Geometry Attention Transformer with Position-aware LSTMs for Image Captioning[2]

Chi Wang[1], Yulin Shen[1], Luping Ji[*]

School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, P.R. China

Abstract

In recent years, transformer structures have been widely applied in image captioning with impressive performance. For good captioning results, the geometry and position relations of different visual objects are often thought of as crucial information. Aiming to further promote image captioning by transformers, this paper proposes an improved *Geometry Attention Transformer* (GAT) model. In order to obtain geometric representation ability, two novel geometry-aware architectures are designed respectively for the encoder and decoder in our GAT. Besides, this model includes the two work modules: i) a geometry gate-controlled self-attention refiner, for explicitly incorporating relative spatial information into image region representations in encoding steps, and ii) a group of position-LSTMs, for precisely informing the decoder of relative word position in generating caption texts. The experiment comparisons on the datasets MS COCO and Flickr30K show that our GAT is efficient, and it could often outperform current state-of-the-art image captioning models.

Keywords: Image captioning, Encoder-decoder framework, Gate-controlled geometry attention, Position-aware LSTM

---

[*]Corresponding author.
 Email addresses: chi.w@std.uestc.edu.cn (Chi Wang),
yulinshen@std.uestc.edu.cn (Yulin Shen), jiluping@uestc.edu.cn (Luping Ji)
 [1]These authors contributed equally to this work.
 [2]The source code is available at https://github.com/UESTC-nnLab/GAT

1. Introduction

Image Captioning is a challenging problem in computer vision[1]. It aims to automatically describe an image using meaningful text, translating an image into natural language. It often requires to not only recognize what visual objects an image contains but also capture what those objects are doing accurately and even to tell us what the interrelations of different objects are. Image captioning could build a powerful bridge between visual images and human languages. With captions, people could better understand an image[2]. Image captioning has been thought of as one quite useful technology in the image analysis field. In recent years, it has been attracting more and more research attention, and diverse captioning models have been emerging for image captioning [3, 4, 5, 6].

For image captioning, its early models could often be roughly classified into two primary categories [7]. One is usually dependent on image retrieval. It generates image captions by analyzing image correlation and then retrieving candidate texts from existing caption pools [8, 9]. In contrast, the other is often summarized as the template-based category. This kind of methods builds captions typically through the syntactic and semantics analysis on images, with visual concept detecting, sentence template matching and optimizing [10, 11, 12]. The most obvious characteristic of these two early categories is that hardcoded rules and hand-engineered features are in common use so that the reliability and accuracy of captioning fluctuate heavily [7].

In recent years, facilitated by booming artificial intelligence, neural networks, the third type of method for image captioning, have been becoming one of the most exciting and powerful tools. For example, we could easily see the huge improvements by early work of deep neural models in [13, 14, 15]. Besides, more neural network based models, such as the multimodal learning [16, 17], the encoder−decoder framework [18], the attention mechanism [19, 20], the compositional architectures [21], the describing method of novel objects [22] and the deep bifurcation network [23], have been proposed one after another.

Through the survey of a large number of publications, it is easy to figure out that Transformer with an encoder-decoder structure is becoming one of the most main-

stream models for image captioning [24]. In general, in a typical transformer model, a given image is often mapped into a set of intermediate vectors by an encoder of CNN-based networks, and then the target caption of this image could be generated word by word through a decoder of RNN-based network [4, 16, 18, 25].

In addition, as mentioned before, the classic attention mechanism has also been widely utilized in image captioning [19, 20]. One of its advantages is that it could often effectively guide the decoder to focus on some specific information, such as context correlation, while generating image captions. On traditional attention models, the self-attention mechanism [26] is often thought of as an effective improvement to the classic version. In recent years, this kind of self-attention model has also quickly emerged in image captioning [27, 28]. From the task perspective, a self-attention module is often regarded as a feature mapper between a set of queries and key/value pairs.

In image captioning, the geometry and position relations of image objects are often very necessary for accurately describing a given image. For example, "*a boy standing on a skateboard*" and "*a boy raises a skateboard in his hands*" indicate two different semantics. However, they are likely to mean the same thing if without relative geometry information. In most existing Transformer-based models with attention modules, we could find that the inherent geometry and position relations between image components have not yet been paid enough attention to and fully utilized [25]. Moreover, although some methods, such as the GSA proposed in [24], carefully considers geometry relations in encoder, more detailed position decoding has not been well solved yet.

Geometry and position relations are so important that accurate image captioning models often have to emphasize them. In order to take full advantage of geometry and position clues, we firstly design an improved encoder model with Geometry self-attention Refiner (GSR) in our transformer structure. This model could explicitly incorporates geometry information into a vanilla self-attention module, implementing the transformation from original appearance queries to geometrical ones. And then, the refined attention weights are calculated to get a mean of weighted appearance values. The geometrical queries and keys in our model are linearly derived from the bounding box coordinates of each component object. In this way, every object could always ob-

tain its appearance features and geometry correlations with each other. In task details, they both are very crucial for a decoder to generate the right caption word sequences with position and semantic.

Furthermore, in the decoder of Transformer, some typical methods often inject the word position information of expected caption texts by adding *sine* or *cosine* operation on the top of the word embedding layer, such as the ones in [26, 27]. This kind of position injection mechanism has been proved effective, in spite of not always conforming to semantics. To overcome this weakness of position decoding, we design a group of position-LSTMs to model the word order of caption texts. It parses a caption sentence word by word in sequence, ensuring the right order of words. In addition, the hidden layers of position-LSTMs could also remember the information that the decoder has generated. This kind of design is very helpful to self-attention modules to concentrate on particular position parts. In the meanwhile, the information stored in hidden states of LSTMs also contains the geometry information of component objects.

Finally, by the integration and collaborative optimization of GSR and the position-LSTM, our captioning model has achieved better performance than state-of-the-art ones on the two classic datasets: MS COCO and Flickr30k. In summery, the main contributions of this paper could be listed as follows:

- We propose an improved image captioning model GAT of transformer structure, which could capture the geometry relations of component objects. Geometry capturing ability could often enable our captioning model to obtain a sense of *'where are target objects'* and *'where is the captioning model currently looking at'*.
- We design an encoder improved by GSR. It could encapsulate the relative geometry information of objects. Moreover, it takes full advantage of geometry relations to refine object representations.
- We develop a decoder promoted by position-LSTMs. It could transport the order information of generated cation words with relative position relations. Therefore, it could execute position reasoning to decide which word to generate next.

