Hindawi
Computational Intelligence and Neuroscience
Volume 2023, Article ID 8600853, 16 pages
https://doi.org/10.1155/2023/8600853

Research Article

A Survey on Learning Objects' Relationship for Image Captioning

Du Runyan, [1,2,3,4] **Zhang Wenkai**, [1,2,3,4] **Guo Zhi**, [1,2,3,4] **and Sun Xian** [1,2,3,4]

[1] *Aerospace Information Research Institute, Chinese Academy Sciences, Beijing, China*
[2] *University of Chinese Academy of Sciences, Beijing, China*
[3] *School of Electronic, Electrical and Communication Engineering, University of Chinese Academy of Sciences, Beijing, China*
[4] *Key Laboratory of Network Information System Technology, Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing, China*

Correspondence should be addressed to Zhang Wenkai; zhangwk@aircas.ac.cn

Received 19 August 2022; Revised 4 February 2023; Accepted 6 February 2023; Published 29 May 2023

Academic Editor: Tapan Senapati

Image captioning is a challenging modality transformation task in computer vision and natural language processing, aiming to understand the image content and describe it with a natural language. Recently, the relationship information between objects in the image has been investigated to be of importance in generating a more vivid and readable sentence. Many types of research have been done in relationship mining and learning for leveraging into the caption models. This paper mainly summarizes the methods of relational representation and relational encoding in image captioning. Besides, we discuss the advantages and disadvantages of these methods and provide commonly used datasets for the relational captioning task. Finally, the current problems and challenges in this task are highlighted.

1. Introduction

Image captioning[1–30] is to understand the content of an image and further inference a natural sentence to describe it. The generated description needs to achieve satisfactory accuracy, adequacy, and readability [9, 31–33]. Readability requires the sentences to satisfy grammatical rules, the accuracy makes the content of generated sentences conform to the content of images, and the adequacy measures the adequacy of the generated sentences to express the image information. The adequacy and accuracy of the sentence include whether the visual vocabulary (describing the category and attributes of the object) and the relational vocabulary (describing the relationship between the objects) are fully reflected and whether they conform to the image's content.

The early captioning methods theoretically use image-to-text retrieval [1, 34] or filling sentence templates [35–37] to improve the adequacy and accuracy of the generated sentences. In technical, they mainly use the static object categories and the statistical language model. In technical, they mainly use the static object categories and the statistical language model. About retrieval methods, Aker and Gaizauskas [34] used a dependency model to summarize the information contained in multiple web documents and localize this information to images. Kulkarni et al. [1] used conditional random fields based on the objects detected in the image to predict the image's label for retrieval. About templates' methods, Li et al. [35] proposed a network-scale-basedn-gram method to collect candidate phrases and other form sentences. Yang et al. [36] proposed a language model trained on the English Gigaword corpus to obtain the action in the image and incorporated them into a hidden Markov model. Lin et al. [37] used a 3D visual analysis system to represent objects, attributes, and relationships in images. They transformed them into a series of semantic trees, from which they learned grammar and generated sentences.

However, the early captioning methods [1, 34–37] are suffered from few shortcomings. The template-based methods would make the generated sentences rigid and lack readability. At the same time, the retrieval would lead to mismatches between images and texts, affecting accuracy or adequacy. With the development of the deep learning technology [38–49], Vinyals et al. [2] proposed an encoder-decoder model, which uses convolutional neural networks [40] to understand objects and scenes in images, and uses LSTM [44, 50] to model the long-term dependency between words. Specifically, the generation of individual words in a sentence depends on the memory state and the image's global information. Xu et al. [3] incorporated an attention mechanism with the encoder-decoder framework to align text to specific regions in an image. Lu et al. [4] proposed an adaptive attention method that utilizes visual sentinels to align nonvisual vocabulary during sentence generation. In the related multimodal field [51–57], Ding et al. [58] introduced the attention mechanism to the video captioning, so that the model can adaptively focus on the elements, parts, or details in the image when dealing with each frame. Qin et al. [59] considered the visual coherence of the attention region and introduced the memory ability in the attention mechanism. For alleviating the accumulated error on sentence generation, they proposed a new language model which generates sentence chunks by chunks instead of words-by-words.

Furthermore, to more accurately align objects with words, Anderson et al. [5] adopted an object detection network to detect objects and constructed a two-LSTMs' decoder to learn the dependencies between words in sentences and the alignment between words and image regions. For enhancing the vocabulary coherence between words and syntactic paradigm of sentences, Ke et al. [60] proposed a new LSTM variant which considered the previous generated words and their relative positional information during decoding. This perception can also bring great improvement when integrating it with the image captioning models. Ding et al. [61] were inspired by the perception of the human brain and adjust the attention weight of each object according to its own color, area of bound box, and visual permutations.

In recent years, with the development of full-attentive models [9, 14, 18, 62, 63], Vaswani et al. [64] proposed the Transformer to use attention to learn interactions of intermodality and intramodality. They obtained excellent achievements in natural language processing, such as machine translation. Zhu et al. [6] applied transformer to image captioning and confirmed the effectiveness of the transformer in the captioning task. The transformer learns the interrelationships between object attribute features in visual sequences through the encoder and utilizes attention in the decoder to align text features with visual features. Under the object features [38, 65] provided by the pretrained object detection network [38, 43, 65], the accuracy and adequacy of the visual vocabulary generation are significantly improved with the reinforcement learning strategy [12, 66]. On the other hand, BERT-based vision-language pretraining methods [67, 68] concentrate on designing a unified framework for multiple vision-language tasks, which first optimize the object's features by specific pretraining objectives and then generating sentence after finetuning the features with the caption objective. Those methods have achieved a new higher-level performance in image captioning. Furthermore, Li et al. [69] have designed a decoupled encoder-decoder framework with a scheduled sampling strategy for countering the incompatibility between VL understanding and caption generation. Recently, Li et al. [70] have used the cross-modal retrieval technique to generate a primary sentence and refine its content with the transformer blocks, which extremely improved the model performance in the end-to-end training mode. In order to have a better caption development, a unified codebase [71] has been proposed which covered many high performance modules in each stage of the cross-modal analytics between vision and language in the multimedia field.

Since 2019, some studies [62, 63, 72–74] have begun to focus on characterizing the relationship between objects based on the abovementioned works to improve the generation of relational vocabulary. For modeling the objects' relationships, researchers first start from the basic spatial relationship to explicitly perceive relational information and establish alignment with relational words. Then, they take a far more step to mine the higher-level semantic relationships hiding in the image. In this process, low-level geometric spatial features are less difficult to be constructed, but the constructed features are also less capable of representing complex relationship categories in textual modality. The relationship between objects can be reflected by multiple relationship categories with similar meanings, which belong to multirelational data. In the case of multirelational data in images, finding higher-level relational features is a difficult challenge. After feature construction, how to effectively combine relational features in the feature optimization stage so that the optimized features can have good separability for different relational categories is a problem worth studying. In order to follow up the development of relational image captioning, it is necessary to overview the previous works about relationships and assist the following researchers in improving the intelligence of captioning models. This paper mainly classifies and summarizes the extraction methods of this relational information and their corresponding encoding methods in the current image captioning. According to the frame shown in Figure 1, we overview the main line of relational captioning and summarize a taxonomy of relational methods. Meanwhile, the commonly used datasets and evaluation measures are available in this paper. The advantages and disadvantages of methods and future development prospects are analyzed.

1.1. Contributions. Our contributions in this paper are shown as follows:

(1) Combining all previous studies in relational image captioning, we summarize a taxonomy of relational information processing in the image, which includes feature construction and encoding. Meanwhile, we introduce the corresponding methods and analyze their strength and weakness.

(2) We review the relevant datasets involved in the relational image captioning, covering relational understanding and image captioning datasets. The metrics used in evaluation are also recorded in this paper.

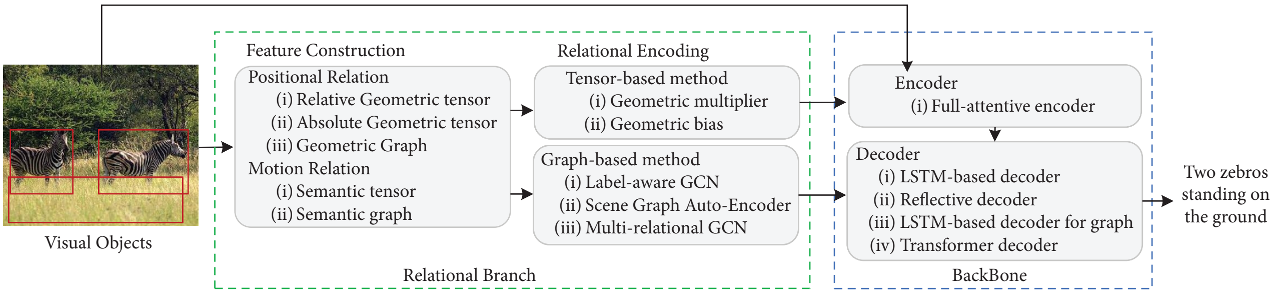


Figure 1: The taxonomy of the visual relationship.

