more applicable for low-resource languages. Although we conducted experiments on various target languages to show the applicability across languages, the utility may deteriorate if our approaches are applied to a language with a culture that is distant from that of LXMERT. In future work, we would like to inspect the impact of this cultural gap for cultural-distance language pairs (e.g., English–Arabic).

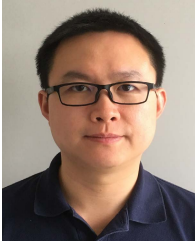
<sup>19</sup>We employed `bert-base-multilingual-uncased` to compute the perplexity.

After manipulating knowledge obtained from monolingual corpora, conventional MMT models still outperformed Transformer-based MMT models in some language pairs. However, through extensive analysis, we found the focus areas to develop better MMT models fused with pre-trained VLMs. In future work, we will examine training tasks for pre-trained VLMs that are more appropriate for multimodal NMT. Further, we will investigate models fused with multiple features that preserve every benefit made by their underlying features.

#### REFERENCES

- [1] D. Elliott, S. Frank, K. Sima’an, and L. Specia, “Multi30K: Multilingual English–German image descriptions,” in *Proc. VL*, 2016, pp. 70–74.
- [2] S. Frank, D. Elliott, and L. Specia, “Assessing multilingual multimodal image description: Studies of native speaker preferences and translator choices,” *Natural Lang. Eng.*, vol. 24, no. 3, pp. 393–413, May 2018.
- [3] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” in *Proc. ICLR*, 2015, pp. 1–15.
- [4] M. Popel, M. Tomkova, J. Tomek, Ł. Kaiser, J. Uszkoreit, O. Bojar, and Z. Žabokrtský, “Transforming machine translation: A deep learning system reaches news translation quality comparable to human professionals,” *Nature Commun.*, vol. 11, no. 1, pp. 1–15, Dec. 2020.
- [5] S. Kiyono, T. Ito, R. Konno, M. Morishita, and J. Suzuki, “Tohoku-AIP-NTT at WMT 2020 news translation task,” in *Proc. 5th Conf. Mach. Transl.* Stroudsburg, PA, USA: Association for Computational Linguistics, Nov. 2020, pp. 145–155. [Online]. Available: <https://aclanthology.org/2020.wmt-1.12>
- [6] L. Barrault et al., “Findings of the 2020 conference on machine translation (WMT20),” in *Proc. 5th Conf. Mach. Transl.* Stroudsburg, PA, USA: Association for Computational Linguistics, Nov. 2020, pp. 1–55. [Online]. Available: <https://aclanthology.org/2020.wmt-1.1>
- [7] P. Young, A. Lai, M. Hodosh, and J. Hockenmaier, “From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions,” *Trans. Assoc. Comput. Linguistics*, vol. 2, pp. 67–78, Feb. 2014. [Online]. Available: <https://aclanthology.org/Q14-1006>
- [8] S.-A. Grönroos, B. Huet, M. Kurimo, J. Laaksonen, B. Merialdo, P. Pham, M. Sjöberg, U. Sulubacak, J. Tiedemann, R. Troncy, and R. Vázquez, “The MeMAD submission to the WMT18 multimodal translation task,” in *Proc. 3rd Conf. Mach. Transl., Shared Task Papers*, 2018, pp. 603–611.
- [9] P. Lison and J. Tiedemann, “OpenSubtitles2016: Extracting large parallel corpora from movie and TV subtitles,” in *Proc. LREC*, 2016, pp. 923–929.
- [10] T. Lin, M. Maire, S. J. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft COCO: Common objects in context,” in *Proc. ECCV*, vol. 8693, 2014, pp. 740–755.
- [11] O. Caglayan, M. Kuyru, M. S. Amac, P. Madhyastha, E. Erdem, A. Erdem, and L. Specia, “Cross-lingual visual pre-training for multimodal machine translation,” in *Proc. 16th Conf. Eur. Chapter Assoc. Comput. Linguistics, Main Volume*. Stroudsburg, PA, USA: Association for Computational Linguistics, Apr. 2021, pp. 1317–1324. [Online]. Available: <https://aclanthology.org/2021.eacl-main.112>
- [12] G. Lample and A. Conneau, “Cross-lingual language model pretraining,” in *Proc. NeurIPS*, vol. 32, 2019, pp. 1–11.