4

2. Related Work

In recent years, a king of encoder-decoder transformer architecture with attention modules has been attracting wide research enthusiasm. In the meanwhile, more and more methods have been developed for image captioning, such as the one with soft & hard attention [29], the one with residual connections [30], and the meshed-memory transformer [31].

This type of transformer architecture, shown in Figure 1, can bee clearly seen in [24, 27, 32, 33]. Typically, it consists of an encoder and a decoder, one for extracting visual object features and the other for generating caption text. In detail, both the encoder and the decoder, represented by Fig. 1(a), are designed as a group of multi-layer residual networks. Moreover, we could also always see a classic component of '**Self-Attention**', shown by Fig. 1(b), in the encoder and decoder [27, 28].

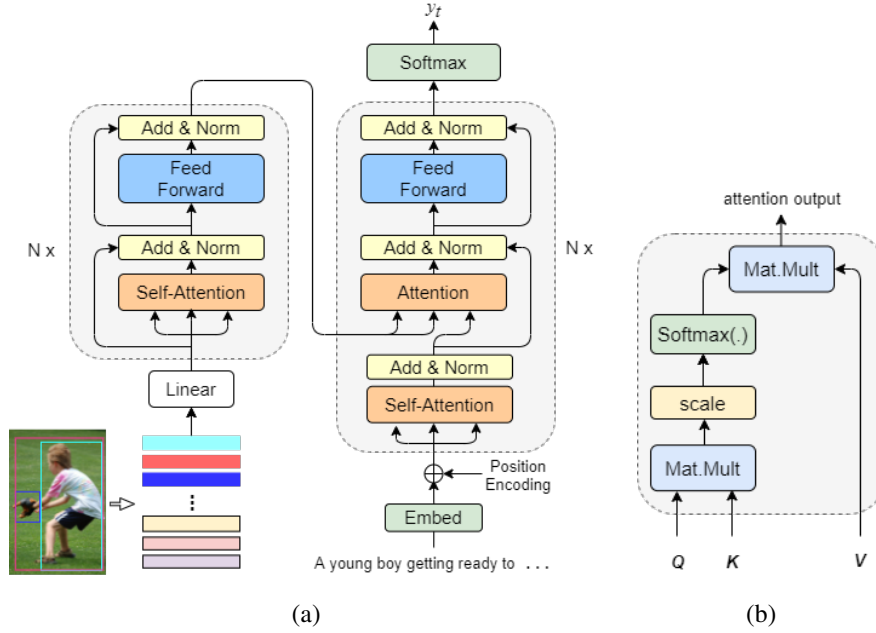(a)   (b)

Figure 1: a typical deep neural network of image captioning [24], (a) encoder-decoder structure, (b) a classic self-attention component widely integrated in both encoder and decoder.

In the typical captioning framework illustrated above, besides traditional object appearance features, the input to the encoder contains some novel geometry cues, such

as object center, height and width [24]. In the decoder, furthermore, a position encoding operation has also been employed to encode the accurate word sequence of the caption text. And in [24], a simple 'sinusoidal function' is utilized to fulfill this task. Experiments have also proved that this kind of improvement by geometry and position features is much efficient.

Moreover, in terms of the attention modules in image captioning, there are also all kinds of variants having been explored. For examples, we could easily see the multi-head attention in [26], the gate-controlled attention [21], the fully attentive paradigms [28, 34], the meshed-connection attention [31], the dual attention [35], etc.

To our knowledge, this kind of encoder-decoder architecture with attention components introduced above is believed to be one of the most potential baselines for image captioning [24, 27]. In the future, more ingenious network design, more sophisticated object features, more functional geometry relations, and position encoding strategies are still kept huge research expectations for further promoting image captioning methodology [36].

3. Proposed Method

Our method generates grounded captions by attending to specific image regions at each step. In terms of structure, it still retains a classic encoder-decoder architecture. In the encoder, a geometry self-attention refiner (GSR) is designed to optimize image representations. Furthermore, in the decoder, a module for accurately decoding word sequences by LSTM is adopted. More details about it can be seen in Figure 2.

Given an input image $\boldsymbol{I}$, in figure 2, we use ($\boldsymbol{X}_A \in \mathbb{R}^{N \times d}$) to represent a set of image appearance regions, where N is the index of image region, and d is the dimensionality of region data. Moreover, ($\boldsymbol{y} = \{\boldsymbol{y}_1, ..., \boldsymbol{y}_T\}$) to indicate the caption word vectors corresponding to $\boldsymbol{I}$.

3.1. Gate-controlled Geometry Self-attention Refiner

Besides appearance information $\boldsymbol{X}_A$, we incorporate geometry information into the vanilla self-attention to refine the representation of the objects for the reason that it is beneficial to comprehend the intrinsic relations among different objects. Therefore, we
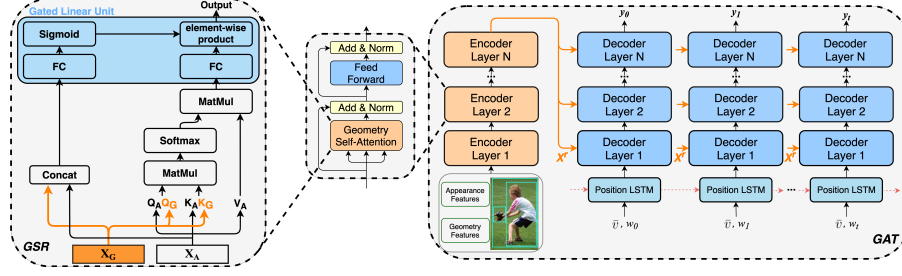
6

Figure 2: The detailed architecture of our GAT. The GSR module in encoder generates some weighted average given appearance features $\boldsymbol{X}_A$ and geometry features $\boldsymbol{X}_G$, which is further refined by a gate-controlled unit. Moreover, a group of position-LSTMs precisely injects word order information into the decoder.

propose the Geometry Self-attention Refiner (GSR) to refine the visual information by taking into account the geometry information of the objects.