(3) We observe and analyze the development of the relational image caption and enumerate the main challenges in this area and future development directions.

This paper is organized as follows: the second section briefly introduces the content of the visual branch in relational captioning, mainly about the basic knowledge and overall framework commonly used in the relational image description. The third section explicitly describes the construction of relational features in images. The fourth section mainly describes the encoding of relational information. The fifth section mainly describes the datasets and related evaluation indicators used to extract and learn relational data in image captioning. The sixth section concludes and presents the prospect of future development in this field.

2. Backbone

The backbone of relational captioning is the standard encoder-decoder framework [2–4] as the common captioning task. It is irrelevant to the relationship but is necessary to discuss for constructing the whole procedure. As shown in Figure 1, the backbone consists of two parts: encoder and decoder. Given an image I, relational captioning begins with objects detected from the object detector [38]. The encoder refines each element in the visual sequence and further feed it into the decoder for generating a natural sentence.

2.1. Encoder

2.1.1. Full-Attentive Encoder. Initializing from the visual sequence $\mathscr{V} = \{v_1, v_2, \ldots, v_n\}$, the purpose of the encoder is to enrich each object's feature. Recently, transformer-dominated full-attentive models [2] play an important role in relational captioning. The most important component in transformer is the scaled dot-product attention operator, whose structure is shown in Figure 2(a). Its calculation formula is shown as follows:

$$\operatorname{Att}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \operatorname{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^{\mathbf{T}}}{\sqrt{d}}\right)\mathbf{V}. \tag{1}$$

It calculates the similarity of each query vector $q \in R^d$ in the query matrix $\mathbf{Q} \in R^{N\times d}$ and each key vector in the key matrix $\mathbf{k} \in R^d$. The generated attention weight $E = \mathbf{Q}\mathbf{K}^{\mathbf{T}}$. E is multiplied with $\mathbf{V}$ so that each output vector comes from a weighted sum of each element in $\mathbf{V}$ and its corresponding weight in the weight matrix. Meanwhile, to further enhance the model representation ability of the attention operator [64] and speed up the convergence of the model during the training process, the multihead attention mechanism [64] is combined with the conventional attention operator, as shown in (b) in Figure 2. Its formula is calculated as follows:

$$\operatorname{MAtt}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \underset{i=1:h}{\operatorname{Concat}}\left(\operatorname{Att}\left(\mathbf{Q}_{\mathbf{i}}, \mathbf{K}_{\mathbf{i}}, \mathbf{V}_{\mathbf{i}}\right)\right). \tag{2}$$

i is the index of each head. Each head is a segmentation of the original feature space. The dimension of each subspace is d/h, where h is the number of total heads. The multihead attention mechanism performs self-attention calculations in each subspace and further fuse all outputs from each subspace with Concat. After passing through the encoder, the optimized sequence of object features is fed into a subsequent decoder to generate sentences.

2.2. Decoder

2.2.1. LSTM-Based Decoder. Decoders for relational captioning are various language models, commonly using LSTM [44], transformer, and their variants. We denote the output of the encoder as $\mathscr{X} = \{x_1, x_2, \ldots, x_n\}$. Given $\mathscr{X}$, Anderson et al. [5] build a decoder with two LSTMs, which contain an attention LSTM and a language LSTM, respectively. The attention LSTM takes the word embedding vector w_{t-1} and the hidden layer state of the language LSTM h_{t-1}^l at the last moment and the global visual feature (average of all object features) $\overline{g}$ as the input to calculate the current moment's hidden layer state h_t^a.

$$\begin{aligned} h_t^a &= LSTM\left(\left[\overline{g}; w_{t-1}; h_{t-1}^l\right], h_{t-1}^a; \theta^a\right),\\ \tilde{\alpha}_{t,i}^c &= w_c^T \tanh\left(W_{xc} x_{t,i} + W_{hc} h_t^a\right),\\ \boldsymbol{\alpha}_{\mathbf{t}}^{\mathbf{c}} &= \operatorname{softmax}\left(\tilde{\boldsymbol{\alpha}}_{\mathbf{t}}^{\mathbf{c}}\right). \end{aligned} \tag{3}$$

As an attention query, h_t^a computes the attention score $\tilde{\alpha}_{t,i}^c$ with each element of $\mathscr{X}$. The $\boldsymbol{\alpha}_{\mathbf{t}}^{\mathbf{c}}$ is the context attention weight for fusing $\mathscr{X}$ into a context vector. The language LSTM takes the current hidden state h_t^a of attention LSTM and the context vector to generate the current word representation w_t.

2.2.2. Reflective Decoder. In the word-by-word decoding process, modeling the previous content and the positional information of each word is beneficial for generating words in the current time step. Ke et al. [60] enhance the LSTM-based decoder with reflective attention and reflective position modules. In the LSTM-based decoder, the output of language LSTM h_t^l is followed by a linear function for generating the current word. In the reflective attention module, it replaces h_t^l with an attended result $\widehat{h}_t^l$ reasoned by the previous generated content.

$$\begin{aligned} \alpha_{i,t}^{\mathrm{ref}} &= \mathbf{W}_h^l \tanh\left(\mathbf{W}_{h_2}^l h_i^l + \mathbf{W}_{h_1}^l h_t^a\right), \\ \alpha_t^{\mathrm{ref}} &= \mathrm{softmax}\left(a_t^{\mathrm{ref}}\right), a_t^{\mathrm{ref}} = \alpha_{i,t\ i=1}^{\mathrm{ref}\,t}, \end{aligned} \tag{4}$$

where $\alpha_{i,t}^{\mathrm{ref}}$ is the attention weight corresponding to each h_i^l in i-th time step. Besides, $\widehat{h}_t^l$ is constrained by the relative position of each word in the sentence with a loss function which minimizes the distance between $\widehat{h}_t^l$ and t/n, where t is the time step of each word and n is the length of the sentence.

2.2.3. LSTM-Based Decoder for Graph. For introducing the graph structure into the language decoder, Chen et al. [74] proposed a variant of a conventional two-LSTMs decoder which consists of two modules: graph-based attention mechanism and graph update mechanism. The graph-based attention mechanism computes two attention weights: $\alpha_{\mathbf{t}}^{\mathbf{c}}$ and $\alpha_{\mathbf{t}}^{\mathbf{f}}$. $\alpha_{\mathbf{t}}^{\mathbf{c}}$ is the context attention weight which follows the two-LSTMs decoder. $\alpha_{\mathbf{t}}^{\mathbf{f}}$ is the flow attention weight which constrains the model to attend the semantically relevant node within the neighbors of the previous attended one. Specifically, it is a soft interpolation of the three flow scores with a dynamic gate. According to the different moving steps, the three flow scores are computed with the adjacency matrix $\mathbf{M_f}$: (1) stay at the same node $\alpha_{\mathbf{t,0}}^{\mathbf{f}} = \alpha_{\mathbf{t-1}}$, (2) move one step $\alpha_{\mathbf{t,0}}^{\mathbf{f}} = \mathbf{M_f}\alpha_{\mathbf{t-1}}$, and (3) move two steps $\alpha_{\mathbf{t,2}}^{\mathbf{f}} = (\mathbf{M_f})^2\alpha_{\mathbf{t-1}}$. The flow attention is computed as follows:

$$\begin{aligned} \mathbf{s_t} &= \mathrm{softmax}\left(W_s\sigma\left(W_{sh}h_t^a + W_{sz}z_{t-1}\right)\right), \\ \boldsymbol{\alpha}_{\mathbf{t}}^{\mathbf{f}} &= \sum_{k=0}^{2} \alpha_{t,k}^{f}, \\ \beta_t &= \mathrm{sigmoid}\left(w_g\sigma\left(W_{gh}h_t^a + W_{gz}z_{t-1}\right)\right), \\ \boldsymbol{\alpha}_{\mathbf{t}} &= \beta_t\boldsymbol{\alpha}_{\mathbf{t}}^{\mathbf{c}} + (1-\beta_t)\boldsymbol{\alpha}_{\mathbf{t}}^{\mathbf{f}}. \end{aligned} \tag{5}$$

The final attention weight $\alpha_{\mathbf{t}}$ takes a balance between $\alpha_{\mathbf{t}}^{\mathbf{c}}$ and $\alpha_{\mathbf{t}}^{\mathbf{f}}$ with a gate function. To avoid repetition and omission in the attention process, Chen el al. [74] use a graph update mechanism to dynamically remove or preserve some nodes with a visual sentinel $\mathbf{u_t}$.

$$\mathbf{u_t} = \mathrm{sigmoid}\left(f_{vs}\left(h_t^l;\theta_{vs}\right)\right)\boldsymbol{\alpha}_{\mathbf{t}}. \tag{6}$$

The scalar $\mathbf{u_{t,i}}$ indicates whether the generated word expresses the attended node. For avoiding repetition, an erase gate for the i-th node $e_{t,i}$ is computed according to its visual sentinel $u_{t,i}$. Meanwhile, if a node needs multiple access, an add gate for the i-th node $a_{t,i}$ is also computed to preserve its status.

$$\begin{aligned} e_{t,i} &= \mathrm{sigmoid}\left(f_{ers}\left(\left[h_t^l; x_{t,i}\right]; \theta_{ers}\right)\right), \\ \widehat{x}_{t+1,i} &= x_{t,i}\left(1 - u_{t,i}e_{t,i}\right), \\ a_{t,i} &= \sigma\left(f_{\mathrm{add}}\left(h_t^l; x_{t,i}; \theta_{\mathrm{add}}\right)\right), \\ x_{t+1,i} &= \widehat{x}_{t+1,i} + u_{t,i}a_{t,i}, \end{aligned} \tag{7}$$

where f_* are fully connected networks and θ_*, W_*, and w_* are the learnable parameters.