- [13] Y. Qi, D. Sachan, M. Felix, S. Padmanabhan, and G. Neubig, "When and why are pre-trained word embeddings useful for neural machine translation?" in *Proc. NAACL*, 2018, pp. 529–535.
- [14] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Trans. Assoc. Comput. Linguistics*, vol. 5, pp. 135–146, Dec. 2017.
- [15] T. Hirasawa, H. Yamagishi, Y. Matsumura, and M. Komachi, "Multimodal machine translation with embedding prediction," in *Proc. NAACL SRW*, 2019, pp. 86–91.
- [16] G. Dinu, A. Lazaridou, and M. Baroni, "Improving zero-shot learning by mitigating the hubness problem," in *ICLR, Workshop Track*, 2015, pp. 1–10.
- [17] M. Faruqui, Y. Tsvetkov, P. Rastogi, and C. Dyer, "Problems with evaluation of word embeddings using word similarity tasks," in *Proc. RepEval*, 2016, pp. 30–35.
- [18] M. Radovanović, A. Nanopoulos, and M. Ivanović, "Hubs in space: Popular nearest neighbors in high-dimensional data," *J. Mach. Learn. Res.*, vol. 11, pp. 2487–2531, Sep. 2010.
- [19] K. Hara, I. Suzuki, M. Shimbo, K. Kobayashi, K. Fukumizu, and M. Radovanović, "Localized centering: Reducing hubness in large-sample data," in *Proc. AAAI*, 2015, pp. 2645–2651.
- [20] J. Mu and P. Viswanath, "All-but-the-top: Simple and effective postprocessing for word representations," in *Proc. ICLR*, 2018, pp. 1–25.
- [21] M. Kaneko and D. Bollegala, "Autoencoding improves pre-trained word embeddings," in *Proc. 28th Int. Conf. Linguistics*. Barcelona, Spain: International Committee on Computational Linguistics, Dec. 2020, pp. 1699–1713. [Online]. Available: <https://aclanthology.org/2020.coling-main.149>
- [22] T. Hirasawa and M. Komachi, "Debiasing word embeddings improves multimodal machine translation," in *Proc. Mach. Transl. Summit XVII, Res. Track*. Dublin, Ireland: European Association for Machine Translation, Aug. 2019, pp. 32–42. [Online]. Available: <https://aclanthology.org/W19-6604>
- [23] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. NAACL*, 2019, pp. 4171–4186.
- [24] Y. Liu and M. Lapata, "Text summarization with pretrained encoders," in *Proc. EMNLP-IJCNLP*, 2019, pp. 3730–3740.
- [25] M. Kaneko, M. Mita, S. Kiyono, J. Suzuki, and K. Inui, "Encoder-decoder models can benefit from pre-trained masked language models in grammatical error correction," in *Proc. ACL*, 2020, pp. 4248–4254.
- [26] J. Zhu, Y. Xia, L. Wu, D. He, T. Qin, W. Zhou, H. Li, and T. Liu, "Incorporating BERT into neural machine translation," in *Proc. ICLR*, 2020, pp. 1–18.
- [27] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. NeurIPS*, 2017, pp. 5998–6008.
- [28] H. Tan and M. Bansal, "LXMERT: Learning cross-modality encoder representations from transformers," in *Proc. EMNLP*, 2019, pp. 5100–5111.
- [29] K. Yawei and K. Fan, "Probing multi-modal machine translation with pre-trained language model," in *Proc. Findings Assoc. Comput. Linguistics (ACL-IJCNLP)*. Stroudsburg, PA, USA: Association for Computational Linguistics, Aug. 2021, pp. 3689–3699. [Online]. Available: <https://aclanthology.org/2021.findings-acl.323>
- [30] J. Helcl, J. Libovický, and D. Variš, "CUNI system for the WMT18 multimodal translation task," in *Proc. 3rd Conf. Mach. Transl., Shared Task Papers*, 2018, pp. 616–623.
- [31] R. Sennrich, B. Haddow, and A. Birch, "Improving neural machine translation models with monolingual data," in *Proc. ACL*, 2016, pp. 86–96.
- [32] S. Kumar and Y. Tsvetkov, "Von Mises–Fisher loss for training sequence to sequence models with continuous outputs," in *Proc. ICLR*, 2019, pp. 1–15.
- [33] G. Lample and A. Conneau, "Cross-lingual language model pretraining," in *Proc. NeurIPS*, 2019, pp. 7057–7067.
- [34] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," OpenAI, San Francisco, CA, USA, Tech. Rep., 2019.
- [35] K. Imamura and E. Sumita, "Recycling a pre-trained BERT encoder for neural machine translation," in *Proc. NGT*, 2019, pp. 23–31.
- [36] Y. Goyal, T. Khot, D. Summers-Stay, D. Batra, and D. Parikh, "Making the V in VQA matter: Elevating the role of image understanding in visual question answering," in *Proc. CVPR*, Jul. 2017, pp. 6904–6913.