Given the geometry features of the objects $\boldsymbol{X}_g \in \mathbb{R}^{N \times 5}$, each row of $\boldsymbol{X}_g$ is a 5-dimensional vector:

$$(x_{min}, y_{min}, x_{max}, y_{max}, S) \tag{1}$$

where (x_{min}, y_{min}), (x_{max}, y_{max}) and S represent the top-left coordinates, the bottom-right coordinates of bounding box, and the relative size to whole image, respectively. And, all of them are normalized to $(0, 1)$.

Firstly, we obtain $\boldsymbol{X}_G \in \mathbb{R}^{N \times d_m}$ by embedding $\boldsymbol{X}_g$ to a higher dimension form with an embedding layer followed by a ReLU non-linearity. Then we combine appearance information with geometry information by modifying the queries and keys as follows:

$$\boldsymbol{Q}^{'} = [\boldsymbol{X}_A \boldsymbol{W}_{\boldsymbol{Q}_A}; \boldsymbol{X}_G \boldsymbol{W}_{\boldsymbol{Q}_G}] \tag{2}$$

$$\boldsymbol{K}^{'} = [\boldsymbol{X}_A \boldsymbol{W}_{\boldsymbol{K}_A}; \boldsymbol{X}_G \boldsymbol{W}_{\boldsymbol{K}_G}] \tag{3}$$

where $\boldsymbol{W}_{\boldsymbol{Q}_A}$, $\boldsymbol{W}_{\boldsymbol{K}_A}$ are learned appearance matrices and $\boldsymbol{W}_{\boldsymbol{Q}_G}$, $\boldsymbol{W}_{\boldsymbol{K}_G}$ are learned geometry matrices, respectively, and are all of dimension $\mathbb{R}^{d_m \times d_m}$. Here [;] indicates concatenation. $\boldsymbol{Q}^{'}$ and $\boldsymbol{K}^{'}$ combine appearance information with geometrical information, which can be seen as a complementary to acquire fine-grained knowledge of

the objects. Then the attention can be calculated as:

$$\mathbf{\Omega}' = \frac{\mathbf{Q}'\mathbf{K}'^{\mathbf{T}}}{\sqrt{2 \times d_k}} \tag{4}$$

Then the output of the GSR can be calculated as:

$$Attention_g(\mathbf{X}) = softmax(\mathbf{\Omega}')\mathbf{V_A} \tag{5}$$

Similar to the original Transformer[26], our geometry refiner is also implemented in a multi-head self-attention framework. In this framework, geometry attention will be repeated h times (so, called h heads). In every time of repetition, it always adopts a group of different projection matrices $(\mathbf{W}_{\mathbf{Q}_A}^k, \mathbf{W}_{\mathbf{K}_A}^k$ and $\mathbf{W}_{\mathbf{V}_A}^k)$ for extracting appearance features, and $(\mathbf{W}_{\mathbf{Q}_G}^k$ and $\mathbf{W}_{\mathbf{K}_G}^k)$ for capturing geometrical features, where $k \in (1, 2, ..., h)$. All the results generated from h self-attention heads will also be concatenated together by a linear projection and then fed forward to the next layer.

Moreover, inspired by the work of Huang et al. [20], we further design a parallel Gate-controlled Linear Unit (GLU) to continually refine the output of the geometry self-attention module (the left region of Figure.2). This GLU still takes the current context $(\tilde{\mathbf{c}} = [\mathbf{X}_A; \mathbf{X}_G])$ as its input to generate a control matrix $\mathbf{Gate_{Ctrl}}$ for modulating final attention output by Hadamard product. This group of neural computation can be mathematically expressed by Eqn. (6):

$$\begin{cases} \mathbf{Gate_{Ctrl}} = sigmoid(\mathbf{W}_g\tilde{\mathbf{c}} + \mathbf{b}_g) \\ \mathbf{Output} = \mathbf{Gate_{Ctrl}} \odot (\mathbf{W}_i\tilde{\mathbf{a}} + \mathbf{b}_i) \end{cases} \tag{6}$$

where $\mathbf{W}_g, \mathbf{W}_i \in \mathbb{R}^{d_m \times d_m}$ are two groups of weights, and $\mathbf{b}_g, \mathbf{b}_i \in \mathbb{R}^{d_m}$ are two groups of neuron biases. Furthermore, $\odot$ denotes a Hadamard product. In terms of structure, our self-attention component and gate-controlled unit can be stacked into multiple layers to further optimize object representations.

Subsequently, the self-attention output is embedded with its original input by the operation of add & Normalization, then transferred to its next layer, $i.e.$, the *Feed-Forward* sub-lay network shown in the middle of Figure 2. In details, this *Feed-*

Forward sub-layer usually contains two groups of nested affine transformations with the activation function, $ReLU(x) = max(0, x)$. Therefore, this group of processing step could be uniformly formulated by

$$FF(\boldsymbol{X}) = max(0, \boldsymbol{X}\boldsymbol{W}_1 + \boldsymbol{b}_1)\boldsymbol{W}_2 + \boldsymbol{b}_2 \tag{7}$$

where $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$ are two groups of weight matrices to be learned, $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$ are two groups of biases. Also following the *Transformer* architecture in [26], each sub-layer is successively followed by a residual connection and a layer normalization. Therefore, this group of computation could be mathematically expressed by

$$\begin{cases} \boldsymbol{Z} = LayerNorm(\boldsymbol{X} + Attention_g(\boldsymbol{X})) \\ \boldsymbol{X}^r = LayerNorm(\boldsymbol{Z} + FF(\boldsymbol{Z})) \end{cases} \tag{8}$$

3.2. Position-aware Self-attention Decoder

i) Encoding positions via LSTM.

In order to address the order problem of a sequence, existing works usually inject some '*positional encoding*' into word embedding by a sinuous function. In our method, we further exploit a position representation strategy and then devise a distinctive LSTM-based position encoding mechanism.

