Visually Grounded Word Embeddings and Richer Visual Features for Improving Multimodal Neural Machine Translation

Jean-Benoit Delbrouck[1], *Stéphane Dupont*[1], *Omar Seddati*[1]

[1]TCTS Lab, University of Mons, Belgium

{Jean-Benoit.DELBROUCK, Stephane.DUPONT, Omar.SEDDATI}@umons.ac.be

Abstract

In Multimodal Neural Machine Translation (MNMT), a neural model generates a translated sentence that describes an image, given the image itself and one source descriptions in English. This is considered as the multimodal image caption translation task. The images are processed with Convolutional Neural Network (CNN) to extract visual features exploitable by the translation model. So far, the CNNs used are pre-trained on object detection and localization task. We hypothesize that richer architecture, such as dense captioning models, may be more suitable for MNMT and could lead to improved translations. We extend this intuition to the word-embeddings, where we compute both linguistic and visual representation for our corpus vocabulary. We combine and compare different configurations and show state-of-the-art results according to previous work.

Index Terms: neural machine translation, multimodal, attention mechanism, image features, visual embeddings, grounding

1. Introduction

In machine translation, neural networks have attracted a lot of research attention. Recently, the encoder-decoder framework [1] has been largely adopted. In this approach, Recurrent Neural Networks (RNNs) map source sequences of words to target sequences. An attention mechanism has been introduced by Bahdanau et al. [2] to learn to focus on different parts of the input sentence while decoding. Attention mechanisms have shown to work with other modalities too, like images, where their are able to learn to attend the salient parts of an image, for instance when generating text captions [3].

The efficient integration of multimodal information still remains a challenging task. It requires combining diverse modality vector representations. A few attempts [4, 5, 6, 7, 8] have been made during the WMT 2016 Multimodal Machine Translation evaluation campaign. These initial efforts have not convincingly demonstrated that visual context can improve translation quality. Meanwhile, few improvements have been made, [9] proposed a doubly-attentive decoder that outperformed all previous baselines with less data and without re-scoring, [10] tried multiple attention models and image attention optimizations such as the gating [3] and pre-attention [11] mechanism. Recently, [12] introduced a model where visually grounded representations are learned.

In this paper, our aim is to propose a first empirical investigation on improving MNMT by using improved visual and word representations. More specifically, we believe that visual and word representations obtained through models pre-trained on large data sets should bring further improvement. Most importantly, we want to leverage models that provide a closer link between image understanding and language understanding.

For extracting image modality vector representations, we will make use of a model trained on a dense captioning task, namely DenseCap [13]. Compared to models trained on object recognition tasks (such as ImageNet [14], as used in previous MNMT proposal), the hope is that the representation contains richer information, also encoding object attributes and important relationship for linguistic description of the images. For extracting the vector representations of the word modality, we will make use of word vectors obtained from large scale text corpora, but we will also use visual representations of the referents of those words, using a recent paradigm of "imagined" visual representations of those words. The hope is that these visually grounded word representation will facilitate the integration of both modalities during the decoding process, hence improving translation results.

Our paper is structured as follows. In section 2, we briefly describe our NMT model as well as the conditional GRU activation used in the decoder. We also explain how multi-modalities can be implemented within this framework. In the following section 3 and 4, we detail the process of our visual embeddings and features creation. Finally, we report and analyze our results in section 6.

