Image Captioning with a Constraint of Image-to-Text Transformation

Jing Wen[1], Han Liu[2], Xiaolin Hu[2], Jianmin Li [2], Zhaoxiang Zhang[3]
[1]Carnegie Mellon University
[2]Tsinghua University
[3]Chinese Academy of Sciences
jingwen2@andrew.cmu.edu, han-liu18@mails.tsinghua.edu.cn, xlhu@tsinghua.edu.cn
lijianmin@mail.tsinghua.edu.cn, zhaoxiang.zhang@ia.ac.cn

Abstract

The encoder-decoder framework is widely adopted in image captioning where the encoder generates image features and the decoder receives the image features and generates captions. However, this framework has insufficient capability to reduce the gap between image and text representations, thus leading to poor generation results. One solution is to embed the two modalities in the same space such that the representation of an image region (e.g., a person) is close to the representation of a corresponding word (e.g., "person") in that space. To achieve this goal, we propose to add a constraint to the encoder-decoder framework such that the image features can be transformed to the text embedding space and represent the captions. By minimizing an auxiliary loss function which encourages the transformed image representation to be close to the caption representation, we explicitly bridge the gap between two modalities. The decoder learns this image-to-text transformation and generates better captions for given images. Experiments on the MSCOCO captioning dataset demonstrate the effectiveness of the proposed method.

1. Introduction

Image captioning is one of the core tasks in artificial intelligence, which requires the machine to understand what it sees and describe the contents with natural language. In recent year, it has received massive attention from the communities of both computer vision and natural language processing. Benefiting from the development of large-scale datasets [5, 16, 21] and the popularity of deep learning [27, 26], automatic caption generation has been largely improved from the perspectives of syntax and semantics.

The encoder-decoder framework was originally proposed for machine translation [4] and showed great potential in image captioning [26] later. When adapted to image captioning, the encoder takes images as inputs and generates features which are forwarded to the decoder to generate the captions (Fig. 1a). The image features usually contain the representations of objects, attributes, relationships, etc. To fulfill this task, this framework should have two capabilities. First, it should be able to bridge the gap between image and text representations since the decoder takes image features as input and generates sentences. In order to obtain good results, the image representation and text representation must have close relationship. For example, ideally, an image region containing a person should be closely related to the words "person" and "human" based on their representations. To obtain image features, one successful practice is to use the Faster R-CNN [22] pretrained on the Visual Genome dataset [16] as the image encoder, and *fix* it in subsequent training [2]. In this case, the decoder alone is responsible for learning the transformation from image representation to text representation. Second, given the transformed image features, the framework, actually the decoder, should be able to organize the knowledge into semantically and syntactically correct sentences. An interesting question arises: if we want to further enhance of the performance of this framework, which capability should we emphasize on?

We empirically find that, when there is no modality difference between the encoder and decoder, an autoencoder architecture (Fig. 1b) has excellent ability to reconstruct the input sentence. Taking the captions as inputs, the autoencoder architecture learns to generate the same sentences as inputs. As shown in Table 1, the Transformer-based text encoder combined with different decoders can obtain the CIDEr-D values higher than 200. In contrast, a recently published image captioning model obtains only 128.0 CIDEr-D [18] (last row of the table; see Section 5.3 for results of more image captioning models). CIDEr-D [25] is a metric to measure the quality of the generated sentence. The higher, the better. These results imply that the decoder has strong capability to organize semantically and syntactically correct sentences. Then the gap of perfor-

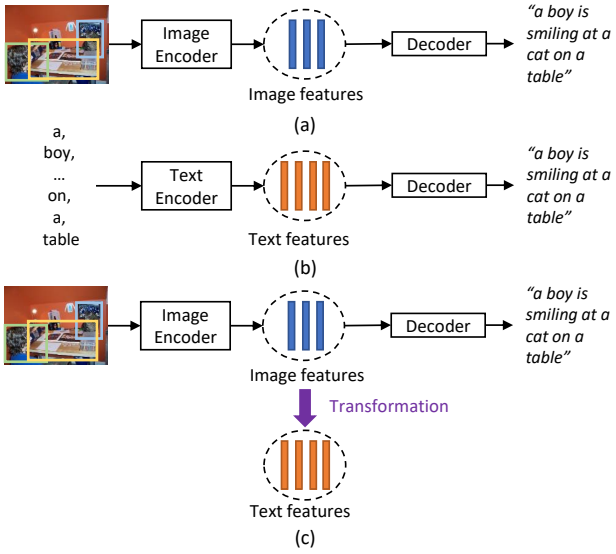


Figure 1: Illustration of the motivation. (a) The standard encoder-decoder framework for image captioning. Every blue bar denotes an image feature for an image region. (b) An autoencoder for text reconstruction. Every orange bar denotes a word representation. (c) The proposed encoder-decoder framework for image captioning. The key component is the module for transforming the image features to text features.