2.2.4. Transformer Decoder. The transformer decoder proposed by Vaswani et al. [64] is also widely used in image captioning, which consists of multiple sublayers. The textual features in each sublayer first learn the interaction within its modality through self-attention, then align specific object features through the cross attention between the textual features and $\mathscr{X}$. They finally pass the fully connected layer to generate the representation w_t of the word at the current moment. w_t finally generates the corresponding word through the mapping matrix and the softmax function.

In summary, relational image description's overall process is generating sentences through the visual branch. At the same time, the relational branch processes the object-level relational features to be integrated into the visual branch. In the vision branch, given an image I, the object feature sequence $\mathscr{V}$ obtained by target detection is used as input, and then $\mathscr{X}$ is obtained by encoder learning. The commonly used models in encoders are mainly transformer encoders or graph convolutional networks [72–76]. Then, V_e is input to the transformer decoder or double LSTM to generate natural sentences word-by-word.

3. Relational Branch

The relational branch is the core of relational captioning. It concentrates on the encoder part and incorporates the relationship between objects into the encoder. It includes two steps:(1) feature construction and (2) relational encoding. The relationships in image can be divided into two categories: (1) position relationships and (2) action relationships, corresponding to the positional words and predicate words. As shown in Figure 3, the position relationship refers to the geometric relationship between the objects, which can be expressed as positional words in sentences, such as "in" and" on." On the other side, the action relationship represents more complicated and higher-level semantic relationship between the subject and the object. In textual modality, a predicate generally represents one kind of action relationship, As shown in Figure 3. This section mainly introduces different relational feature construction methods and feature encoding methods according to the different types of relations.

3.1. Feature Construction. The first step in relational captioning is extracting and constructing relational features. Many studies have explored the relationship between objects in images in visual relationship detection and scene understanding. The position relationship represents the up-

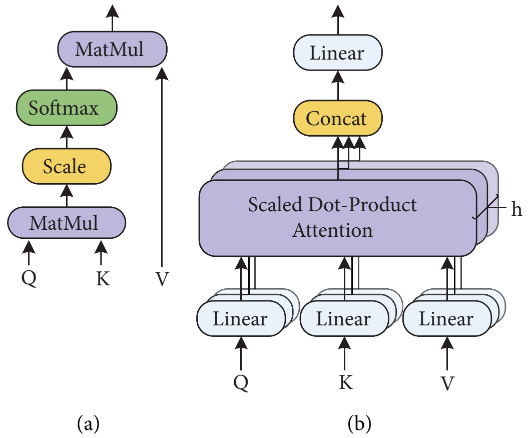


Figure 2: The scaled dot-product attention and multihead attention.

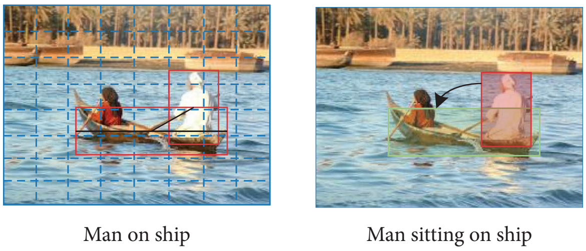


Figure 3: The example for illustrating the positional relation and motion relation.

down, left-right relationship between two objects in the 2-dimensional space. It corresponds to the words describing the position in the sentence, such as “on” and “near”. The action relationship between objects represents a specific action, which is corresponding to a particular predicate verb in the generated sentence. Figure 3 defines the above-mentioned two kinds of relationships. In this section, we mainly summarize the current extraction methods of these two kinds of relational information and list the advantages and disadvantages of each technique.

3.1.1. Positional Relationship. The positional relationship between objects is usually represented by the geometric relationship between two objects’ bounding boxes in two-dimensional space. Given an image I and N object boxes in it, the position vector of each object box is represented as (x_i, y_i, w_i, h_i), and the geometric relationship between the object boxes includes the relative distance, relative angle, and relative area between the object boxes. According to the different data structures, the representation methods can be divided into two types: (1) tensor and (2) graph.

3.1.2. Relative Geometric Tensor. The main idea is to construct a $N \times N \times d$ tensor to represent all $N \times N$ object pairs. Each of these relations is a d-dimensional vector. Herdade et al. [62] and Guo et al. [63] used the relative distances of the box’s center and relative size ratios between objects’ boxes to construct geometric vectors:

$$\left(\log\left(\frac{\left|x_j - x_i\right|}{w_i}\right), \log\left(\frac{\left|y_j - y_i\right|}{h_i}\right), \log\left(\frac{w_j}{w_i}\right), \log\left(\frac{h_j}{h_i}\right)\right). \tag{8}$$

The subscripts i and j represent the image’s i-th and j-th objects. The external logarithmic function plays a numerically stable role in ensuring that when the width and height of the object box i are very small. The output value will not be too far away from the mean value, resulting in excessive variance and making the model difficult to converge. All the $N \times N$ object pairs’ geometric vectors form the $N \times N \times 4$ geometric tensor. Meanwhile, the activation ReLU filters the negative elements when two objects’ boxes are very close.

In summary, the geometric feature mainly describes the relative distance between the center points of the two object boxes and the relative size ratio between the object boxes. It can provide basic prior information about the object’s size and location, which is very helpful for image understanding. However, the geometric features extracted by this method are not enough to represent high-level semantic relationship categories, and they are also interfered by the scale information of the bounding box when representing different spatial orientations, that is, the amount of relationship that needs to be calculated is large, and all object pairs in the image need to be considered in the calculation process. In practical use, if a complex network model is constructed to learn geometric feature tensors, it often brings a lot of computational costs. To a certain extent, the learning ability of the model for the position relationship information between objects is limited.

3.1.3. Absolute Geometric Tensor. The absolute geometric tensor directly maps the coordinates of the object frame in the image to the feature space. Luo et al. [77] designed a transformer variant for processing grid features and object features and used an absolute geometric tensor to encode the positional information of each grid in the feature map. It is represented by the concatenation of two 1-d sine and cosine embeddings:

$$\begin{aligned}\mathrm{GPE}(i,j) &= \left[\mathrm{PE}_i;\mathrm{PE}_j\right],\\ \mathrm{PE}(\mathrm{pos},2k) &= \sin\left(\frac{\mathrm{pos}}{10000^{2k/(d/2)}}\right),\\ \mathrm{PE}(\mathrm{pos},2k+1) &= \cos\left(\frac{\mathrm{pos}}{10000^{2k/(d_{\mathrm{model}}/2)}}\right),\end{aligned} \tag{9}$$

where i and j are the row and column indices of the grid, respectively, and PE_* is the position encoding vector of the $d/2$ dimension. pos is the corresponding position, and k is each dimension. For object features, it directly maps the coordinates to the feature space. Its formula is as follows:

$$\mathrm{RPE}(i) = B_i W_{\mathrm{emb}}, \tag{10}$$

where $B_i = (x_{\min}, y_{\min}, x_{\max}, y_{\max})$ are the coordinates of the upper left corner and lower right corner of the object bounding box. W_{emb} is the embedding matrix. Absolute geometric features are geometric features aimed at fixed image regions, which can effectively improve the spatial separability of features, but they lack flexibility.

3.1.4. Geometric Graph. The data structure of a graph can naturally use edges to represent the relationship between nodes. Therefore, using the graph to represent the relationship in relational captioning is natural. Specifically, for the graph structure data $G = (V, E)$, its composition includes the node set V and the edge set E. Each node corresponds to an object in the image. In related tasks in the multimodal field, nodes generally contain corresponding node features, and the representation matrix of all nodes in the node set is $X \in R^{n\times d}$. In addition to the nodes, each edge in the edge set is represented as $e_{ij} = (v_i, v_j) \in E$. At the same time, if edge features are required, all edge feature matrices are $X^e \in R^{m\times c}$, where the feature of each edge between i-th and j-th objects is a c-dimensional vector $X^e_{i,j} \in R^c$.