2. Model

2.1. Text-based NMT

We describe the attention-based NMT model introduced by [2] in this section. Given a source sentence $X = (x_1, x_2, \ldots, x_M)$, the neural network directly models the conditional probability $p(Y|X)$ of its translation $Y = (y_1, y_2, \ldots, y_N)$. The network consists of one encoder and one decoder with one attention mechanism. Each source word x_i and target word y_i are a column index of the embedding matrices $\boldsymbol{E}_X$ and $\boldsymbol{E}_Y$. The encoder is a bi-directional RNN with Gated Recurrent Unit (GRU) layers [15, 16], where a forward RNN $\overrightarrow{\Psi}_{\text{enc}}$ reads the input sequence as it is ordered (from x_1 to x_M) and calculates a sequence of forward hidden states $(\overrightarrow{\boldsymbol{h}}_1, \overrightarrow{\boldsymbol{h}}_2, \ldots, \overrightarrow{\boldsymbol{h}}_M)$. A backward RNN $\overleftarrow{\Psi}_{\text{enc}}$ reads the sequence in the reverse order (from x_M to x_1), resulting in a sequence of backward hidden states $(\overleftarrow{\boldsymbol{h}}_M, \overleftarrow{\boldsymbol{h}}_{M-1}, \ldots, \overleftarrow{\boldsymbol{h}}_1)$. We obtain an annotation for each word x_i by concatenating the forward and backward hidden state $\boldsymbol{h}_t = [\overrightarrow{\boldsymbol{h}}_t; \overleftarrow{\boldsymbol{h}}_t]$. Each annotation $\boldsymbol{h}_t$ contains the summaries of both the preceding words and the following words. The representation C for each source sentence is the set of annotations $C = (\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_M)$.

The decoder is an RNN that uses a conditional GRU[1] (cGRU) with an attention mechanism to generate a translated word y_t.

---

[1]https://github.com/nyu-dl/dl4mt-tutorial/blob/master/docs/cgru.pdf

More precisely, the conditional GRU has three main components computed at each time step t of the decoder:

- REC1 computes a hidden state proposal s'_t based on the previous hidden state s_{t-1} and the previously emitted word y_{t-1};

- f_{att} [2] is an attention mechanism over the hidden states of the encoder and computes a time-dependent context vector c_t using the annotation set C and the hidden state proposal s'_t;

- REC2 computes the final hidden state s_t using the hidden state proposal s_t and the context vector c_t;

Both s_t and c_t are further used to decode the next symbol. We use a deep output layer [17] to compute a vocabulary-sized vector :

$$o_t = L_o \tanh(L_s s_t + L_c c_t + L_w E_Y[y_{t-1}]) \qquad (1)$$

where L_o, L_s, L_c, L_w are model parameters. We can parameterize the probability of decoding each word y_t as:

$$p(y_t|y_{t-1}, s_t, c_t) = \text{Softmax}(o_t) \qquad (2)$$

The initial state of the decoder s_0 at time-step $t = 0$ is initialized by the following equation :

$$s_0 = f_{init}(h_M) \qquad (3)$$

where f_{init} is a feed-forward network with one hidden layer.

We use the soft attention mechanism for the f_{att} component. Soft attention has firstly been used for syntactic constituency parsing by [18] but has been widely used for translation tasks ever since. The idea of the soft attentional model is to consider all the annotations when deriving the context vector c_t. It consists of a single feed-forward network used to compute an expected alignment e_t between text annotation h_i and the target word to be emitted at the current time step t. The inputs are the annotations and the intermediate representation of REC1 s'_t:

$$e_{t,l} = v^T \tanh(U_a s'_t + W_a a_l) \qquad (4)$$

$$\alpha_{t,i} = \frac{\exp(e_{t,i})}{\sum_{j=1}^{L} \exp(e_{t,j})} \qquad (5)$$

where $\alpha_{t,i}$ is the normalized alignment matrix between each source annotation vector h_i and the word y_t to be emitted at time step t. In the above expressions, v^T, U_a and W_a are trained parameters. Finally, the modality time-dependent context vector c_t is computed as a weighted sum over the annotation vectors (equation 6).

$$c_t = \sum_{i=1}^{L} \alpha_{t,i} a_i \qquad (6)$$

2.2. Multimodal NMT (MNMT)

In multimodal NMT, the second modality is usually an image, for which feature maps are computed using a Convolutional Neural Network (CNN). The annotations $a_1, a_2, \ldots, a_L$ are spatial features (i.e. each annotation represents features for a