- [37] J. Lu, D. Batra, D. Parikh, and S. Lee, "ViLBERT: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks," in *Proc. NeurIPS*, 2019, pp. 13–23.
- [38] L. Zhou, H. Palangi, L. Zhang, H. Hu, J. J. Corso, and J. Gao, "Unified vision-language pre-training for image captioning and VQA," in *Proc. AAAI*, 2019, pp. 13041–13049.
- [39] L. H. Li, M. Yatskar, D. Yin, C.-J. Hsieh, and K.-W. Chang, "VisualBERT: A simple and performant baseline for vision and language," 2019, *arXiv:1908.03557*.
- [40] W. Su, X. Zhu, Y. Cao, B. Li, L. Lu, F. Wei, and J. Dai, "VL-BERT: Pre-training of generic visual-linguistic representations," in *Proc. ICLR*, 2020, pp. 1–16.
- [41] Y. Chen, L. Li, L. Yu, A. E. Kholy, F. Ahmed, Z. Gan, Y. Cheng, and J. Liu, "UNITER: Universal image-text representation learning," in *Proc. ECCV*, vol. 12375, 2020, pp. 104–120.
- [42] O. Caglayan, A. Bardet, F. Bougares, L. Barrault, K. Wang, M. Masana, L. Herranz, and J. van de Weijer, "LIUM-CVC submissions for WMT18 multimodal translation task," in *Proc. 3rd Conf. Mach. Transl., Shared Task Papers*, 2018, pp. 597–602.
- [43] D. Elliott and A. Kádár, "Imagination improves multimodal translation," in *Proc. IJCNLP*, 2017, pp. 130–141.
- [44] J. Libovický and J. Helcl, "Attention strategies for multi-source sequence-to-sequence learning," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*. Vancouver, BC, Canada: Association for Computational Linguistics, Jul. 2017, pp. 196–202. [Online]. Available: <https://aclanthology.org/P17-2031>
- [45] M. Zhou, R. Cheng, Y. J. Lee, and Z. Yu, "A visual attention grounding neural model for multimodal machine translation," in *Proc. EMNLP*, 2018, pp. 3643–3653.
- [46] K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder–decoder approaches," in *Proc. SSST*, 2014, pp. 103–111.
- [47] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *Proc. ACL*, 2016, pp. 1715–1725.
- [48] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. NeurIPS*, 2015, pp. 91–99.
- [49] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. NeurIPS*, 2013, pp. 3111–3119.
- [50] J. Pennington, R. Socher, and C. Manning, "GloVe: Global vectors for word representation," in *Proc. EMNLP*, 2014, pp. 1532–1543.
- [51] O. Caglayan, M. García-Martínez, A. Bardet, W. Aransa, F. Bougares, and L. Barrault, "NMT-PY: A flexible toolkit for advanced neural machine translation systems," *Prague Bull. Math. Linguistics*, vol. 109, no. 1, pp. 15–28, Oct. 2017.
- [52] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, Jun. 2016, pp. 770–778.
- [53] D. Elliott, S. Frank, L. Barrault, F. Bougares, and L. Specia, "Findings of the second shared task on multimodal machine translation and multilingual image description," in *Proc. 2nd Conf. Mach. Transl.*, 2017, pp. 215–233.
- [54] K. Papineni, S. Roukos, T. Ward, and W. J. Zhu, "BLEU: A method for automatic evaluation of machine translation," in *Proc. 40th Annu. Meeting Assoc. Comput. Linguistics*. Philadelphia, PA, USA: Association for Computational Linguistics, Jul. 2002, pp. 311–318. [Online]. Available: <https://aclanthology.org/P02-1040>
- [55] M. Denkowski and A. Lavie, "Meteor universal: Language specific translation evaluation for any target language," in *Proc. 9th Workshop Stat. Mach. Transl.* Baltimore, MD, USA: Association for Computational Linguistics, Jun. 2014, pp. 376–380. [Online]. Available: <https://aclanthology.org/W14-3348>
- [56] G. A. Miller, *WordNet: An Electronic Lexical Database*. Cambridge, MA, USA: MIT Press, 1998.
- [57] O. Caglayan, "Multimodal machine translation," Ph.D. dissertation, Lab. d'Informatique de l'Université du Mans, Université du Maine, Orono, ME, USA, 2019.
- [58] D. Elliott, "Adversarial evaluation of multimodal machine translation," in *Proc. EMNLP*, 2018, pp. 2974–2978.
- [59] O. Caglayan, P. Madhyastha, L. Specia, and L. Barrault, "Probing the need for visual context in multimodal machine translation," in *Proc. NAACL*, 2019, pp. 4159–4170.