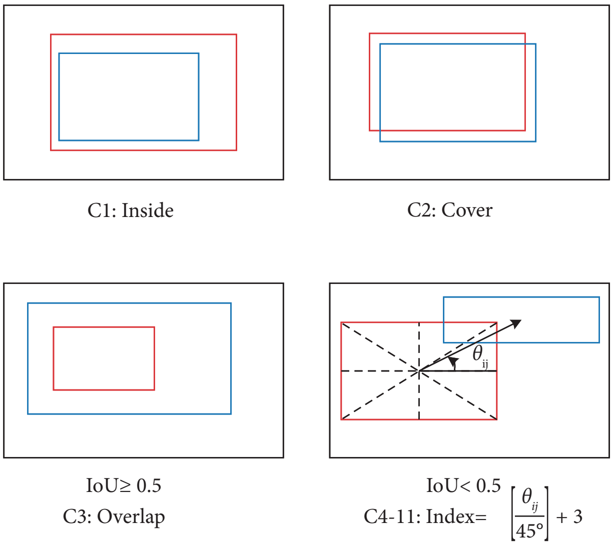
Since the edge represents the relationship between two objects, it can be expressed formally as follows: <subject-relation-object>, where subject indicates that the subject-object corresponds to v_i, an object indicates that the object corresponds to v_j. The neighbors of a node v can be expressed as $N(v) = \{u \in V | (v,u) \in E\}$. Its adjacency matrix A is a matrix of $n\times n$, where $A_{ij} = 1$ if $e_{ij} \in E$, $A_{ij} = 0$ if $e_{ij} \notin E$.

One approach to embedding relational information into the edges is to classify the positional relation and assign it as a label to each edge. Yao et al. [72] discretized the positional relationship based on the geometric features between two objects' boxes and assigned categories to each edge to build a directed graph. Specifically, according to the difference in the positional relationship between the two object boxes, they can be divided into 11 categories, as shown in Figure 4. Specifically, categories 1 and 2 are the inclusion and included relationships between the subject and the object, respectively. Category 3 is the overlapping relationship between the two objects with their IoU greater than or equal to 0.5. The remaining categories are divided into 8 categories according to the relative angle between the center points, representing 8 different positions, respectively. After classifying the positional relationship into a number of specific categories, the corresponding label is further assigned to each edge to construct the graph. An example of its graph structure is shown in Figure 5(a), which belongs to a directed fully connected graph. The feature corresponding to each edge is a specific category of positional relationship.

In summary, the graph-based approach can naturally utilize the adjacency matrix to characterize the relationship between objects. The graph is more interpretable and controllable than the tensor method. The tensor method is equivalent to processing an undirected fully connected graph when it uses full attention for subsequent learning. However, the relational content represented by each edge in the graph still depends on a small number of spatial categories, which result in poor performance in representing complex relational words in sentences.

3.2. Motion Relationship. The action relationship between objects is more specific than the positional relationship, which reflects the relationship at a higher semantic level. With the different data structures, the motion relation can also be divided into the following two forms: (1) tensor and (2) graph. The first method is more intuitive. The complexity of the motion relation makes it difficult to represent by the geometric feature. Therefore, many studies [73, 74, 78–81] begin to directly mine the information from the image content, extract the features of relevant image regions, and represent them in the form of tensor. The second method uses the graph pretrained by the upstream tasks to generate a suitable graph.

3.2.1. Semantic Tensor. Given an image and its N objects, the motion relation is represented in the form of a $N\times N\times d$ tensor. Specifically, for the action relationship between object i and object j, the tensor-based method attempts to extract the union content of the two objects in the image to represent the corresponding relationship. The extracted image area must contain two objects' bounding boxes simultaneously to ensure that the extracted content contains an accurate action relationship and avoid other noises as much as possible. The image region from which Zhang et al. [82] extracted features is the minimum circumscribing moment of the two object boxes, as shown in Figure 5. Specifically, for the coordinate (x_i, y_i, w_i, h_i) of the object i and the space coordinate vector (x_j, y_j, w_j, h_j) of the object j, the coordinate of the union box is follows:



C1: Inside

C2: Cover

IoU≥ 0.5
C3: Overlap

IoU< 0.5
C4-11: Index= $\left[\frac{\theta_{ij}}{45^\circ}\right] + 3$

FIGURE 4: The discretization of positional relation of each object's pair. The bounding boxes of subject and object are marked with red and blue, respectively.

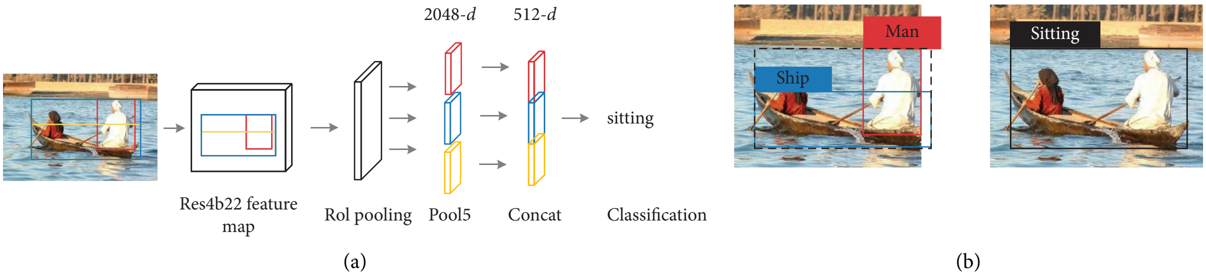


FIGURE 5: (a) The left part infers the corresponding relationship labels from the pretrained relationship detection network and (b) the right part represents the specific relationship through the feature of the union box between the two objects.

$$\min\left(x_i - \frac{w_i}{2}, x_j - \frac{w_j}{2}\right), \min\left(y_i - \frac{h_i}{2}, y_j - \frac{h_j}{2}\right),$$
$$\max\left(x_i + \frac{w_i}{2}, x_j + \frac{w_j}{2}\right), \max\left(y_i + \frac{h_i}{2}, y_j + \frac{h_j}{2}\right). \quad (11)$$

The union image area passes through the pretrained convolutional network to obtain the corresponding features. Each image can obtain a relation matrix of $N \times N \times d$ for different downstream tasks.

In summary, the tensor-based method stores the image features that characterize each relational region into relational tensors for the subsequent learning of relational information. This method is relatively straightforward, but it inevitably introduces noise. The noise here refers to relational information that is irrelevant to the relation contained in the generated sentence. At the same time, in general, there are many objects obtained by object detection. In the image description task, the model directly calculates all $N \times N$ relational features will bring a lot of computational costs. In terms of model performance, the quality of generated sentences is determined by the extracted features, which further depend on the structure of the pretrained convolutional network and its training objectives in upstream tasks. This leads to researchers needing to spend more energy on additional tasks. At the same time, after considering the additional pretrained network, the caption model is more computationally intensive overall.

3.2.2. Semantic Graph. The graph method use pretrained relationship detection networks in visual relation detection to extract action relations between objects and construct corresponding scene graphs. Specifically, Yao et al. [72] used the abovementioned method to build the graph, as shown in Figure 5. The pretrained model predicts the action relationship and uses the relationship category as the edge label. In each relational tuple <subject-predicate-object>, the subject and object are the 2048-dimensional attribute feature from the object detection network's RoI pooling. The image region feature corresponding initializes the feature of the predicate to the minimum circumscribing moment of two bounding boxes belonging to the subject and object. The above features are concatenated together and then input to the subsequent classification layer for obtaining the relationship category of the predicate. The $N \times (N - 1)$ relational tuples are input into (excluding self-relations) the relational classification network. Edges with a probability larger than 0.5 are kept to form an action graph, as shown in Figure 6(b).

Yang et al. [73] constructed scene graphs based on reference sentences in the training phase to reconstruct the sentence to accomplish the auto-encode training. The scene graph divides its nodes into three categories: object nodes, relational nodes, and attribute nodes. For each <subject-predicate-object> tuple, the subject and object correspond to the object node o_i and o_j. The l attribute of the object corresponds to the attribute node $a_{i,l}$, and the relationship between the two objects i, j corresponds to the relationship node r_{ij}. Each node in the scene graph is represented by a feature vector of $e_o, e_a, e_r \in R^d$, respectively. The object node o_i and all of its attribute nodes $a_{i,l}$ have connections by an edge from the object node to the attribute node. If there is a relationship node, the subject-object node o_i will first connect to the relationship node r_{ij}, and then the relationship node r_{ij} will connect to the object object node o_j. The constructed graph is shown in Figure 6(c). In terms of implementation, they adopt the scene graph constructor used in [83] first to convert sentences into syntactically independent trees and then convert the trees into scene graphs according to the rules mentioned in [75].

Chen et al. [74] designed a customized captioning model to generate sentences according to an abstract graph. The abstract graph is a scene graph customized according to the user's wish. The different forms of description graphs determine the level of detail in the generated caption. Specifically, the abstract graph is constructed by the combination of three types of nodes: (1) object nodes, (2) attribute nodes (representing a specific attribute of an object node), and (3) relationship nodes. The construction of the abstract graph is to add the nodes and edges into the graph according to the user's interests. Specifically, given all N object boxes of an image, if the user wants to know the content of the i object box, the object node o_i is added to the abstract graph. At the same time, if the user wants to know about the attribute characteristics contained in the object node o_i, l attribute nodes are added, and each attribute node corresponds to a path from o_i to $a_{i,l}$ directed edges. If the user wants to describe the relationship between two objects, add the corresponding relationship node $r_{i,j}$ in the abstract graph, and build the edge connection between the subject and the object. The subject-object node o_i points to the relationship node $r_{i,j}$, and then the relationship node $r_{i,j}$ points to the object object node o_j. The features corresponding to the object nodes and attribute nodes in the abstract graph adopt the visual features of the corresponding object bounding box. The extraction method for the relational node is mainly used to extract the union frame features of two objects. The result of its construction is shown in Figure 6(d).