---

[2] called ATT in the aforementioned paper

specific region in the image). More formally, given a set of image modality annotations $I = (a_1, a_2, \ldots, a_L)$, we compute a an image context vector i_t based on the same intermediate hidden state proposal:

$$i_t = f'_{att}(I, s'_t) \qquad (7)$$

This new time-dependent context vector is an additional input to a modified version of REC2 which now computes the final hidden state s_t using the intermediate hidden state proposal s'_t and both time-dependent context vectors c_t (for text) and i_t (for image). In addition, i_t is weighted with the gating scalar mechanism as seen in [3]

$$z_t = \sigma\left(W_z c_t + W_z i_t + U_z s'_t\right)$$
$$r_t = \sigma\left(W_r c_t + W_r i_t + U_r s'_t\right)$$
$$\underline{s}_t = \tanh\left(W c_t + W i_t + r_t \odot (U s'_t)\right)$$
$$s_t = (1 - z_t) \odot \underline{s}_t + z_t \odot s'_t \qquad (8)$$

The probabilities for the next target word (from equation 1) also takes into account the new context vector i_t:

$$L_o \tanh(L_s s_t + L_c c_t + L_i i_t + L_w E_Y[y_{t-1}]) \qquad (9)$$

where L_i is a new trainable parameter.

3. Improved Word Embeddings

In previous works on MNMT, word embeddings are usually trained along with the model. Both matrices E_X and E_Y are considered trained model parameters. This approach does not allow to exploit large scale text corpora that can be available in the source language and that could be leveraged to obtain useful distributed semantic representations of the words (such as Word2Vec [19], or Glove [20]). Here, we will make use of Glove to build a multimodal representation, textual and visual, for our whole source vocabulary. To do so, we try out an effective method that learns a language-to-vision mapping as described in [21]. The learned model outputs visual predictions of a word given its semantic representation.

3.1. Language-to-vision mapping

Concretely, we consider two embedding spaces: a linguistic space $\mathcal{L} \subset \mathbb{R}^{d_l}$ and a visual space $\mathcal{V} \subset \mathbb{R}^{d_v}$ where d_l and d_v are the sizes of the text and visual representations respectively. For a given dataset of words $\mathcal{W} = \{w_1, w_2, \ldots, w_N\}$, each word w_i has a linguistic representation $l_{w_i} \in \mathcal{L}$ and a visual representation $v_{w_i} \in \mathcal{V}$. The aim is to learn a mapping function $f : \mathcal{L} \to \mathcal{V}$ such that the prediction (or imagined representations) $f(l_{w_i})$ is close to the actual visual vector v_{w_i}. A training example is thus a pair $\{l_{w_i}; v_{w_i}\}$ and the dataset is composed of N examples.

3.2. Visually Grounded Word Embeddings

ImageNet is used as source of visual information. ImageNet covers a total of 21,841 WordNet synsets and has 14,197,122 images. For the experiment, only synsets with more than 50 images are kept, and an upper bound of 500 images per synset is used to reduce computation time. With this restriction, 9,251 unique words are covered. The training set is composed of $N = 9,251$ examples. For each of these words, we use a pre-trained

VGG-m-128 CNN model [22] to extract visual features from each image. We take the 128-dimensional activation of the last layer. The visual representation v_{w_i} is computed as an element-wise averaging of the features vectors from different images picturing the object the word refers to. As previously mentioned, the textual representation l_{w_i} is obtained with the word embeddings algorithm Glove. We use the pre-trained model on the Common Crawl corpus consisting of 840B tokens and a 2.2M words. The mapping function f consists of a simple perceptron composed of a d_l dimensional input layer and a linear output layer with d_v units.

3.3. Imagined multimodal embeddings

We use the pre-trained model made available by [21]. Their training is done with a mean squared error (MSE) loss function and stochastic gradient descent optimizer. A learning rate of 0.1 and dropout rate of 0.1 is chosen, running for 175 epochs. GloVe vectors are of size d_l = 300 and a low-dimensional d_v = 128 is picked to reduce the number of parameters and thus the risk of overfitting. The multimodal representation of word is built by concatenating the $\ell 2$-normalized imagined representations $f(l_{w_i})$ with the textual representations l_{w_i}. Hence, the multimodal representation is of size 428. We compute it using our model for every word in our source vocabulary.