Encoder	Decoder	BLEU-4	CIDEr-D	SPICE
T-T(2)	Att2in	96.2	259.9	41.1
T-T(2)	Up-Down	93.4	249.6	39.6
T-T(6)	T-D	98.4	266.9	42.2
T-I	T-D	39.0	128.0	22.5

Table 1: Comparison of the caption reconstruction results and image captioning results. In the encoder, T-T(n) represents the text encoder stacking n Transformer's encoder layers. T-I represents the image encoder with Transformer. The decoders are taken from the Att2in [27], Up-Down [2] and Transformer [24] (denoted by T-D), respectively. The first block shows the caption reconstruction results. The second block shows the results of a typical image captioning model, which differs from the third model only in the encoder. See Section 5.2 for details.

mances between image captioning and caption reconstruction may be caused by the weak capability of the decoder to bridge the gap between the image and text representations.

The text features from the caption reconstruction are good guidance for generating text-aware image representations, since they almost encode all the information needed for caption generation. One naive method is to learn a transformation from the image feature space to the text feature space, so that the image features can be directly transformed to the text feature space. Denote a image feature by $v_I \in \mathbb{R}^{d_1}$ (a feature vector extracted in an image region) and a text feature (an word embedding) by $v_T \in \mathbb{R}^{d_2}$. Define a transformation $f : \mathbb{R}^{d_1} \to \mathbb{R}^{d_2}$, i.e. $f(v_I; \theta) = v_T$, where θ denotes the parameters. This goal is hard to achieve, since the image regions and words are not one-to-one matched. We propose to align the image features to the corresponding text features and minimize the distance between aligned image features and text features. When forwarding the image features to the decoder, the transformation from image features to text features can be easily captured by the decoder.

The proposed approach is shown in Fig. 1c. It differs from the conventional encoder-decoder framework only in the training phase by adding an Image-to-Text Transformation (ITT) module. It is desired that by modifying the image features such that the difference between the transformed image features and text features is minimized. To achieve this goal, it is important that the transformation function should involve few extra parameters, preferably no extra parameters except the image features, because otherwise the extra parameters may account for the minimization of the distance but the image features change little. We propose to project the image features to the text feature space by multiplying a similarity matrix between image region features and word features. This similarity matrix does not introduce extra parameters, and its function is to align the image features to the text features. By minimizing the distance between the transformed image features and text features, this alignment can be achieved.

It is seen that the image and text alignment is not supervised explicitly because the image captioning datasets do not provide such alignment annotations. To further enhance the alignment learning, a partial-order restricted training strategy based on the fuzzy set theory [31] is proposed to self-supervise the training process.

We conduct experiments on the Karpathy's split [14] of MSCOCO dataset [3]. It is shown that the proposed ITT module and partial-order restriction technique can significantly boost the performances of several baseline models.

2. Related Work

Image Captioning Image captioning has been largely improved by the encoder-decoder framework [26] and attention mechanism [27] these years. Since the image representation as well as the image encoder is an important part in those models, lots of efforts have been made to generate informative image features. We summarize these improvements into two categories. One category is to learn from external knowledge [2, 28]. For example, NIC [26]

uses CNN models pretrained on ImageNet [5] as the image feature extractors. Up-Down model [2] first adopts Faster R-CNN [22] pretrained on the Visual Genome dataset [16] to extract the image features which enables to learn the attention on objects and salient regions. GCN-LSTM [29] goes one step further. It uses GCN to encode the semantic and spatial relationships between detected regions. SGAE [28] uses scene graphs to structure the image representation. Some other works [32, 7] guide the sentence generation explicitly by detected objects, attributes and relationships. The other category is to explore the image captioning datasets, for example, finetuning the image encoder on image captioning datasets [11] and exploring the relationships in image-caption pairs to introduce language inductive bias into features [28]. Our model belongs to the second category. Different from SGAE [28], our model do not introduce extra modules in inference phase, since we expect the image encoder obtaining the capability to bridge the gap between modalities instead of an extra memory module. Besides, we do not use external structural knowledge, such as scene graphs, to learn the relationships between image and text representations. The proposed method learns the relationships in self-supervised manner without external knowledge.

Vision-Language Embedding Tasks involved vision-language embedding, such as visual-semantic embedding [9, 15, 8, 12] and visual grounding [30, 10, 20], requires to generate shared feature space where paired images and texts are supposed to closer than those unpaired. Earlier works [9, 8] on visual-semantic embedding encodes the each image and sentence into single feature vector, which rarely keep the fine-grained knowledge. Recent years, some works [15, 12] turn to explore the fine-grained alignment. In image captioning, however, image representations are supposed to cover more details which are not necessary for visual-semantic embedding. Compared to visual-semantic embedding, in visual grounding, models need to align between concepts and image regions. However, the alignment can be trained under supervision in visual grounding [30, 10, 20]. In image captioning, we hardly get the explicit alignment between words and image regions.

3. Image-to-Text Transformation

The ITT module projects the image features to text features and learns to cluster the words and image regions with the similar meaning. Unlike coarse-grained alignment in image or text retrieval, the ITT module learns to align image regions and words instead of images and sentences. To alleviate the misalignment under weak supervision, we propose partial-order restriction technique which applies latent property of features to supervise the training.