In summary, the graph method represents more complex action relationships between objects than the tensor method. At the same time, some unnecessary relationship information is also eliminated, which can better retain important relationship content. There has also been a more significant improvement in computational cost and model performance. But the disadvantage is that it depends on the effectiveness of the relationship detection network and relies on training additional relationship information, which increases the complexity of the entire process. In the geometric graph, each edge represents a certain orientation. But in the semantic graph, each edge directly corresponds to a relational category. This more detailed representation of the relationship makes the semantic graph more effective to model the alignment of relational words. However, the limited number of relational categories also limits the variety of generated relational words. At the same time, the semantic similarity between different categories is also eliminated due to the classification operation.

3.3. Relational Encoding.

For a different type of relational data structure, the encoding methods can be divided into two methods: (1) tensor-based method and (2) graph-based method. This section mainly focuses on different relational encoding methods used in relational captioning.

3.3.1. Tensor-Based Method.

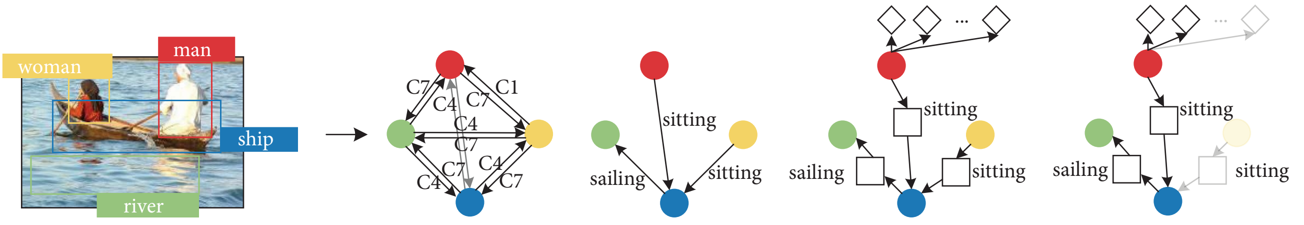
The tensor-based method is adopted when the positional relation information or the action relation information is extracted as a relation feature tensor. In this case, each image will correspond to a relational feature tensor $N \times N \times d$. If it is a geometric feature tensor between objects, then d is of size 4. And if it is the relational feature tensor extracted from the relational action information between objects, then the data of d depend on the dimension of the model.

3.3.2. Geometric Multiplier.

For the geometric tensor, Herdade et al. [62] used the tensor as a multiplier to adjust the attention weight in the self-attention of the encoder side. In Section 2, the weight calculation in the self-attention operator relies on the similarity between the query vector and the critical vector. The geometric tensor, the prior information of the positional relationship between objects, is used to adjust each weight element in the self-attention operator. Herdade et al. [62] use the following formula:

$$\omega_G^{i,j} = \mathrm{ReLU}\left(\mathrm{Emb}\left(\lambda\left(i,j\right)\right)W_G\right), \tag{12}$$

where $\lambda(i,j)$ represents the (i,j)th vector in the geometric tensor. Emb is an embedding layer, which first maps the geometric vector of 4 dimension to high-dimensional feature space and then calculates each element's positional information through sinusoidal position encoding. Finally, the d-dimensional vector is transposed to a scalar factor through W_G, and negative values are filtered through the ReLU activation function. Noted that the attention weight E in the self-attention operator describes the similarity of i-th and j -th objects in each element, which is the same as the geometric tensor (describing the positional information of i-th and j-th objects). As a result, taking $\omega_G^{i,j}$ as the scaling factor, adjust the element with the same i and j indexes in the attention weight E. The formula is shown as follows and Figure 7(a) shows the framework:



(a) (b) (c) (d)

Figure 6: Different types of graph structures are used when modeling the relationship between objects in an image. From the left to right, respectively, (a), (b), (c), and (d).

$$\omega^{i,j} = \frac{\omega_G^{ij}\exp\left(\omega_A^{ij}\right)}{\sum_{l=1}^{N}\omega_G^{il}\exp\left(\omega_A^{il}\right)}. \tag{13}$$

The geometric multiplier is designed to modulate the attention weight between each pair of objects for introducing the prior positional knowledge. Each value of the conventional attention weight E is like the similarity between i-th and j-th objects. With the shape identity, each value of the geometric tensor is assigned to the corresponding value with the same index in the attention weight. It is an ingenious and convenient way to introduce positional information in interactive learning. However, the effectiveness of generating better sentences is agnostic and uncontrollable.

3.3.3. Geometric Bias. In addition to scaling the similarity between the i-th and the j-th object in the weight matrix, Guo et al. [39] adopted a biased method to adjust attention weight. Specifically, the geometric tensor passes through a series of functions and is added to the original weight matrix as a deviation. Guo et al. [39] designed 3 functions for three types of geometric bias: (1) content-independent geometric bias, (2) query-dependent geometric bias, and (3) key-dependent geometric bias. The content-independent geometric bias is reasoned from the geometric tensor and is independent of the visual content. The geometric tensor is transformed into a scalar through a learnable parameter w_g^T. Then, it is directly added to the weight in the self-attention operator after being filtered by the ReLU nonlinear function. As shown in Figure 7(b), its calculation formula is as follows:

$$\begin{aligned} G_{ij} &= \mathrm{ReLU}\left(FC\left(f_{ij}^{g}\right)\right),\\ E &= \mathbf{QK}^{\mathbf{T}} + ReLU\left(\omega_g^{T}\mathbf{G}\right). \end{aligned} \tag{14}$$

Unlike the independent bias, the query-dependent and key-dependent geometric biases take a further step to compute the similarity with the visual query or key. As shown in Figure 7(c), the specific calculation method is as follows:

$$\begin{aligned} E &= \mathbf{QK}^{\mathbf{T}} + \mathbf{Q}'^{T}\mathbf{G},\\ E &= \mathbf{QK}^{\mathbf{T}} + \mathbf{K}'^{T}\mathbf{G}. \end{aligned} \tag{15}$$

Compared with the previous method, Luo et al. [83] used the geometric tensor, including the absolute position geometric tensor and the relative position geometric tensor. The absolute position geometric tensor is directly added to the query vector and key vector as the position feature vector, and the relative position geometric tensor is added as the deviation of the attention weight E. As shown in Figure 7(d), the calculation formula is as follows:

$$E = \frac{\left(\mathbf{Q} + \mathbf{pos_q}\right)\left(\mathbf{K} + \mathbf{pos_k}\right)^{T}}{\sqrt{d_k}} + \log(\boldsymbol{\Omega}), \tag{16}$$

where $\mathbf{pos}_*$ is the absolute position geometry tensor corresponding to each element in the query vector or key vector. Ω is the relative position geometry tensor. Like the multiplier method, the tensor-based process uses each element of the geometric tensor to function on the element of the attention weight with the same position. This method is straightforward and effective but less interpretable.

3.4. Graph-Based Methods. The graph-based method is specific to processing the graph data. The graph-structured data filter some unreasonable relationships through the prior knowledge learned in the pretrained model.

3.4.1. Label-Aware GCN. Yao et al. [72] designed a graph convolutional network to take the knowledge from the labeled edge and its direction (Figure 8). Each node considers all the connected labeled edges to fuse the relational label and its connected nodes.

Specifically, each image can be transformed into a semantic and positional graph to represent the motion and position relation. The semantic graph is directed, and its edges are labeled with the action relationship. The positional graph is an undirected graph with labeled edges. To make the graph convolutional network aware of the edge's label and its direction, each layer is designed as follows:

$$\begin{aligned} v_i^{t} &= \rho\left(\sum_{v_j\in N(v_i)} g_{v_i,v_j}\left(W_{\mathrm{dir}\left(v_i,v_j\right)}v_j + b_{\mathrm{lab}\left(v_i,v_j\right)}\right)\right),\\ g_{v_i,v_j} &= \sigma\left(\widetilde{W}_{\mathrm{dir}\left(v_i,v_j\right)}v_j + \widetilde{b}_{\mathrm{lab}\left(v_i,v_j\right)}\right), \end{aligned} \tag{17}$$

where $W_{\mathrm{dir}(v_i,v_j)}$ selects different transformation matrices according to the type of each edge. Specifically, if the i object v_i is the subject in a relation tuple <subject-relation-object>, then the transformation matrix is W_1; if the i object v_i is the object, then the transformation matrix becomes W_2.