The position-LSTM in our decoder of transformer could model the order of image caption words in decoding process. At each time step t, we define the input of the position-LSTM as:

$$\boldsymbol{x}_t = [\boldsymbol{w}_t, \bar{\boldsymbol{v}}] \tag{9}$$

where $\boldsymbol{w}_t$ is the word embedding derived by a one-hot vector, and $\bar{\boldsymbol{v}} = \frac{1}{k}\Sigma_i \boldsymbol{v_i}$ denotes the mean pooling of image features. Therefore, we could obtain:

$$\boldsymbol{h}_t, \boldsymbol{c}_t = LSTM\big(\boldsymbol{x}_t, (\boldsymbol{h}_{t-1}, \boldsymbol{c}_{t-1})\big) \tag{10}$$

In view of the sequentiality output of LSTM, $\boldsymbol{h}_t$ could often be treated as an encoding of sequence order for caption words. Such an encoding could provide its subsequent

decoder with precise position information in two aspects. One is what words have been generated by far, and the other is where the decoder is for directing the current relative position of objects. In addition, this position encoding will also be updated at each time step, and meanwhile, it could also guide the decoder to focus on its correlated regions in geometry-aware mode adaptively.

ii) Injecting position encodings into the decoder.

Given a set of refined region features $\boldsymbol{X}^r$ and a group of current position encoding vectors $\boldsymbol{h}_t$, our decoder generates a multi-layer structure sequence $\hat{\boldsymbol{y}}$. As shown in Eq.(7), each decoding layer of our model contains a sub-layer of attention and a sub-layer of feed-forward output. Sometimes, such a layer may look a little redundant. Therefore, in experiments, we consider removing one self-attention sub-layer so as to compress network size and analyzing the influence of removing the attention sub-layer on model performance.

Furthermore, the group of decoders at the bottom firstly achieves the current position encoding $\boldsymbol{h}_t$ to calculate the $\boldsymbol{Query}$, and uses the region features $\boldsymbol{X}^r$ to calculate the $\boldsymbol{Keys}$ and $\boldsymbol{Values}$. Then, the scaled attention results by dot-product could be obtained from a multi-head pattern. For the back layers of decoder, however, they always adopt the output of previous layers as the $\boldsymbol{Query}$. As a result, at each time step t, through the conditioning on the output of top decoder, the distribution of the t-th word could be calculated by:

$$p(\boldsymbol{y}_t \mid \boldsymbol{y}_{1:t-1}) = softmax(\boldsymbol{W}_p\boldsymbol{\mathcal{F}}_t + \boldsymbol{b}_p) \tag{11}$$

where $\boldsymbol{W}_p \in \mathbb{R}^{|\Sigma| \times m}$ and $\boldsymbol{b}_p \in \mathbb{R}^{|\Sigma|}$ are also two groups of learnable parameters, and $\boldsymbol{\mathcal{F}}_t$ is the final output of designed decoder.

4. Experiments

4.1. Datasets and Metrics

We evaluate our model on the dataset MS COCO [37] and dataset Flickr30k [38]. MS COCO is one of the largest datasets for image captioning, consisting of 123,287

images with five captions labeled for each. For validation and off-line comparison, we employ the widely-used "Karpathy" split [16], which contains 113,287 training images, 5000 for validation and 5000 for testing. Moreover, Flickr30k contains 31,783 images also with five captions for each. For both datasets, we truncate the captions longer than 16 words and convert all sentences into lower case, obtaining an experimental vocabulary with 9,487 words for COCO and 7,000 words for Flickr30k.

Following public convention, there are five classic metrics to be adopted for evaluating the performance of our model. They contain the BLEU [39], the METEOR [40], the ROUGE [41], the CIDEr [42] and the SPICE [43].

4.2. Implementation Details

We employ a pre-trained Faster R-CNN model with a backbone of ResNet-101 [4], to extract 36 features for each image. The dimension of feature vectors is 2048-D. Thereafter we project them into 512-D to reduce memory consumption. The hidden state size of LSTM is set to 1024. In addition, the dimensionality of the input to both the GSR and the self-attention module is set to 512. Moreover, the number of self-attention heads is 8. We set the number of layers to 3 in both the encoder and the decoder. In LSTM, the dropout rate is 0.5, and the dropout rate of all self-attention layers is set to 0.1. In the stage of cross-entropy training, we train our model using an initial learning rate of (5×10^{-4}) with a decay rate of 0.8 for every three epochs. In the CIDEr optimizing, we train our model in 30 epochs, by a learning rate of (2×10^{-5}) with a decay factor of 0.8 also for every three epochs. All compared models are trained with the Adam optimizer with batch size 50. In testing, we always use the same beam size of five for all.

4.3. Ablation Experiments

To quantify the performance of our proposed modules, we design a group of ablative experiments on MS COCO. We use a *vanilla Transformer* (see Figure 1) as our experiment '*base*', also similar to the self-attention network [24]. Its encoders do not consider the geometry information, and decoders are only combined with common sinuous position encodings.

11

Table 1: Ablation experiment comparisons. The results are reported after cross-entropy loss stage on the dataset "Karpathy" test split [16].

Model	BLEU-1	BLEU-4	METEOR	ROUGE	CIDEr	SPICE
Base	75.0	32.8	27.3	55.5	109.0	20.6
Base+GSR	76.9	35.6	28.1	57.0	115.1	21.4
Base+position-LSTM	76.5	34.5	28.0	56.8	114.9	21.3
Full: GAT	**77.5**	**37.8**	**28.5**	**57.6**	**119.8**	**21.8**

Table 2: The comparison results of different component combination strategies, including geometry Queries and Keys, and the GLU module in the Transformer structure.