3.1. Image-to-Text Transformation Module

The ITT module takes the text feature V_T and image feature V_I as inputs. In order to keep the semantic details, the text feature is represented as the features of m words in the sentence and the image feature is presented as the features of n image regions. Mathematically, $V_T = [v_T^1, v_T^2, \dots, v_T^m]^\top \in \mathbb{R}^{m\times c}$, where $v_T^i \in \mathbb{R}^c$ is the representation of the i-th word and $V_I = [v_I^1, v_I^2, \dots, v_I^n]^\top \in \mathbb{R}^{n\times c}$, where $v_I^j \in \mathbb{R}^c$ is the representation of the j-th image region.

Note that v_T^i and v_I^j with similar meanings are expected to be clustered in the joint embedding space. However, the image regions and words are unpaired, so that the match between v_T^i and v_I^j is unavailable. Thus we need to first project the image feature to the text feature to get the paired features. This is achieved by multi-head scaled dot attention [24]. The attention matrix $W_{I\to T} = \{w_{I\to T}^{ij}\}_{m\times n}$ between image regions and words is computed by the similarity between V_I and V_T, in which $w_{I\to T}^{ij}$ is the attention weight between the word feature v_T^i and the feature of image region v_I^j. Mathematically,

$$W_{I\to T} = \text{softmax}(\frac{V_T V_I^\top}{\sqrt{c}}). \tag{1}$$

The softmax function is computed along the second dimension (n) to regularize among image regions. Next we compute the aligned image feature $V_{I\to T} = [v_{I\to T}^1, v_{I\to T}^2, \dots, v_{I\to T}^m]^\top$ according to the attention matrix $W_{I\to T}$,

$$V_{I\to T} = W_{I\to T} V_I. \tag{2}$$

Note that the i-row of $V_{I\to T}$ corresponds to a transformed "word" from n image features. It is a weighted sum of n image features, where the weight $W_{I\to T}^{ij} (j = 1, ..., n)$ is higher if the j-th image feature is more similar to V_T^i.

To cluster the words and image regions with similar meanings, the ITT module is trained to minimize the cosine distance between the aligned image feature $V_{I\to T}$ and the text feature V_T, which follows the common practice in word embedding:

$$Loss_{dis} = \sum_{i=1}^{m} \text{TF-IDF}_i (1 - \frac{{v_T^i}^T v_{I\to T}^i}{\|v_T^i\| \|v_{I\to T}^i\|}). \tag{3}$$

$Loss_{dis}$ is the summation of the weighted cosine distance between all v_T^i and $v_{I\to T}^i$.

Note that words do not have equal importance in providing knowledge. For instance, nouns like "woman" and adjectives like "red" are more important than prepositions like "of" and articles like "the". Therefore, the TF-IDF of the i-th word, represented as TF-IDF_i, is used to weight the distance, and can be regarded as an approximation of the importance of each word.

3.2. Partial-Order Restricted Training

Explicit alignment between words and image regions is hardly provided in large-scale datasets because of high cost for annotating. It means that there is no explicit supervision on computing the matrix $W_{I\to T}$. The loss in (3) only provides weak supervision. Here a self-supervised training method, named partial-order restricted training (POR), is introduced to reinforce the relationship between image regions and words.

We first give the definition of the fuzzy set [31]. The fuzzy set A is defined as a pair of a set U and a membership function μ, i.e. $A = (U, \mu)$. The membership function $\mu : U \to [0, 1]$ associates with each point in U a real number interval of $[0, 1]$, with the value of $\mu(x)$ representing the "grade of membership" of x in A.

In this case, the fuzzy set $A_i = (U, \mu_i)$ is defined to represents the assessment of image regions matching the i-th word, i.e. the similarity. $U = \{1, 2, \ldots, n\}$ is the indexes of image regions. $\mu_i : U \to [0, 1]$ is the membership function that assesses the matching degree between image regions and the i-th word. Randomly sampling $\Omega \subseteq \{1, 2, \ldots, m\}$, the inequality should hold:

$$\bigcap_{i\in\Omega} A_i \subseteq \bigcup_{i\in\Omega} A_i. \tag{4}$$

In our case, we define $\mu_i(j) = w^{ij}_{I\to T}$ which is the similarity between i-th word and j-th image region. The membership function of fuzzy set $\bigcap_{i\in\Omega} A_i$ is

$$\mu^{\cap}_{\Omega}(j) = \max_{i\in\Omega} w^{ij}_{I\to T}. \tag{5}$$

For the right side of (4), we define in another way. Given the set Ω, the joint feature vector of Ω is defined as

$$v^{\Omega}_T = \frac{1}{|\Omega|}\sum_{i\in\Omega} v^i_T. \tag{6}$$

Then the similarity between image regions and the word set is computed with (1). That is

$$w_{I\to\Omega} = \text{softmax}(\frac{v^{\Omega}_T V^{\top}_I}{\sqrt{c}}), \tag{7}$$