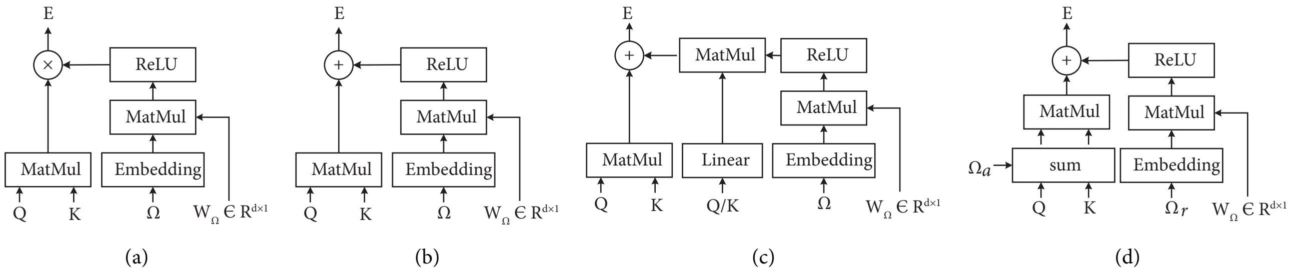


Figure 7: Geometric tensor methods.

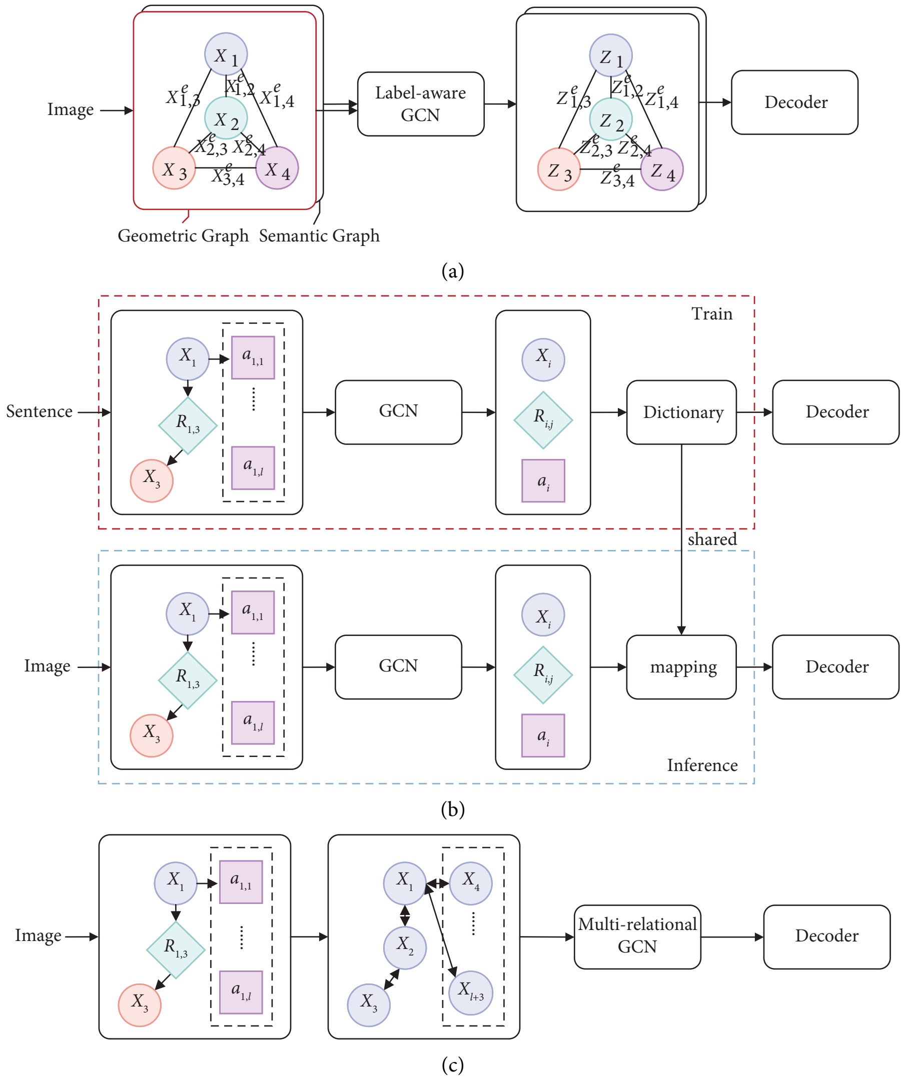


Figure 8: Graph-based methods: (a) label-aware GCN; (b) SGAE; (c) multirelational GCN with customized abstract graph.

Similarly, when dealing with the self-connected edge, the transformation matrix is set to be W_3. lab (v_i, v_j) represents the category of the edge. g_{v_i,v_j} is a weight function to determine the importance of the edge in the calculation. Compared with the conventional GCN, the label-aware GCN introduces the relationship information in each edge with the corresponding relational label. The label triggers the embedding function to form the edge features to fuse the connected nodes' relational information further. By introducing the graph, the connection between nodes determines the interactive learning and guides the model to generate the content between corresponding objects. It is more explainable than the geometric methods, which use the full-connected graph.

3.4.2. Scene Graph Auto-Encoder. Yang et al. [73] proposed the Scene Graph Auto-Encoder (SGAE) model to learn a recoder to optimize the original visual features through reconstruction of the sentence in training. The scene graph is constructed from the ground-true sentence, and each visual feature further fuses features according to the connection in the graph. It is shown in Figure 6(c), which includes object nodes, relational nodes, and attribute nodes.

$$\begin{aligned} x_{r_{ij}} &= g_r\left(e_{o_i}, e_{r_{ij}}, e_{o_j}\right), \\ x_{a_i} &= \frac{1}{Na_i}\sum_{l=1}^{Na_i} g_a\left(e_{o_i}, e_{a_{il}}\right), \\ x_{o_i} &= \frac{1}{Nr_i}\sum_{o_j\in<o_i-r_{i*}-o_*>} g_s\left(e_{o_i}, e_{r_{ij}}, e_{o_j}\right) \\ &\quad + \sum_{o_k\in<o_*-r_{*i}-o_i>} g_o\left(e_{o_k}, e_{r_{ki}}, e_{o_i}\right), \end{aligned} \tag{18}$$

where $x_{r_{ij}}$ is the node feature of the relation node r_{ij}, and its neighbor node features e_{o_i}, $e_{r_{ij}}$, and e_{o_j} belong to the corresponding node in the relation tuple $<o_i\text{-}r_{ij}\text{-}o_j>$. x_{a_i} represents the attribute information of the i object node, and its neighbor e_{o_i} and $e_{a_{il}}$ belong to the object node i and l-th attribute feature. An object may have multiple attributes, each attribute corresponds to an attribute node. N is the total number of all attributes. x_{o_i} represents the feature of the i-th object node, $<o_i\text{-}r_{i*}\text{-}o_*>$ represents all the tuples whose i -th object as the subject. $<o_*\text{-}r_{*i}\text{-}o_i>$ represents all the tuples whose i-th node is the object. After passing the abovementioned embedding, they use the form of a memory network to set up a dictionary matrix $\mathbf{D} \in R^{d\times V}$ to optimize the input node feature x. The calculation formula is as follows:

$$\hat{x} = \mathbf{D}\text{softmax}\left(\mathbf{D}^T x\right). \tag{19}$$

The optimized feature $\hat{x}$ is input to the subsequent decoder to regenerate the sentence and compare with the real input sentence. The error is fed back to the network for self-encoding training. The auto-encoder method uses the reconstruction to learn the semantic knowledge which begins from the sentence and regenerates it. The semantic knowledge reflects in the scene graph and assists the inference process. The whole framework is shown in Figure 8

3.4.3. Multirelational GCN. Chen et al. [74] proposed a customized abstract graph to generate specific captions. For representing each node, the features of the object nodes and attribute nodes adopt the visual features of the corresponding object bounding box, which are reasoned from the object detection network. The union bounding box's feature of two objects is used for the relational node. At the same time, Chen et al. made various types of nodes corresponding to different transformation matrices in feature embedding to further distinguish different kinds of nodes. The formula is shown as follows:

$$x_i^{(0)} = \begin{cases} v_i \odot W_r[0], & \text{if } i \in 0; \\ v_i \odot \left(W_r[1] + \text{pos}[i]\right), & \text{if } i \in a; \\ v_i \odot W_r[2], & \text{if } i \in r; \end{cases} \tag{20}$$

where $W_r[k]$ is the transformation matrix and its three matrices corresponding to three types of nodes. pos$[i]$ adds the order information for different attribute nodes $a_{i,l}$. According to the abovementioned embedding methods, the features of each node in the abstract graph are fused with their adjacency nodes. Meanwhile, the directed abstract graph is converted into an undirected graph which fits with the GCN. Chen et al. [74] designed a multirelational GCN (Figure 8) so that graph convolution learns different sets of parameters according to the edge types. There are six different types of edges: (1) object node to attribute node, (2) subject node to relational node, and (3) object node to relational node point and their inverse edges. The transformation transforms the direct graph into a unidirected graph and feeds into the multi-relational GCN to refine each node's feature. Different transformation matrices in each layer of the graph convolutional network are used to map the edges of different categories. Specifically, each layer is calculated as follows:

$$x_i^{l+1} = \sigma\left(W_o^l x_i^l + \sum_{r\in R}\sum_{j\in N}\frac{1}{N}W_r^l x_j^l\right), \tag{21}$$

where l represents the different layers in the graph convolutional network, the parameters for different classes of edges in each layer are shared. Through stacking encoders, each node feature is learned according to the connection between the nodes in the graph. The multirelational GCN is based on the abstract graph, which the user designs for generating the customized caption. The controllable ability has been improved, and the abstract graph determines the attribute, object, and relationship feature fed into the model.