4. Visual Features

So far in MNMT, the spatial visual features $a_1, a_2, \ldots, a_L$ of the images are extracted with a 16 or 19-layers version of VG-GNet, or a Deep Residual Network [23] pre-trained on Ima-geNet for an object detection and localization task. As mentioned in [10], such features may not be suited for the translation of complex captions, which involves objects but also their attributes and relationships (as shown in Figure 1). In this work, we focus on using features extracted on a model pre-rained for a dense captioning task, namely DenseCap pre-trained on Visual Genome [24]. Its architecture is composed of a Convolutional Network, a novel dense localization layer, and Recurrent Neural Network language model that generates the label sequences. The CNN is pretrained on ImageNet and fine-tuned during the training (except for the first four convolutional layers). We extract the features at the last convolutional layer. Due to the high sparsity of these features, we apply an ℓ_2-normalization.

In this experiment, we compare two image annotations used by our decoder: features of size $14 \times 14 \times 1024$ extracted from a ResNet-50 pre-trained on ImageNet (at its res4f layer) and $14 \times 14 \times 512$ from DenseCap as described above.
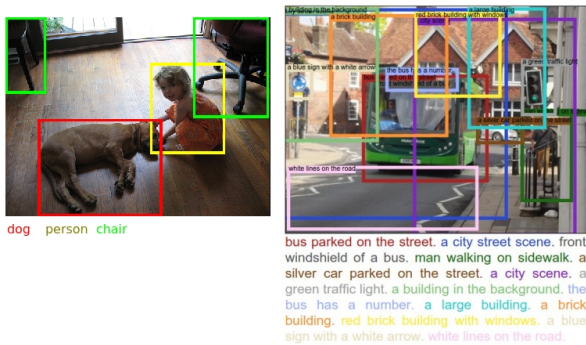

Figure 1: *Left: Object localization and detection. Right: Dense captioning*

5. Dataset and model settings

For this experiments on Multimodal Machine Translation, we used the Multi30K dataset [25] which is an extended version of the Flickr30K Entities. For each image, one of the English descriptions was selected and manually translated into German by a professional translator. As training and development data, 29,000 and 1,014 triples are used respectively. A test set of size 1000 is used for metrics evaluation.

All our models are build on top of the nematus framework [26]. The encoder is a bidirectional RNN with GRU, one 1024D single-layer forward and one 1024D single-layer backward RNN. Non-recurrent matrices are initialized by sampling from a Gaussian $\mathcal{N}(0, 0.01^2)$, recurrent matrices are random orthogonal and bias vectors are all initialized to zero. The word embeddings matrices E_X and E_Y are either trained along the model and initialized accordingly or pre-trained as described in section 3. Embeddings size depends on the experiment and are explicitly mentioned in the score tabular (Table 1) for every model. We apply dropout with a probability of 0.3 on the embeddings, on the hidden states in the bidirectional RNN in the encoder as well as in the decoder. In the decoder, we also apply dropout on the text annotations h_i, the image features a_i, on both modality context vector and on all components of the deep output layer before the readout operation. Dropout is applied using one same mask in all time steps [27].

We normalize and tokenize English and German descriptions using the Moses tokenizer scripts [28]. We use the byte pair encoding algorithm on the target train set to convert space-separated tokens into subwords [29], reducing the German vocabulary to 14957 words. Our source vocabulary, in English, is of size 11187. The visual features of the Flickr30K images are extracted with DenseCap or ResNet-50 in our experiments (as described in section 4).

All variants of our model were trained with ADADELTA [30], with mini-batches of 40 examples. We apply early stopping for model selection based on BLEU4 : training is halted if no improvement on the development set is observed for more than 20 epochs. We use the metrics BLEU4 [31], METEOR [32] and TER [33] to evaluate the quality of our models' translations.

6. Results and Future work

We report our results in table 1. We structure our analysis in three main sections. We start by discussing the effectiveness of the DenseCap features, followed by our impressions about the impact of the multimodal word embeddings. Finally, we qualitatively compare the translations of some of our models to get more concrete and tangible results of our choices.