where $w_{I\to\Omega} = \{w^j_{I\to\Omega}\}_n$. Thus, the membership function of $\bigcup_{i\in\Omega} A_i$ is

$$\mu^{\cup}_{\Omega}(j) = w^j_{I\to\Omega}. \tag{8}$$

Deriving from (4), we have

$$\max_{i\in\Omega} w^{ij}_{I\to T} \geq w^j_{I\to\Omega}, \forall j. \tag{9}$$

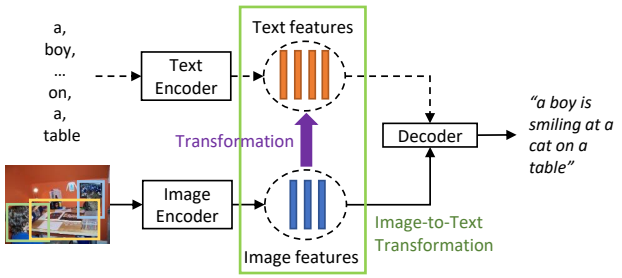


Figure 2: The overall model has four modules, the image encoder, text encoder, Image-to-Text Transformation module and decoder.

To keep the partial-order relation in (9), a loss is introduced as follows:

$$Loss_{POR} = \sum_{k=1}^{N}\sum_{j=1}^{n} \min\left(0, \xi - \max_{i\in\Omega_k} w^{ij}_{I\to T} + w^j_{I\to\Omega_k}\right), \tag{10}$$

where ξ is the margin. For each training sample, we randomly sample Ω for N times, named $\Omega_k, k = 1, 2, \ldots, N$.

The overall loss function for the ITT module is

$$Loss_{embed} = Loss_{dis} + Loss_{POR}. \tag{11}$$

4. Overall Model

The overall model (Fig. 2) consists of four modules: (1) Image encoder encodes the image regions into image feature vectors. (2) Text encoder encodes the words into text feature vectors. (3) ITT module projects the image features into the features space of text representations. (4) Decoder generates the sentences.

Text Encoder The encoder in Transformer [24] is adopted as our text encoder which results in almost perfect reconstruction performance. Inputting the sentence with length m, the text encoder generates the text feature $V_T = [v^1_T, v^2_T, \ldots, v^m_T]^{\top}$, which provides separate but context-aware feature vector for each word.

Image Encoder The architecture of the image encoder can be arbitrary image encoders proposed before. However, some image encoders which are pretrained from other datasets and fix when extracting image features on image captioning datasets, such as Up-Down model [2] and Att2in model [27] needs to add extra layers. In this way, the image encoder can be refined when training together with the ITT module. To stay the same with the text encoder, the encoder modules in Transformer are adopted to build the extra layers. The final image feature is $V_I = [v^1_I, v^2_I, \ldots, v^n_I]^{\top}$, which provides a single feature vector for each image region.

ITT Module The ITT module takes the image features and text features as inputs. Since the ITT module is trained together with the image encoder, it could back propagate to the image encoder and thus refines the image encoder.

Decoder Both the image features and text features could be the inputs of the decoder. When training the caption construction (Fig. 1b), we forward the text features to decoder. When switching to image captioning, the decoder takes the image features as input while the text features are only used to refine the image features through the ITT module.

Training We adopt a three-stage training process. At the first stage, the caption reconstruction is trained to obtain the text representations. In this stage, the text encoder and decoder (dotted path in Fig. 2) is involved in the training. The training loss for caption reconstruction is the cross-entropy (XE) loss, which is

$$Loss_{recon} = -\sum_{i=1}^{m} p(w_i|w_1, w_2, \cdots, w_{i-1}; \theta). \quad (12)$$

After finishing the text reconstruction stage, we fix the parameters in the text encoder.

At the second stage, the model is optimized by minimizing the cross-entropy loss for caption generation. Meanwhile, the ITT module is also trained with (11). In this stage, the image encoder, ITT module and decoder are involved in training. The loss for this stage is the summation of cross-entropy loss and $Loss_{embed}$ for ITT module:

$$\begin{aligned} Loss_{caption} = &-\sum_{i=1}^{m} p(w_i|w_1, w_2, \cdots, w_{i-1}; \theta) \\ &+ \alpha Loss_{embed}, \end{aligned} \quad (13)$$

where α is a hyperparameter to weight the item.

At this last stage, we adopt Self-critical Sequence Training (SCST) [23] to optimize towards the evalutaion metrics such as CIDEr-D [25]. Only the image encoder and decoder are involved in this stage.

Inference Since the captions are not provided in inference and the image encoder are assumed to learn the language inductive bias, we simply use the image encoder and decoder to generate the captions.

5. Experiments

5.1. Datasets and Evaluation Metrics

The proposed methods are validated on MSCOCO captioning dataset [3]. Each image is captioned with 5 sentences. The Karpathy's split from [14] is used, which have 113287 image in the training set, 5000 in the validation set and 5000 in the test set. The experimental results are reported on the test set of the Karpathy's split and evaluated on BLEU [19], METEOR [6], ROUGE-L [17], CIDEr-D [25] and SPICE [1].