Modules	Strategy	BLEU-1	BLEU-4	METEOR	ROUGE	CIDEr	SPICE
For Geometry Q & K	*add.*	76.0	35.1	27.2	56.0	116.4	20.7
	concat.	**77.5**	**37.8**	**28.5**	**57.6**	**119.8**	**21.8**
For GLU	without GLU	76.7	35.2	27.6	56.5	114.9	21.2
	GLU(enc.)	**77.7**	**37.8**	**28.4**	**57.4**	**119.4**	**21.8**
	GLU(enc. and dec.)	77.4	**37.8**	28.2	57.1	118.9	21.5

i) Effect of Geometry Self-attention

We first apply the GSR to the '*base*' model to evaluate its effect on encoders. GSR refines raw image representations by injecting explicit geometry relations. From Table 1, we could see that it obtains a huge improvement of CIDEr score (from 109.0 to 115.1) by applying our GSR. This comparison demonstrates that the base model without object geometry relations may be confused by irrelevant regions and misled by wrong information. Admittedly, our GSR furnishes the base model with a sense of '*where*'. It means that the model could look purposefully and thus generate the caption with precise order and geometry-aware words.

ii) Effect of Position LSTM

We further evaluate the impact of the position-LSTM module on captioning performance. We replace the simple sinuous position encoding in [24], with our LSTM-based ones. From the experimental comparisons in Table 1, we could see that our position-LSTM has raised the CIDEr score of the 'base' model by 5.9. Compared with sinuous encoding, our position-LSTM could provide the decoder with an expressive encoding for '*where*' to decode and also guide the decoder to capture correlated semantic information from image regions.

iii) Geometry Queries and Keys

To verify the efficiency of concatenating geometrical queries and keys with appearance queries and keys, we investigate different straggles to combine them. Mainly, we directly compare two operations, including '*add*' (adding) and '*concat*' (concatenating). Table 2 shows the comparison results of these two operations. We could figure out that the performance promotion by concatenating appearance queries and keys with geometry queries and keys is better than by adding them, although they both could surpass the base model in terms of most metrics. For example, the BLEU-1 score of the base model is only 75.0, while it could rise to 76.0 and 77.5 by '*add*' and '*concat*', respectively.

iv) Gated Linear Unit. In our architecture, GLUs play an important role in refining the output of original self-attention layers. Therefore we also have tested its effect on both encoders and decoders. Table 2 lists the comparison results. We could see that GLUs could always achieve a good performance promotion on the base model. However, we could also see that the GLU works best when it cooperates alone with encoders. If GLUs are assembled with encoders and decoders at the same time, the model could even get worse performance. For example, in Table 2, the SPICE by the base model without GLUs is 21.2, and it gets to 21.8 when GLUs are integrated only with encoders. However, the SPICE drops to 21.5 when GLUs are combined with all encoders and decoders. This finding is exactly consistent with [20]. It means that stacking too many GLUs on a group of position-LSTM decoder layers could damage the gradient of network training, depressing the capacity of self-attention layers.

4.4. Comparison with State-of-the-Art Models

We have also compared our GAT model with several state-of-the-art approaches of recent years. On dataset MS COCO, these comparison methods mainly include the SCST [3] optimizing evaluation metrics directly, the Up-Down model [4] with two-layer LSTM structure for extracting bottom-up features, and the ORT [27] employing a Transformer-like model with an object relation module. Besides, we also compare our model with the RF-Net [44] which employs features by multiple CNNs, the GCN-LSTM [45] exploiting pairwise relations by Graph Convolutional Network, the GAE [46] via auto-encoding scene graphs, the AoANet [20] refining self-attention

results also by GLUs. On Flickr30k, the models compared with ours include the Soft-Attention & Hard-Attention [29], the Deep VS [16], the NIC [18], the m-RNN [47], the adaptive model [25], the SEM architecture [48] and the DA framework [30].

Table 3: The off-line comparisons with some state-of-the-art methods on the '*Karpathy*' test splits of MS COCO. The higher, the better for all data in columns. The visual features of compared models are extracted by a same structure with the Faster RCNN [4]. Our model has reduced the dimensionality of embedding from 2048 to 512, due to the limitation of main memory capability.

Model	Cross-Entropy Loss						CIDEr-D Optimization					
	BLEU-1	BLEU-4	METEOR	ROUGE	CIDEr	SPICE	BLEU-1	BLEU-4	METEOR	ROUGE	CIDEr	SPICE
Single Model												
SCST [3]	-	30.0	25.9	53.4	99.4	-	-	34.2	26.7	55.7	114.0	-
Up-Down [4]	77.2	36.2	27.0	56.4	113.5	20.3	79.8	36.3	27.7	56.9	120.1	21.4
RFNet [44]	76.4	35.8	27.4	56.8	112.5	20.5	79.1	36.5	27.7	57.3	121.9	21.2
GCN-LSTM [45]	77.3	36.8	27.9	57.0	116.3	20.9	80.5	38.2	28.5	58.3	127.6	22.0
SGAE [46]	-	-	-	-	-	-	80.8	38.4	28.4	58.6	127.8	22.1
ORT [27]	-	-	-	-	-	-	80.5	38.6	28.7	58.4	128.3	22.6
AoANet [20]	77.4	37.2	28.4	57.5	**119.8**	21.3	80.2	38.9	29.0	58.8	129.8	22.4
GAT (ours)	**77.5**	**37.8**	**28.5**	**57.6**	119.8	**21.8**	**80.8**	**39.7**	**29.1**	**59.0**	**130.5**	**22.9**
Ensemble/Fusion												
SCST [3][Σ]	-	32.8	26.7	55.1	106.5	-	-	35.4	27.1	56.6	117.5	-
RFNet [44][Σ]	77.4	37.0	27.9	57.3	116.3	20.8	80.4	37.9	28.3	58.3	125.7	21.7
GCN-LSTM [45][Σ]	77.4	37.1	28.1	57.2	117.1	21.1	80.9	38.3	28.6	58.5	128.7	22.1
SGAE [46][Σ]	-	-	-	-	-	-	81.0	39.0	28.4	58.9	129.1	22.2
AoANet [20][Σ]	78.7	38.1	28.5	58.2	122.7	21.7	**81.6**	40.2	29.3	59.4	132.0	22.8
GAT (Ours)[Σ]	**79.0**	**38.8**	**28.7**	**58.9**	**123.7**	**22.1**	**81.6**	**40.7**	**29.4**	**59.6**	**133.4**	**23.2**

i) Off-line Evaluation

We also evaluate our model on the '*Karpathy*' test split of MS COCO, as in [16]. All models are firstly trained by cross-entropy loss and then optimized for the CIDEr score. For a fair comparison, the visual features fed to all models are directly extracted by a Faster R-CNN of the same structure. The top half of Table 3 (*i.e.*, Single Model) shows the performance comparisons of all models. By comparisons, it could be seen that our GAT outperforms all other methods on almost all metrics in terms of cross-entropy loss and CIDEr Optimization. Compared with the ORT [27], on CIDEr-D optimization, our GAT has obtained a BLEU-4 rise of 1.1%, METEOR rise of 0.4%, ROUGE rise of 0.6%, CIDEr rise of 2.2%, and SPICE rise of 0.3% by a single model. These rise values look obvious. Especially on BLEU-4, it achieves an increase from 38.6 to 39.7, and on CIDEr, it rises from 128.3 to 130.5. Similarly, our GAT also outperforms the AOANet [20], in terms of almost all metrics except the CIDEr of Cross-Entropy.