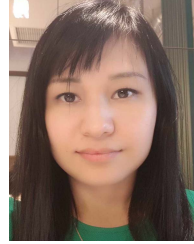


**TOSHO HIRASAWA** received the B.S. degree from Kyoto University, in 2009, and the master's degree in information science from Tokyo Metropolitan University, in 2021, where he is currently pursuing the Ph.D. degree with the Graduate School of System Design. His research interests include machine translation and multimodal natural language processing.



**MASAHIRO KANEKO** received the B.E. degree from the Kitami Institute of Technology, in 2016, the M.E. degree from Tokyo Metropolitan University, in 2018, and the Ph.D. degree in information science from the Graduate School of System Design, Tokyo Metropolitan University, in 2021. He was a Research Fellow (DC2) of the Japan Society for the Promotion of Science, from 2019 to 2021. He is currently a Postdoctoral Researcher with the Department of Computer

Science, Tokyo Institute of Technology. He is also a Visiting Researcher in Tokyo Metropolitan University. His research interests include machine learning and natural language processing.



**AIZHAN IMANKULOVA** received the B.E. degree from Kazakh National Technical University, in 2011, the M.E. degree from the International University of Information Technologies, in 2014, and the Ph.D. degree in information science from the Graduate School of System Design, Tokyo Metropolitan University, in 2021. She received the Japanese Government (MEXT) Scholarship, from 2016 to 2020.

She is currently a Data Scientist and an Engineer with CogSmart. She is also a Visiting Researcher in Tokyo Metropolitan University.



**MAMORU KOMACHI** received the M.Eng. and Ph.D. degrees from the Nara Institute of Science and Technology (NAIST), in 2007 and 2010, respectively. He was an Assistant Professor at NAIST. He is currently a Professor at Tokyo Metropolitan University. His research interests include semantics, information extraction, and educational applications of natural language processing.

...