5.2. Implementation Details

To prepare the annotations, we first tokenize the words in captions and replace those occurring less than 5 times with UNK. The vocabulary contains 9487 words. The maximum length of captions is set to 16. For those exceeds the maximum length, we simply truncate the sentences.

Transformer We use primary image features extracted from Faster R-CNN [22] which is pretrained on the Visual Genome dataset [16], as presented in [2]. In the image encoder, the primary image features are first projected to lower-dimensional space (512-dim) by an embedding layer and then forwarded to $N = 6$ identical layers. Each of them has a multi-head attention module and a fully-connected feed forward network. The primary image features have $d_{init} = 2048$ dimensions. The outputs of each layer have $d = 512$ dimensions while the intermediate outputs inside layers have $d_{ff} = 2048$ dimensions. The text encoder is almost the same as the image encoder, except that we embed each word and apply positional embedding at the beginning. The decoder is also stacked with $N = 6$ identical layers, the same as the structure in [24].

The Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.98$ and $\epsilon = 10^{-9}$ is adopted to train the model. The text reconstruction and image captioning with the loss in (12) and (13) respectively are trained for 15 epochs respectively. For the learning rate schedule, we follow [24], i.e. $learning\_rate = d^{-0.5} \min(iter^{-0.5}, iter \cdot warmup^{-1.5})$, where $iter$ refers to the number of iterations, and $warmup = 20000$. When training the image captioning with SCST [23], we use the initial learning of 10^{-5} and decay by 0.8 every 3 epochs. This step is trained for 20 epochs. We use beam search with $beam\_size = 2$ for outputs. The batch size of 10 is used throughout the experiment. We set the margin $\xi = 0$ in (10) and $\alpha = 1.0$ in (13).

Att2in Att2in [27, 23] uses the features extracted from ImageNet [5] as the primary image features. To refine the image encoder, we add 2 extra layers. The modified image encoder and text encoder are the same with the Transformer model mentioned above, except that N is set to 2 instead of 6. The Adam optimizer with initial learning rate of $5^{-}4$ is adopted for the stages of caption reconstruction and image captioning (without SCST). These two stages are trained for 20 epochs respectively. At the last stage, SCST is trained for 20 epochs. The initial learning rate is 10^{-5}.

Up-Down The Up-Down model [2] takes the image features extracted from Faster R-CNN as inputs. The Up-Down model used in our experiments is also modified as the way to Att2in. The hyperparameters for the optimizer and training process are the same as Att2in.

5.3. Quantitative Results

In Table 2, the performance of the proposed models and the comparison with baselines are reported. We conduct our experiments on three baselines, i.e. Att2in, Up-Down and Transformer. Note that in the original Att2in and Up-Down model, the image encoders are fixed without finetuning on image captioning dataset. Since the proposed module improves the baselines by refining the image encoder, we add two extra layers to the original Att2in and Up-Down model. To make fair comparison, the performance of modified baselines are reported in Table 2.

The results show that the proposed method outperform all the three baselines both with and without SCST, which validate the efficiency of the proposed module.

The experiments on the efficiency of partial-order restricted training (POR) on Transformer are also conducted. By adopting POR, the performance further improves from 129.9 to 130.9.

Table 3 shows the comparison between our model and previously proposed models. our single model achieves 39.5 on BLEU-4, 130.9 on CIDEr-D and 23.0 on SPICE, which outperforms many existing models including SGAE, GCN-LSTM and AoANet with a noticeable margin.

5.4. Qualitative Analysis: Visualizing the Alignments

The alignments between image regions and words are visualized to validate that the proposed ITT module with partial-order restricted training can learn reasonable alignments. The top-3 similar image regions aligned to each word are shown in Fig. 5. During visualization, we re-regularize the similarity of the top-3 scored image regions first and apply Gaussian mask on the regions. The Gaussian mask is used to take the center of the rectangle regions as the mask's center, and the width and height of the region is used as σ_x and σ_y. We only use the first "head" attention matrix in the multi-head attention module. Note that the visualizations to other heads are also consistent. In the visualization, the regions masked with lighter blue are those attract more attention.

Fig. 5 shows the comparison between models without partial-order restricted training and with partial-order restricted training. For the caption "a bowl of broccoli on a cutting board", the model without partial-order restricted training can only correctly align to the word "brocolli" and the aligned regions through the sentences are disordered, as is shown in the first set of Fig. 5a. When it comes to the the results with partial-order restricted training (the second set of Fig. 5a), it can easily tell that the aligned regions move from the regions of bowls and broccoli to the regions of the surrounding which is the cutting board in this case. For the caption "a cat with its nose next to a sandwich on a white plate", the model without partial-order restricted training almost attend to all the regions of cat, sandwich and plate for each word (the first set of Fig. 5b). The model with partial-order restricted training works much better (the second set of Fig. 5b). It attends to the region of the cat at the first few words and then move the region of the plate and sandwich. It shows that with partial-order restricted training, the model can learn better alignments to words and the aligned regions move along with the changes of described objects in sentences.