14

Moreover, in the down half of Table 3 (*i.e.*, Ensemble/Fusion), it shows the comparisons of 6 different captioning models. Every model is run four times respectively under four different initialization conditions, and then only the mean of four outputs is used for the comparisons of captioning models. We could also see that our GAT could surpass all other models on almost all metrics while optimized by Cross-Entropy Loss. Even on the metrics of CIDEr-D optimization, our BLEU-1 could almost keep the same as the AOANet [20], the best of all other models. On the rest metrics of CIDEr-D optimization, our GAT could always outperform all other models. For example, on the CIDEr of Cross-Entropy loss, our model achieves a result of 123.7, just 1.0 larger than the AOANet [20]. Moreover, the CIDEr score of CIDEr-D optimization of our model rises to 133.4, while the one of AOANet [20] is only 123.0, about 1.4 less than ours.

Table 4: The off-line comparison results with state-of-the-art methods on dataset Flickr30k. The higher, the better for the data in all data columns.

	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	CIDEr
Deep VS [16]	57.3	36.9	24.0	15.7	15.3	24.7
Soft-Attention [29]	66.7	43.4	28.8	19.1	18.5	-
Hard-Attention [29]	66.9	43.9	29.6	19.9	18.5	-
Google NIC [18]	66.4	42.3	27.7	18.3	-	-
m-RNN [47]	60.0	41.0	28.0	19.0	-	-
Adaptive [25]	67.6	49.4	35.4	25.1	20.4	53.1
SEM [48]	73.1	55.1	40.1	29.0	22.0	66.8
DA [30]	73.8	55.1	40.3	29.4	23.0	66.6
GAT (ours)	**74.4**	**56.7**	**41.8**	**30.8**	**23.4**	**68.0**

In Table 4, we compare the performance of our GAT with state-of-the-art models on dataset Flickr30k. It could be seen that our model outperforms all the compared methods by a large margin. For example, compared with Soft-Attention, which also employs an attention mechanism [29], our GAT obtains an improvement of 7.5 on BLEU-1, 10.9 on BLEU-4 and 4.9 on METEOR. Even compared with DA model[30], the best of the other eight models, our GAT still could often achieve a higher score than DA on all metrics. For example, our GAT could get an increase of about 0.6, 1.4 and 1.4 on BLEU-1, BLEU-4 and CIDEr, respectively. This comparison further demonstrates the superiority of our GAT method.

ii) On-line Evaluations

Table 5: The leaderboard of various methods on the online test server specially for COCO. Our method, GAT, went into the top 10 on the leaderboard when submitted online, on June 5th, 2021. The higher, the better for all listed data.

Model	BLEU-1		BLEU-2		BLEU-3		BLEU-4		METEOR		ROUGE		CIDEr	
	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40
SCST [3]	78.1	93.7	61.9	86.0	47.0	75.9	35.2	64.5	27.0	35.5	56.3	70.7	114.7	116.0
Up-Down [4]	80.2	95.2	64.1	88.8	49.1	79.4	36.9	68.5	27.6	36.7	57.1	72.4	117.9	120.5
RFNet [44]	80.4	95.0	64.9	89.3	50.1	80.1	38.0	69.2	28.2	37.2	58.2	73.1	122.9	125.1
GCN-LSTM [45]	-	-	65.5	89.3	50.8	80.3	38.7	69.7	28.5	37.6	58.5	73.4	125.3	126.5
SGAE [46]	81.0	**95.3**	65.6	89.5	50.7	80.4	38.5	69.7	28.2	37.2	58.6	73.6	123.8	126.5
AoANet [20]	81.0	95.0	65.8	89.6	51.4	81.3	39.4	71.2	**29.1**	**38.5**	58.9	**74.5**	126.9	129.6
NG-SAN [24]	80.8	95.0	65.4	89.3	50.8	80.6	38.8	70.2	29.0	38.4	58.7	74.0	126.3	128.6
GAT (Ours)	**81.1**	95.1	**66.1**	**89.7**	**51.8**	**81.5**	**39.9**	71.4	**29.1**	38.4	**59.1**	74.4	**127.8**	**129.8**

For a fairer comparison and getting to know the true position of our GAT in the image captioning task, we also submitted it to the official MS COCO test server. The last line in Table 5 lists the test results of our GAT model. In this test, an ensemble of 4 different outputs of our model trained by the '*Karpathy*' split is used for comparing with others. From Table 5, we could see that our GAT obtains the best captioning performance on each c5 of almost all metrics, and it's only a bit lower than others on the c40 of three metrics. For example, it's 127.8 on CIDEr(c5), almost 0.9 higher than AoANet [20], the best of the others. Even on BLEU-1(C40) and ROUGE(c40), our GAT is only paltry 0.2 lower than the SGAE [46] and 0.1 lower than the AoANet [20]. Therefore, on the whole, our GAT model could be synthetically deemed to be more superior than other compared ones.

4.5. Caption Text Comparisons

To qualitatively illustrate the superiority of our method, we present six images with their captions generated respectively by our GAT and the *Vanilla Transformer* model (*i.e.*, base), as well as three manual captions of ground truth (*i.e.*, GT1∼ GT3) in Figure 3. It can be seen that GAT could generate an accurate image caption with complete semantic structure and clear position relations. For example, the caption by GAT could geometrically describe that "people" are "*under* umbrellas" and "in *front* of store" in the top-left sub-image of Figure 3, and "dog laying *under* the chair" rather

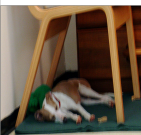than "laying *on* the chair' in the middle-right sub-image of Figure 3.
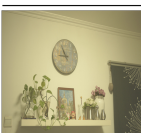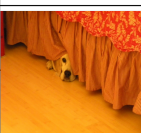
GAT: A group of people sitting under umbrellas in front of a restaurant.
Base: A group of people sitting at tables outside of a restaurant.
GT1: A group of people eating at tables under umbrellas in front of a restaurant.
GT2: Many peoplea re sitting outside of a local restaurant.
GT3: Some people are sitting outside in chairs with umbrellas.