Flickr30K Visual Features We observe a noticeable improvement when using DenseCap visual features instead of ResNet-50. We show an amelioration, between model B1 and B2, of +1.51 BLEU and +0.7 METEOR when embeddings are trained along with the model. With pre-trained GloVe embeddings, between systems L1 and L2, we notice an improvement of +0.55 BLEU and 0.7 METEOR. Lastly, with multimodal embeddings, model M3 scores +0.69 BLEU and +0.5 METEOR over M2. Overall, the results confirm our intuition that rich dense captioning task improves translation's quality, especially on the METEOR metric which proves that

Table 1: *Results on the test triples of the Multi30K dataset. "Along" means embeddings are initialized with a Gaussian and trained along with the model. "Fixed" means that the embeddings are frozen during the whole training. Embeddings size are between brackets at the end of each model description. Each score is compared with the model B1.*

Model	Test Scores					
	BLEU↑		METEOR↑		TER↓	
Previous work						
[9] ResNet-50 + along (620)	36.50		55.0		43.7	
[12] GoogLeNet v3 + along (620)	36.8 ± 0.8		55.8 ± 0.4		-	
[12] GoogLeNet v3 + along (620) + COCO + NC	37.8 ± 0.7		57.1 ± 0.2		-	
Baseline						
(B1) ResNet-50 + along (428)	36.27		53.9		43.6	
(B2) DenseCap + along (428)	37.78	↑ +1.51	54.6	↑ +0.7	42.3	↓ -0.3
Linguistic word embeddings						
(L1) ResNet-50 + GloVe (300)	37.40	↑ +1.13	55.0	↑ +1.1	42.1	↓ -0.5
(L2) DenseCap + GloVe (300)	37.95	↑ +1.68	55.7	↑ +1.8	42.1	↓ -0.5
Imagined multimodal word embeddings						
(M1) ResNet-50 + GloVe + Visual (428, fixed)	35.52	↓ -0.75	53.7	↓ -0.2	43.3	↑ +0.7
(M2) ResNet-50 + GloVe + Visual (428)	37.51	↑ +1.24	55.2	↑ +1.3	42.1	↓ -0.5
(M3) DenseCap + GloVe + Visual (428)	38.20	↑ +1.93	55.7	↑ +1.8	41.9	↓ -0.7

the attention model benefits from this change. The use of pre-trained embeddings lowers the gap betweens image features efficiency (ie. the scores difference between M2-M3 and L1-L2 are slightly lower than M1-M2). Moreover, it is interesting to note that the Denscap features (of size $14 \times 14 \times 512$) are twice smaller than the ResNet-50 features ($14 \times 14 \times 1024$).

Word embeddings Obviously, adding GloVe word embeddings helps the model to better translate. Yet, the addition of the visual embeddings to the linguistic embeddings only brings a small improvement: +0.11 BLEU and +0.2 METEOR for ResNet-50 (L1-M2) and +0.35 BLEU and +0.0 METEOR for DenseCap (L2-M3). We can hypothesize two reasons. Firstly, the visual embeddings of size 128 may be too small to be really significant in rather large networks such as these presented and the model in section 3 might need some changes. Another explanation could be that the visual information used during the training of the mapping function (the function f that outputs our visual embeddings of size 128) are extracted with a VGG-128 network. However, our models are trained using a ResNet-50 or DenseCap network for visual features extraction. It is possible that too much model parameters are requested to do the mapping between the two embedding spaces (the word embeddings and the visual features) and therefore impact the models' translation quality. On a side note, we tried to freeze the loaded embeddings during the training (referred as "fixed" in the score tabular) but lead to unpleasant results of -0.75 BLEU and -0.2 METEOR (B1-M1). We conclude that the models needs to slightly change the representations of the words in order to generate the strongest textual and visual context vectors.