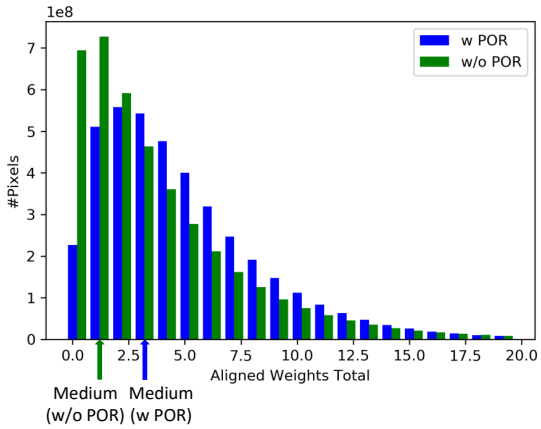


Figure 3: The number of pixels against the total aligned weight on each pixel. With POR, less pixels are assigned to low scores, which means the distribution of aligned scores are more clustered to certain regions.

Statistical Analysis on the Distribution of Aligned Scores The distribution of the aligned scores is also plotted, as illustrated in Fig. 3. For each sentence, the noun words extracted by Part-of-Speech provided in NLTK library are gone through and the aligned scores are added to each image region (all the pixels inside). We then group the total aligned scores of each pixel into 20 buckets. The i-th bucket represents the range $(\frac{i}{10}, \frac{i+1}{10}]$. Then the numbers of pixels against aligned scores of pixels are plotted. Fig. 3 shows that, when adopting POR, the model tends to assign higher scores to more pixels and lower scores to less pixels. This means that the attended regions are less disperse, which is consistent with the goal of alignment.

In 4, we plot the loss in (10) and CIDEr against the epoch. When adding the (10) to previous models, the CIDEr

Method	SCST	BLEU-1	BLEU-4	METEOR	ROUGE-L	CIDEr-D	SPICE
Att2in [27, 23]		-	31.3	-	54.3	101.3	26.0
Att2in*		75.4	34.0	26.5	55.1	108.6	20.2
Att2in* + ITT		75.3	34.0	26.6	55.2	109.4	20.1
Att2in [23]	✓	-	33.3	-	55.3	111.4	26.3
Att2in*	✓	77.0	35.1	26.8	56.1	116.0	20.5
Att2in* + ITT	✓	**77.2**	**35.3**	**27.0**	**56.5**	**117.1**	**20.8**
Up-Down[2]		77.2	36.2	27.0	56.4	113.5	20.3
Up-Down*		76.8	35.8	27.5	56.6	113.7	20.7
Up-Down* + ITT		76.6	35.8	27.7	56.7	114.2	20.9
Up-Down[2]	✓	**79.8**	36.3	27.7	56.9	120.1	21.4
Up-Down*	✓	78.9	**36.9**	27.8	57.5	122.9	**21.3**
Up-Down* + ITT	✓	79.1	36.8	**28.0**	**57.6**	**123.5**	**21.3**
Transformer [18]		76.4	35.4	27.9	56.5	114.5	21.1
Transformer + ITT (w/o POR)		76.2	35.6	28.0	56.5	115.2	21.1
Transformer + ITT		76.4	36.4	28.4	57.1	117.1	21.4
Transformer [18]	✓	80.1	39.0	28.8	58.6	128.0	22.5
Transformer + ITT (w/o POR)	✓	80.9	39.3	29	58.8	129.9	22.8
Transformer + ITT	✓	**81.1**	**39.5**	**29.2**	**59.1**	**130.9**	**23.0**

Table 2: The comparison with Att2in, Up-Down and Transformer. The performance of both original baseline models and modified baseline models are shown in the table. The modified baseline models are marked with "*". Our module integrating into the baselines are presented with "+ ITT". The models with ITT, if not specified, are trained with POR training, while models with "(w/o POR)" are trained with ITT but not POR training. The "SCST" column indicates whether trained with SCST. The best result in each block are highlighted with bold.

Method	BLEU-1	BLEU-4	METEOR	ROUGE-L	CIDEr-D	SPICE
Up-Down [2]	79.8	36.3	27.7	56.9	120.1	21.4
RFNet [13]	79.1	36.5	27.7	57.3	121.9	21.2
RFNet$^{\Sigma}$ [13]	80.4	37.9	28.3	58.3	125.7	21.7
GCN-LSTM [29]	80.5	38.2	28.5	58.3	127.6	22.0
SGAE [28]	80.8	38.4	28.4	58.6	127.8	22.1
SGAE$^{\Sigma}$ [28]	81.0	39.0	28.4	58.9	129.1	22.2
AoANet [11]	80.2	38.9	29.2	58.8	129.8	22.4
Transformer [18]	80.3	38.9	28.8	58.6	127.5	22.6
Transformer + ITT	**81.1**	**39.5**	**29.2**	**59.1**	**130.9**	**23.0**

Table 3: Quantitative Comparison with previous models. Our single model outperforms the previous models on all evaluation metrics. $^{\Sigma}$ indicates ensemble models while others are single models. The best result on each evaluation metric is highlighted with bold. The first block is the performances published and the second block is our best single model.

further improves from 112.9 to 114.6 with 5 epochs, which confirm the capability of adopting the loss in (10).