In summary, Table 1 summarizes the methods used in relational feature construction and relational encoding by current methods in relational captioning.

4. Dataset and Evaluation

4.1. Dataset. The main datasets used in relational captioning are the following 4 datasets: (1) VisualGenome [84]; (2) MSCOCO [85]; (3) Flickr8K [86]/Flickr30k [87]; (4) PASCAL 1K [7].

4.1.1. VisualGenome. There are 108K images in total and many object annotations, attribute information annotations, and relationship annotations between objects for tasks such as object detection and visual relationship detection. In relational captioning, it is mainly used as a pretraining dataset to pretrain the object detection or the visual relationship detection network. In the pretraining stage, the training, validation, and test dataset split is followed by Anderson et al. [5]. Specifically, 98K images are used for training, and the remaining 10K images are divided into validation and test sets, respectively. When Yao et al. [72]

Table 1: Summary of the various methods in the relational captioning.

Methods	Feature construction	Relational encoding	Decoder
GCN-LSTM [72]	Positional relation: directed graph with label Motional relation: directed scene graph	Convolutional graph network	Two-LSTMs decoder
SGAE [73]	Motional relation: directed scene graph	Auto-encoder	Two-LSTMs decoder
ORT [62]	Positional relation: directed graph with label	Attention multiplier	Transformer
NG-SAN [39]	Positional relation: directed graph with label	Attention bias	Transformer
DLCT [83]	Positional relation: directed graph with label	Attention bias	Transformer

pretrained the target detection network, the dataset was filtered to retain 1600 object categories and 400 attribute categories. When dealing with pretrained object detection networks, it mainly selects the top 50 standard action relationships and artificially classifies them into 20 categories.

4.1.2. MSCOCO. The Microsoft COCO Captions dataset [85] is developed by Microsoft Team with the goal of scene understanding, capturing images from complex scenes, and can perform multiple tasks such as image recognition, segmentation, and captioning. The dataset uses Amazon's "Mechanical Turk" service to manually generate at least five sentences for each image. It contains more than 1.5 million sentences. The training set contains 82,783 images, the validation set contains 40,504 images, and the test set contains 40,775 images. In captioning tasks, the "Karpathy" split [5] is the standard data split method, which takes 5000 images in the validation set for evaluation and 5000 images for testing. The rest of the training and validation datasets are used for training.

4.1.3. Flickr8K/Flickr30k. Flickr8k [86] images are from Yahoo's photo album website Flickr, including 8,000 images, 6,000 images for training, 1,000 for evaluation, and 1,000 for testing. Flickr30k [87] contains 31,783 images collected from the Flickr website, mainly depicting human engagement. The manual label corresponding to each image is still five sentences.

4.1.4. PASCAL 1K. It is a subset of the well-known PASCAL VOC challenge image dataset [7], which provides a standard image annotation dataset and a standard evaluation system. The PASCAL VOC dataset consists of 20 categories. Amazon's Turk Robot service was then used to label each image with five descriptions manually. The dataset has the excellent image quality and complete annotation, which is suitable for testing algorithm performance.

4.2. Evaluation. The evaluation standard of relational captioning is consistent with the standard evaluation used in natural language processing to evaluate the similarity between the generated sentence and the ground-truth sentence. The evaluation metrics: BLEU [88], METEOR [89], ROUGE [90], CIDEr [91], and SPICE [92]. For the five metrics, BLEU and METEOR are used for machine translation, ROUGE for automatic translation summaries, and CIDEr and SPICE for image captioning. In principle, the abovementioned evaluation metrics measure the n-gram consistency between generated sentences and reference sentences and are also affected by the importance and rarity of n-grams in the corpus.

4.2.1. BLEU. As a widely used and essential evaluation metric in machine translation, BLEU [88] mainly measures the degree of the repetition between the generated sentence and the reference sentence. The number of identical n-grams in both generated and reference sentences determines the BLEU score. With the more significant number, the BLEU score is higher, meaning the generated sentences are closer to the reference sentences. With the increase of the n in n-gram, BLEU considers the correlation no longer limited to several words but prefers the correlation between contents. The higher the BLEU score, the better the generated sentences.

4.2.2. METEOR. METEOR [89] mainly considers the influence of synonyms and word forms in comparing generated sentences with all reference sentences. When evaluating the fluency of the sentence, METEOR is computed based on the chunks, which are constructed by considering the combination of semantically consecutive words. The word's consistency between the candidate and reference sentences is measured by the chunk. At the same time, METEOR is calculated by combining the precision, recall, and F-values of matching various cases. The higher the METEOR score, the better the sentence performance.

4.2.3. ROUGE. ROUGE [90] is a set of evaluation metrics designed to evaluate text summarization. ROUGE-L is used in relational captioning. It is calculated using the longest common subsequence between the generated and reference sentences. The score is calculated by summing the recall and precision of the longest common subsequence. The higher the ROUGE score, the better the sentence performance.

4.2.4. CIDEr. CIDEr [91] is an evaluation metric specially designed for captioning. It measures the consistency of image annotations by performing a term frequency-inverse document frequency (TF-IDF) weight calculation for each n-gram. This metric treats each sentence as a "document," represented as a TF-IDF vector, and then computes the cosine similarity between the generated sentence and the reference sentence. This indicator makes up for a shortcoming of BLEU, in which all words on the match are treated

Table 2: The evaluation scores of relative caption methods on COCO "*Karpathy*" test split.

Methods	B-1	B-4	M	R	C	S
GCN-LSTM [72]	80.5	38.2	28.5	58.3	127.6	22.0
SGAE [73]	80.8	38.4	28.4	58.6	127.8	22.1
ORT [62]	80.5	38.6	28.7	58.4	128.3	22.6
NG-SAN [39]	—	39.9	29.3	59.2	132.1	23.3
DLCT [83]	81.4	39.8	29.5	59.1	133.8	23.0

equally. Meanwhile, it considers the importance of the information of each word itself. Likewise, the higher the CIDEr score, the better the performance.

4.2.5. SPICE. SPICE [92] is a semantic evaluation metric for image captions, which measures how effectively image captions recover objects, attributes, and relationships between them. On the image captioning dataset, SPICE can better capture human judgments of model captions than existing n-gram metrics.

Table 2 shows the scoring index ranking of the models used in the current relational image description on the MSCOCO dataset.

5. Conclusion

This paper mainly summarizes the procedure of relational captioning and the development of each part in recent years. The relational captioning further focuses on the relationship between objects in the image. By introducing and incorporating the relationship information, the sentences generated by the model have better sufficiency and accuracy. We summarize the framework used in relational captioning and divide the relational procedure into two parts: feature construction and feature encoding. Combined with the characteristics of the relationship between objects, the relationship is further divided into the positional relationship and action relationship. The methods used for learning each relationship are discussed in the feature construction and encoding stages. In addition, we also summarize the datasets commonly used in relational captioning and the related evaluation metrics of the model.

We conclude by summarizing the current challenges in relational caption and clarifying our vision for this aspect. There are two main challenges in relational captioning, which is existed in feature construction and feature encoding. In terms of feature construction, it is challenging to find an appropriate method which considers as many relationship categories as possible while satisfying the content correlation between each relationship category on the textual modality. Second, in terms of feature encoding, it is challenging to make the feature perceive the semantic difference of various relational information and maintain its original visual knowledge. According to the abovementioned two challenges, we believe that future work has the following space for improvement in relational captioning:

(1) The feature construction of positional relationships is mainly limited to the handmade geometric feature extracted from objects' bounding box in 2-dimensional space. The geometric feature is susceptible to the scale of the object box.

(2) The feature of motional relationship depends on the performance of the pretrained feature extracted network. Better features can be obtained by adjusting the training objectives of the pretrained network in upstream tasks.

(3) About feature encoding, the current cross entropy or reinforcement learning training objectives make it difficult for the features output by the encoder to fully reflect the differences between different relationship categories while retaining visual knowledge. Compared with the end-to-end training method, the current pretraining-finetuning method [67–69] could use specialized objective function to obtain more powerful features.

(4) The alignment between relational features and relational vocabulary is ambiguous. The generation of relational vocabulary mainly depends on the global image information instead of relational features.

Data Availability

This paper is an overview paper in which the data reported are derived from corresponding published research studies. These prior studies (and datasets) are cited at relevant places within the text as references.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This study benefited from the Aerospace Information Research Institute.

References

[1] G. Kulkarni, V. Premraj, V. Ordonez et al., "Babytalk: understanding and generating simple image descriptions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 12, pp. 2891–2903, 2013.

[2] O. Vinyals, T. Alexander, S. Bengio, and D. Erhan, "Show and tell: a neural image caption generator," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, Nevada, USA, July, 2015.