Translation comparison We illustrate some hand-picked examples of significant improvements on the test set we noticed between our models. We start by comparing the two baseline systems B1 and B2. We pick a sentence that involves a positional relationship between a man and a dog (Figure 2):

Source:	A man is dancing with a dog between his legs .
Reference:	Ein Mann tanzt mit einem Hund zwischen den Beinen .
B1:	Ein Mann tanzt und ein Hund zwischen seinen Beinen .
B2:	Ein Mann tanzt mit einem Hund zwischen den Beinen .

The sentence-level BLEU score is of 100 for B2 (DenseCap) and 31.40 for B1 (ResNet-50). We now compare our two best models M2 and M3 on a more complex and descriptive sentence:

Source:	A person in a red jacket with black pants holding rainbow ribbons .
Reference:	Eine Person in einer roten Jacke und schwarzen Hosen hält Regenbogenbänder .
M2:	Eine Person in roter Jacke mit schwarzen Hosen hält eine Znde in der Hand .
M3:	Eine Person in einer roten Jacke und schwarzen Hosen hält einen Zebnde .

M3 and M2 respectively have a sentence-level BLEU score of 77.19 and 23.66.
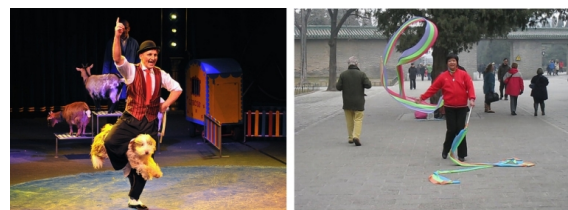

Figure 2: *Left: First sentence Right: Second sentence*

In future work, it would be interesting to use the same CNN for all the model's components (multimodal word embeddings and visual features). The experiment could be re-attempt with DenseCap on all fronts. Another work could be a more in-depth investigation of the attention model behavior using DenseCap. Its not clear if the improved behavior learned by the model is brought by the word-embeddings (and therefore GloVe and the VGG-128 of section 3) or the attention model on the visual features (extracted with DenseCap).

7. Acknowledgements

8. References

[1] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014, pp. 3104–3112.

[2] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *CoRR*, vol. abs/1409.0473, 2014.

[3] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, D. Blei and F. Bach, Eds. JMLR Workshop and Conference Proceedings, 2015, pp. 2048–2057. [Online]. Available: http://jmlr.org/proceedings/papers/v37/xuc15.pdf

[4] I. Calixto, D. Elliott, and S. Frank, "Dcu-uva multimodal mt system report," in *WMT*, 2016.

[5] J. Libovický, J. Helcl, M. Tlustý, O. Bojar, and P. Pecina, "Cuni system for wmt16 automatic post-editing and multimodal translation tasks," in *Proceedings of the First Conference on Machine Translation*. Berlin, Germany: Association for Computational Linguistics, August 2016, pp. 646–654. [Online]. Available: http://www.aclweb.org/anthology/W16-2361

[6] O. Caglayan, W. Aransa, Y. Wang, M. Masana, M. García-Martínez, F. Bougares, L. Barrault, and J. van de Weijer, "Does multimodality help human and machine for translation and image captioning?" in *Proceedings of the First Conference on Machine Translation*. Berlin, Germany: Association for Computational Linguistics, August 2016, pp. 627–633.

[7] P.-Y. Huang, F. Liu, S.-R. Shiang, J. Oh, and C. Dyer, "Attention-based multimodal neural machine translation," in *WMT*, 2016.

[8] K. Shah, J. Wang, and L. Specia, "Shef-multimodal: Grounding machine translation on image," in *In Proceedings of the First Conference on Machine Translation*, 2016, p. 660665.

[9] I. Calixto, Q. Liu, and N. Campbell, "Doubly-attentive decoder for multi-modal neural machine translation," *CoRR*, vol. abs/1702.01287, 2017. [Online]. Available: http://arxiv.org/abs/1702.01287

[10] J.-B. Delbrouck and S. Dupont, "An empirical study on the effectiveness of images in multi-modal neural machine translation - under review," 2017.