GAT: Two people laying on the beach under an umbrella.
Base: A woman laying on the beach with a red umbrella.
GT1: Two people on the beach laying under a red umbrella.
GT2: A woman in a bikini laying under a red umbrella.
GT3: The woman lies next to a man on the beach under an umbrella.

GAT: A black train on the tracks under a bridge.
Base: A black train is sitting on the tracks.
GT1: A black train is driving under a bridge.
GT2: A black train going down the train track.
GT3: A black train passes under a small bridge.

GAT: A dog laying on the floor under a wooden chair.
Base: A dog laying on the floor next to a turns one leg.
GT1: A dog that is laying under a chair.
GT2: A close up of a dog sleeping under a chair.
GT3: A blurry photo of a dog sleeping underneath a chair.

GAT: A clock on a wall above a bunch of flowers.
Base: A wall with a clock and a vase on it.
GT1: There is a clock above a shelf of knickknacks
GT2: There is a wall clock above the mantel piece.
GT3: A shelf on the wall with plants and decor under a clock.

GAT: A white dog laying on the floor under a bed.
Base: A dog laying on the floor next to a curtains.
GT1: A dog that is laying under a bed.
GT2: A dog laying under a bed on top of a wooden floor.
GT3: A white dog under a bed with its eye wide open.

Figure 3: The output comparison of captions generated respectively by our GAT and a base model, as well as three manual description captions of ground truth, GT1~GT3.

The novel characteristic that our GAT is able to accurately describe the geometry & position relations mainly owes to the GSR module, which can explicitly incorporate the spatial correlations of image regions into object feature representations. Moreover, the position-LSTM could also keep reminding the decoder, at every decoding step, which visual object should be attended to. On the whole, our GAT could often express almost all exact objects contained in a given image, such as "*people*", "*umbrellas*" and "*restaurant*", etc. in the top-left sub-image of 3. Furthermore, GAT could also capture the geometry relations between any two detected objects and then decide whether to specify the correlations of detected objects, benefited by which it could often generate the highly-reliable image captions with spatial-aware semantic.

5. Conclusions

In this paper, we propose the Geometry Attention Transformer (GAT), an extension scheme to the well-known Transformer for image captioning in recent years. It is able to explicitly refine image representations by incorporating the geometry features of visual objects into region encodings. Moreover, the position-LSTM in decoder layers

17

could often fulfill a precise encoding for the word order of caption texts. The ablation experiments on the base model show that the GSR for capturing geometry features and the position-LSTM for injecting position encodings could be effective. Each of them, if cooperating with a base model, could promote image captioning performance. In addition, the experimental comparisons (off-line & on-line) also show that our GAT framework could often outperform state-of-the-art ones on the datasets MS COCO and Flickr30k, respectively.

Acknowledgement

References

[1] A. Farhadi, M. Hejrati, M. A. Sadeghi, P. Young, C. Rashtchian, J. Hockenmaier, D. Forsyth, Every picture tells a story: Generating sentences from images, in: European conference on computer vision, Springer, 2010, pp. 15–29.

[2] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, et al., Visual genome: Connecting language and vision using crowdsourced dense image annotations, International Journal of Computer Vision 123 (1) (2017) 32–73.

[3] S. J. Rennie, E. Marcheret, Y. Mroueh, J. Ross, V. Goel, Self-critical sequence training for image captioning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 7008–7024.

[4] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, L. Zhang, Bottom-up and top-down attention for image captioning and visual question answering, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

[5] M. Cornia, M. Stefanini, L. Baraldi, R. Cucchiara, Meshed-memory transformer for image captioning, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 10575–10584. `doi:10.1109/CVPR42600.2020.01059`.

[6] J. Zhang, K. Li, Z. Wang, X. Zhao, Z. Wang, Visual enhanced glstm for image captioning, Expert Systems with Applications 184 (2021) 115462. `doi:https://doi.org/10.1016/j.eswa.2021.115462`.

[7] S. Bai, S. An, A survey on automatic image caption generation, Neurocomputing 311 (2018) 291–304.

[8] V. Ordonez, G. Kulkarni, T. L. Berg, Im2text: Describing images using 1 million captioned photographs, in: Proceedings of the Advances in Neural Informa- tion Processing Systems (NIPS), 2011, pp. 1143–1151.

[9] A. Gupta, Y. Verma, C. V. Jawahar, Choosing linguistics over vision to describe images, in: In Twenty-Sixth National Conference on Artificial Intelligence, 2012, pp. 606–612.

[10] R. Socher, L. Fei-Fei, Connecting modalities: Semi-supervised segmentation and annotation of images using un-aligned text corpora, in: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE, 2010, pp. 966–973.

[11] G. Kulkarni, V. Premraj, V. Ordonez, S. Dhar, S. Li, Y. Choi, A. C. Berg, T. L. Berg, Babytalk: Understanding and generating simple image descriptions, IEEE Transactions on Pattern Analysis and Machine Intelligence 35 (12) (2013) 2891–2903. doi:10.1109/TPAMI.2012.162.

[12] Y. Ushiku, M. Yamaguchi, Y. Mukuta, T. Harada, Common subspace for model and similarity: Phrase learning for caption generation from images, in: Proceedings of 2015 IEEE International Conference on Computer Vision (ICCV), 2015, pp. 2668–2676. doi:10.1109/ICCV.2015.306.

[13] A. Karpathy, A. Joulin, F. F. Li, Deep fragment embeddings for bidirectional image sentence mapping, in: Proceedings of the Twenty Seventh Advances in Neural Information Processing Systems (NIPS), Vol. 3, 2014, pp. 1889–1897.

[14] L. Ma, Z. Lu, L. Shang, H. Li, Multimodal convolutional neural networks for matching image and sentence, in: Proceedings of IEEE International Conference on Computer Vision, 2015, pp. 2623–2631.

[15] F. Yan, K. Mikolajczyk, Deep correlation for matching images and text, in: Proceedings of 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 3441–3450. doi:10.1109/CVPR.2015.7298966.