6. Conclusions

In this paper, a Image-to-Text Transformation module is proposed to bridge the gap between image and text representations. This module learns to project the image features to text features. This relationship is encouraged by modifying the image features, and it is expected the decoder has the ability to learning this relationship, and as a consequence, generate better captions given an image. In order to obtain better alignment between image regions and words, we further propose a partial-order training strategy which self-supervise the training of the alignment.

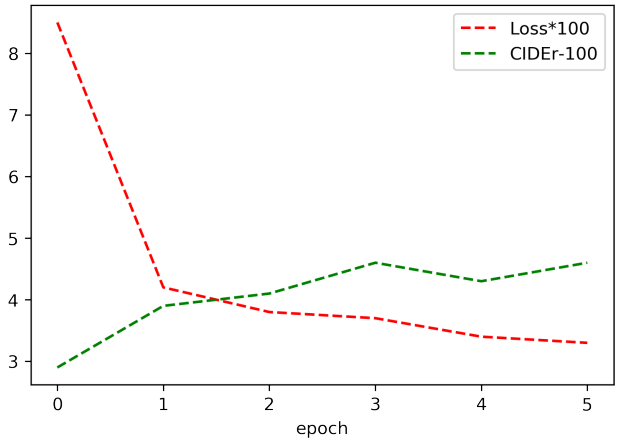


Figure 4: Loss in (10) and CIDEr during training. When adding the (10) to previous models, the CIDEr further improves from 112.9 to 114.6.

References

[1] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. Spice: Semantic propositional image caption evaluation. In *European Conference on Computer Vision*, pages 382–398. Springer, 2016.

[2] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6077–6086, 2018.

[3] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015.

[4] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua 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)*, pages 1724–1734, Doha, Qatar, Oct. 2014. Association for Computational Linguistics.

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[6] Michael Denkowski and Alon Lavie. Meteor 1.3: Automatic metric for reliable optimization and evaluation of machine translation systems. In *Proceedings of the sixth workshop on statistical machine translation*, pages 85–91. Association for Computational Linguistics, 2011.

[7] Zhihao Fan, Zhongyu Wei, Siyuan Wang, and Xuan-Jing Huang. Bridging by word: Image grounded vocabulary construction for visual captioning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6514–6524, 2019.

[8] Faghri Fartash, J. Fleet David, Ryan Kiros Jamie, and Fidler Sanja. VSE++: improving visual-semantic embeddings with hard negatives. In *British Machine Vision Conference*, page 12, 2018.

[9] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. In *Advances in neural information processing systems*, pages 2121–2129, 2013.

[10] Ronghang Hu, Marcus Rohrbach, Jacob Andreas, Trevor Darrell, and Kate Saenko. Modeling relationships in referential expressions with compositional modular networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1115–1124, 2017.

[11] Lun Huang, Wenmin Wang, Jie Chen, and Xiao-Yong Wei. Attention on attention for image captioning. In *International Conference on Computer Vision*, 2019.

[12] Vendrov Ivan, Kiros Ryan, Fidler Sanja, and Urtasun Raquel. Order-embeddings of images and language. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.

[13] Wenhao Jiang, Lin Ma, Yu-Gang Jiang, Wei Liu, and Tong Zhang. Recurrent fusion network for image captioning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 499–515, 2018.

[14] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3128–3137, 2015.

[15] Ryan Kiros, Ruslan Salakhutdinov, and Richard S Zemel. Unifying visual-semantic embeddings with multimodal neural language models. *arXiv preprint arXiv:1411.2539*, 2014.

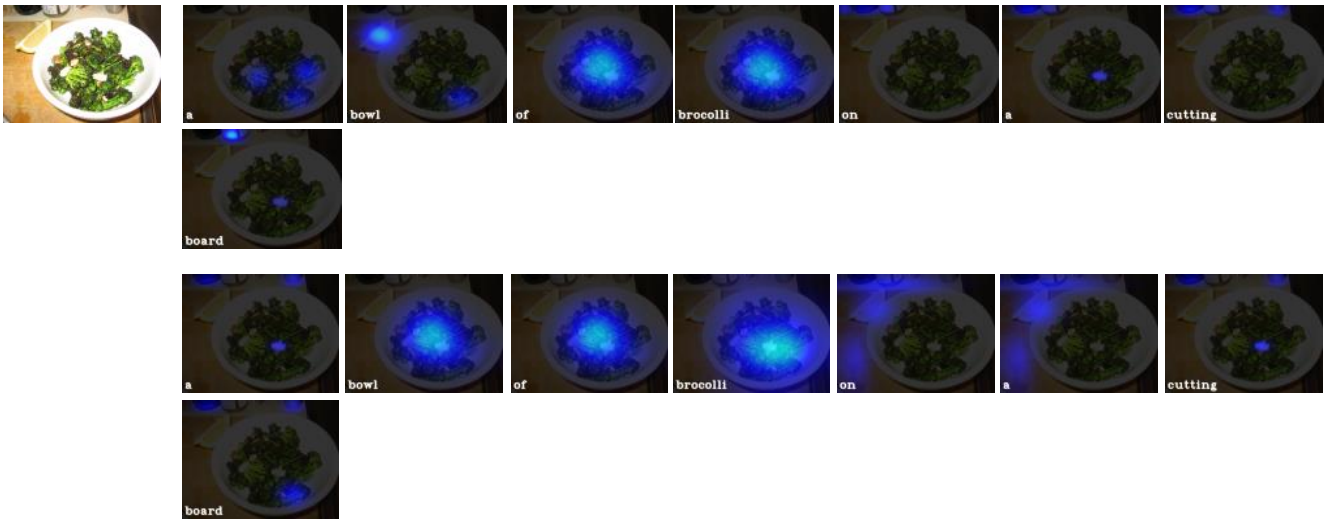
[16] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123(1):32–73, 2017.