[11] J. Delbrouck and S. Dupont, "Multimodal compact bilinear pooling for multimodal neural machine translation," *CoRR*, vol. abs/1703.08084, 2017. [Online]. Available: http://arxiv.org/abs/1703.08084

[12] D. Elliott and Á. Kádár, "Imagination improves multimodal translation," *CoRR*, vol. abs/1705.04350, 2017.

[13] J. Johnson, A. Karpathy, and L. Fei-Fei, "Densecap: Fully convolutional localization networks for dense captioning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[14] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "Imagenet large scale visual recognition challenge," *Int. J. Comput. Vision*, vol. 115, no. 3, pp. 211–252, Dec. 2015. [Online]. Available: http://dx.doi.org/10.1007/s11263-015-0816-y

[15] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, *Empirical evaluation of gated recurrent neural networks on sequence modeling*, 2014.

[16] K. Cho, B. van Merriënboer, Ç. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder–decoder for statistical machine translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1724–1734. [Online]. Available: http://www.aclweb.org/anthology/D14-1179

[17] R. Pascanu, C. Gulcehre, K. Cho, and Y. Bengio, *How to construct deep recurrent neural networks*, 2014.

[18] O. Vinyals, L. u. Kaiser, T. Koo, S. Petrov, I. Sutskever, and G. Hinton, "Grammar as a foreign language," in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Curran Associates, Inc., 2015, pp. 2773–2781. [Online]. Available: http://papers.nips.cc/paper/5635-grammar-as-a-foreign-language.pdf

[19] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing Systems 26*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2013, pp. 3111–3119.

[20] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543. [Online]. Available: http://www.aclweb.org/anthology/D14-1162

[21] G. Collell, T. Zhang, and M.-F. Moens, "Imagined visual representations as multimodal embeddings," in *AAAI*, 2017, pp. 4378–4384.

[22] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[24] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, M. Bernstein, and L. Fei-Fei, "Visual genome: Connecting language and vision using crowdsourced dense image annotations," 2016. [Online]. Available: https://arxiv.org/abs/1602.07332

[25] D. Elliott, S. Frank, K. Simaan, and L. Specia, "Multi30k: Multilingual english-german image descriptions," 2016.

[26] R. Sennrich, O. Firat, K. Cho, A. Birch, B. Haddow, J. Hitschler, M. Junczys-Dowmunt, S. L"aubli, A. V. Miceli Barone, J. Mokry, and M. Nadejde, "Nematus: a Toolkit for Neural Machine Translation," in *Proceedings of the Demonstrations at the 15th Conference of the European Chapter of the Association for Computational Linguistics*, 2017.

[27] Y. Gal and Z. Ghahramani, "A theoretically grounded application of dropout in recurrent neural networks," in *Advances in Neural Information Processing Systems 29 (NIPS)*, 2016.

[28] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens,

C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: Open source toolkit for statistical machine translation," in *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ser. ACL '07. Stroudsburg, PA, USA: Association for Computational Linguistics, 2007, pp. 177–180. [Online]. Available: http://dl.acm.org/citation.cfm?id=1557769.1557821

[29] R. Sennrich, B. Haddow, and A. Birch, "Neural Machine Translation of Rare Words with Subword Units," in *In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2016. [Online]. Available: http://www.aclweb.org/anthology/P16-1162

[30] M. D. Zeiler, "ADADELTA: an adaptive learning rate method," *CoRR*, vol. abs/1212.5701, 2012. [Online]. Available: http://arxiv.org/abs/1212.5701

[31] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: A method for automatic evaluation of machine translation," in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ser. ACL '02. Stroudsburg, PA, USA: Association for Computational Linguistics, 2002, pp. 311–318. [Online]. Available: http://dx.doi.org/10.3115/1073083.1073135

[32] M. Denkowski and A. Lavie, "Meteor universal: Language specific translation evaluation for any target language," in *Proceedings of the EACL 2014 Workshop on Statistical Machine Translation*, 2014.

[33] M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul, "A study of translation edit rate with targeted human annotation," in *In Proceedings of Association for Machine Translation in the Americas*, 2006, pp. 223–231.