[16] A. Karpathy, L. Fei-Fei, Deep visual-semantic alignments for generating image descriptions, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 3128–3137.

[17] A. Karpathy, L. Fei-Fei, Deep visual-semantic alignments for generating image descriptions, IEEE Transactions on Pattern Analysis and Machine Intelligence 39 (4) (2017) 664–676. doi:10.1109/TPAMI.2016.2598339.

[18] O. Vinyals, A. Toshev, S. Bengio, D. Erhan, Show and tell: A neural image caption generator, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 3156–3164.

[19] Q. You, H. Jin, Z. Wang, C. Fang, J. Luo, Image captioning with semantic attention, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 4651–4659. doi:10.1109/CVPR.2016.503.

[20] L. Huang, W. Wang, J. Chen, X.-Y. Wei, Attention on attention for image captioning, in: Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 4634–4643.

[21] R. M. Oruganti, S. Sah, S. Pillai, R. Ptucha, Image description through fusion based recurrent multi-modal learning, in: 2016 IEEE International Conference on Image Processing (ICIP), 2016, pp. 3613–3617. `doi:10.1109/ICIP.2016.7533033`.

[22] J. Mao, X. Wei, Y. Yang, J. Wang, Z. Huang, A. L. Yuille, Learning like a child: Fast novel visual concept learning from sentence descriptions of images, in: 2015 IEEE International Conference on Computer Vision (ICCV), 2015, pp. 2533–2541. `doi:10.1109/ICCV.2015.291`.

[23] M. Nabati, A. Behrad, Multimodal video-text matching using a deep bifurcation network and joint embedding of visual and textual features, Expert Systems with Applications 184 (2021) 115541, online. `doi:https://doi.org/10.1016/j.eswa.2021.115541`.

[24] L. Guo, J. Liu, X. Zhu, P. Yao, S. Lu, H. Lu, Normalized and geometry-aware self-attention network for image captioning, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 10324–10333. `doi:10.1109/CVPR42600.2020.01034`.

[25] J. Lu, C. Xiong, D. Parikh, R. Socher, Knowing when to look: Adaptive attention via a visual sentinel for image captioning, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[26] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in neural information processing systems, 2017, pp. 5998–6008.

[27] S. Herdade, A. Kappeler, K. Boakye, J. Soares, Image captioning: Transforming objects into words, in: Advances in Neural Information Processing Systems, 2019, pp. 11135–11145.

[28] G. Li, L. Zhu, P. Liu, Y. Yang, Entangled transformer for image captioning, in: Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 8928–8937.

[29] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, Y. Bengio, Show, attend and tell: Neural image caption generation with visual attention, in: International conference on machine learning, 2015, pp. 2048–2057.

[30] L. Gao, K. Fan, J. Song, X. Liu, X. Xu, H. T. Shen, Deliberate attention networks for image captioning, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33, 2019, pp. 8320–8327.

[31] M. Cornia, M. Stefanini, L. Baraldi, R. Cucchiara, Meshed-memory transformer for image captioning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 10578–10587.

[32] L. Zhou, H. Palangi, L. Zhang, H. Hu, J. J. Corso, J. Gao, Unified vision-language pre-training for image captioning and vqa., in: AAAI, 2020, pp. 13041–13049.

[33] H. Chen, Y. Wang, X. Yang, J. Li, Captioning transformer with scene graph guiding, in: 2021 IEEE International Conference on Image Processing (ICIP), 2021, pp. 2538–2542. `doi:10.1109/ICIP42928.2021.9506193`.

[34] X. Zhu, L. Li, J. Liu, H. Peng, X. Niu, Captioning transformer with stacked attention modules, Applied Sciences 8 (5) (2018) 739.

[35] L. Yu, J. Zhang, Q. Wu, Dual attention on pyramid feature maps for image captioning, IEEE Transactions on Multimedia (2021) 1–1`doi:10.1109/TMM.2021.3072479`.

[36] S. Liu, Z. Ren, J. Yuan, Sibnet: Sibling convolutional encoder for video captioning, IEEE Transactions on Pattern Analysis and Machine Intelligence 43 (9) (2021) 3259–3272. `doi:10.1109/TPAMI.2019.2940007`.

[37] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, Microsoft coco: Common objects in context, in: European conference on computer vision, Springer, 2014, pp. 740–755.

[38] P. Young, A. Lai, M. Hodosh, J. Hockenmaier, From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions, Transactions of the Association for Computational Linguistics 2 (2014) 67–78.

[39] K. Papineni, S. Roukos, T. Ward, W.-J. Zhu, Bleu: a method for automatic evaluation of machine translation, in: Proceedings of the 40th annual meeting on association for computational linguistics, Association for Computational Linguistics, 2002, pp. 311–318.

[40] S. Banerjee, A. Lavie, Meteor: An automatic metric for mt evaluation with improved correlation with human judgments, in: Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization, 2005, pp. 65–72.

[41] C.-Y. Lin, ROUGE: A package for automatic evaluation of summaries, in: Text Summarization Branches Out, Association for Computational Linguistics, Barcelona, Spain, 2004, pp. 74–81.

[42] R. Vedantam, C. Lawrence Zitnick, D. Parikh, Cider: Consensus-based image description evaluation, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 4566–4575.

[43] P. Anderson, B. Fernando, M. Johnson, S. Gould, Spice: Semantic propositional image caption evaluation, in: ECCV, 2016.

[44] W. Jiang, L. Ma, Y.-G. Jiang, W. Liu, T. Zhang, Recurrent fusion network for image captioning, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 499–515.

[45] T. Yao, Y. Pan, Y. Li, T. Mei, Exploring visual relationship for image captioning, in: Proceedings of the European conference on computer vision (ECCV), 2018, pp. 684–699.

[46] X. Yang, K. Tang, H. Zhang, J. Cai, Auto-encoding scene graphs for image captioning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 10685–10694.

[47] J. Mao, W. Xu, Y. Yang, J. Wang, Z. Huang, A. Yuille, Deep captioning with multimodal recurrent neural networks (m-rnn), arXiv preprint arXiv:1412.6632.

[48] W. Cai, Q. Liu, Image captioning with semantic-enhanced features and extremely hard negative examples, Neurocomputing 413 (2020) 31–40. doi:https://doi.org/10.1016/j.neucom.2020.06.112.