[17] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004.

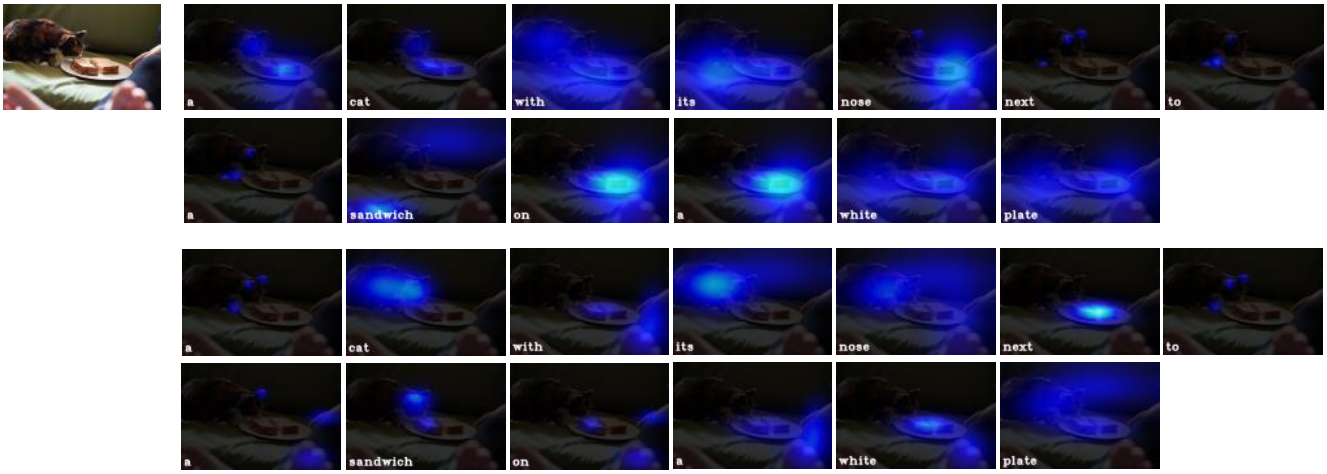
[18] Ruotian Luo, Brian Price, Scott Cohen, and Gregory Shakhnarovich. Discriminability objective for training descriptive captions. *arXiv preprint arXiv:1803.04376*, 2018.

[19] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics, 2002.

[20] Bryan A Plummer, Arun Mallya, Christopher M Cervantes, Julia Hockenmaier, and Svetlana Lazebnik. Phrase localization and visual relationship detection with comprehensive image-language cues. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1928–1937, 2017.



(a) “a bowl of brocolli on a cutting board”.



(b) “a cat with its nose next to a sandwich on a white plate”.

Figure 5: Visualizing the alignments. In each subfigure, the first set is the alignment of the model without POR, while the second set is the alignment of the model with POR. Original images are shown on the left.

[21] Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the IEEE international conference on computer vision*, pages 2641–2649, 2015.

[22] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[23] Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. Self-critical sequence training for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7008–7024, 2017.

[24] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[25] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575, 2015.

[26] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164, 2015.

[27] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption gen-

eration with visual attention. In *International conference on machine learning*, pages 2048–2057, 2015.

[28] Xu Yang, Kaihua Tang, Hanwang Zhang, and Jianfei Cai. Auto-encoding scene graphs for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10685–10694, 2019.

[29] Ting Yao, Yingwei Pan, Yehao Li, and Tao Mei. Exploring visual relationship for image captioning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 684–699, 2018.

[30] Licheng Yu, Hao Tan, Mohit Bansal, and Tamara L Berg. A joint speaker-listener-reinforcer model for referring expressions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7282–7290, 2017.

[31] Lotfi A Zadeh. Fuzzy sets. *Information and control*, 8(3):338–353, 1965.

[32] Yue Zheng, Yali Li, and Shengjin Wang. Intention oriented image captions with guiding objects. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8395–8404, 2019.