[3] K. Xu, Ba Jimmy, K. Ryan et al., "Show, attend and tell: neural image caption generation with visual attention," in *Proceedings of the International Conference on Machine Learning*, Guangzhou, China, July, 2015.

[4] J. Lu, C. Xiong, D. Parikh, and R. Socher, "Knowing when to look: adaptive attention via a visual sentinel for image captioning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, August, 2017.

[5] P. Anderson, X. He, C. Buehler et al., "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 (CVPR)*, Salt Lake City, UT, USA, June, 2018.

[6] X. Zhu, L. Li, J. Liu, H. Peng, and X. Niu, "Captioning transformer with stacked attention modules," *Applied Sciences*, vol. 8, no. 5, p. 739, 2018.

[7] A. Farhadi, M. Hejrati, M. A. Sadeghi et al., "Every picture tells a story: generating sentences from images," in *Proceedings of the European Conference on Computer Vision*, Glasgow, UK, August, 2010.

[8] M. Cornia, L. Baraldi, and R. Cucchiara, "Show, control and tell: a framework for generating controllable and grounded captions," in *Proceedings of the European Conference on Computer Vision*, Tel Aviv, Israel, October, 2019.

[9] G. Li, L. Zhu, and P. Liu, "Entangled transformer for image captioning," in *Proceedings of the IEEE Transactions on Circuits and Systems for Video Technology*, Seoul, Korea (South), November, 2019.

[10] B. Z. Yao, X. Yang, L. Lin, Mun Wai Lee, and Song-Chun Zhu, "I2T: image parsing to text description," *Proceedings of the IEEE*, vol. 98, no. 8, pp. 1485–1508, 2010.

[11] Q. You, H. Jin, and Z. Wang, "Image captioning with semantic attention," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, July, 2016.

[12] S. J. Rennie, E. Marcheret, and Y. Mroueh, "Self-critical sequence training for image captioning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, June, 2017.

[13] M.'A. Ranzato, S. Chopra, and M. Auli, "Sequence level training with recurrent neural networks," in *Proceedings of the International Conference on Learning Representations*, Vancouver, BC, Canada, April, 2015.

[14] M. Cornia, M. Stefanini, and L. Baraldi, "Meshed-memory transformer for image captioning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Seattle, WA, USA, June, 2020.

[15] L. Guo, J. Liu, and J. Tang, "Aligning linguistic words and visual semantic units for image captioning," in *Proceedings of the 27th ACM International Conference on Multimedia*, Nice, France, October, 2019.

[16] W. Jiang, Ma Lin, and Yu-G. Jiang, "Recurrent fusion network for image captioning," in *Proceedings of the European Conference on Computer Vision*, Chapel Hill, NC, UK, May, 2018.

[17] L. Huang, W. Wang, and ie Chen, "Attention on attention for image captioning," in *Proceedings of the International Conference on Computer Vision*, Cambridge, MA, USA, June, 2019.

[18] Y. Pan, T. Yao, and Y. Li, "X-linear attention networks for image captioning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, June, 2020.

[19] Y. N. Dauphin, A. Fan, and M. Auli, "Language Modeling with Gated Convolutional Networks," in *Proceedings of the 34th International Conference on Machine Learning*, ICLR, Sydney, Australia, August, 2016.

[20] T. Yao, Y. Pan, and Y. Li, "Hierarchy parsing for image captioning," in *Proceedings of the International Conference on Computer Vision*, Xiamen China, August, 2019.

[21] D. Liu, Z.-J. Zha, and H. Zhang, "Context-aware visual policy network for sequence-level image captioning," in *Proceedings of the 2018 ACM Multimedia Conference on Multimedia Conference*, Seoul Republic of Korea, October, 2018.

[22] J. Lu, J. Yang, and D. Batra, "Neural baby talk," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, June, 2018.

[23] G. Deco and M. L. Kringelbach, "Hierarchy of information processing in the brain: a novel 'intrinsic ignition' framework," *Neuron*, vol. 94, no. 5, pp. 961–968, 2017.

[24] A. Farhadi, M. Hejrati, and M. A. Sadeghi, "Every picture tells a story: generating sentences from images," in *Proceedings of the European Conference on Computer Vision*, Glasgow, UK, December, 2010.

[25] V. Ordonez, G. Kulkarni, and T. L. Berg, "Im2text: Describing Images Using 1 Million Captioned Photographs," in *Proceedings of the 24th International Conference on Neural Information Processing Systems*, Guangzhou, China, November, 2011.

[26] J. Aneja, A. Deshpande, and S. Alexander, "Convolutional image captioning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, June 2018.

[27] L. Zhou, Y. Zhang, and Yu-G. Jiang, "Re-caption: saliency-enhanced image captioning through two-phase learning," *IEEE Transactions on Image Processing*, vol. 32, 2020.

[28] T. Yao, Y. Pan, Y. Li, Z. Qiu, and T. Mei, "Boosting image captioning with attributes," in *Proceedings of the IEEE Conference on International Conferenceon Computer Vision*, pp. 4904–4912, Las Vegas, NV, USA, June 2016.

[29] M. Pedersoli, T. Lucas, C. Schmid, and J. Verbeek, "Areas of attention for image captioning," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Vegas, NV, USA, June 2017.

[30] H. R. Tavakoli, R. Shetty, B. Ali, and J. Laaksonen, "Paying attention to descriptions generated by image captioning models," in *Proceedings of the IEEE Conference on International Conference on Computer Vision*, pp. 2506–2515, Venice, Italy, October 2017.

[31] R. Gerber and H.-H. Nagel, "Knowledge Representation for the Generation of Quantified Natural Language Descriptions of Vehicle Traffic in Image Sequences," in *Proceedings of the 3rd IEEE International Conference on Image Processing*, Lausanne, Switzerland, September, 1996.

[32] S. Rothe, S. Narayan, and A. Severyn, "Leveraging Pre-trained Checkpoints for Sequence Generation Tasks," *Transactions of the Association for Computational Linguistics*, 2019.

[33] C. Lu, R. Krishna, and M. Bernstein, "Visual Relationship Detection with Language Priors," in *Proceedings of the European Conference on Computer Vision*, Glasgow, UK, September, 2016.

[34] A. Aker and R. Gaizauskas, "Generating image descriptions using dependency relational patterns," *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, vol. 49, no. 9, pp. 1250–1258, 2010.

[35] S. Li, G. Kulkarni, T. L. Berg, and Y. Choi, "Composing simple image descriptions using web-scale N-grams," in *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*, pp. 220–228, Association for Computational Linguistics, Portland, OR, USA, June 2011.

[36] Y. Yang, C. L. Teo, H. Daume, and Y. Aloimonos, "Corpusguided sentence generation of natural images," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 444–454, Edinburgh, UK, July 2011.

[37] D. Lin, C. Kong, S. Fidler, and R. Urtasun, "Generating multisentence lingual descriptions of indoor scenes," pp. 2333–9721, Computer Science, 2015, http://arxiv.org/abs/1503.00064.

[38] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: towards real-time object detection with region proposal networks," in *Proceedings of the Neural Information Processing Systems (NIPS)*, pp. 91–99, Lake Tahoe Nevada, December, 2015.

[39] O. Russakovsky, J. Deng, H. Su et al., "ImageNet large scale visual recognition challenge," 2014, https://arxiv.org/abs/1409.0575.

[40] K. He, X. Zhang, and S. Ren, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, July, 2016.

[41] J. B. Lei, J. R. Kiros, and G. E. Hinton, "Layer normalization," 2016, https://arxiv.org/abs/1607.06450.

[42] D. P. Kingma and B. Jimmy, "Adam: a method for stochastic optimization," in *Proceedings of the International Conference on Learning Representations*, Vancouver, BC, Canada, May, 2015.

[43] R. Girshick, "Fast R-CNN," in *Proceedings of the International Conference on Computer Vision (ICCV)*, Montreal, BC, Canada, August, 2015.

[44] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[45] A. Mathews, L. Xie, and X. He, "SemStyle: learning to generate stylised image captions using unaligned text," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, June 2018.

[46] T.-H. Chen, Y.-H. Liao, C.-Y. Chuang, W.-T. Hsu, J. Fu, and M. Sun, "Show, adapt and tell: adversarial training of crossdomain image captioner," in *Proceedings of the IEEE Conference on International Conference on Computer Vision and Pattern Recognition*, pp. 521–530, Honolulu, HI, USA, July 2017.

[47] C. C. Park, B. Kim, and G. Kim, "Towards personalized image captioning via multimodal memory networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 99, p. 1, 2018.

[48] R. Zhou, X. Wang, N. Zhang, X. Lv, and L.-J. Li, "Deep reinforcement learning-based image captioning with embedding reward," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1151–1159, Honolulu, HI, USA, July 2017.

[49] A. Jacob, R. Marcus, Darrell Trevor, and D. Klein, "Neural module networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5735–5744, Honolulu, HI, USA, June, 2016.

[50] X. Chen, Ma Lin, W. Jiang, J. Yao, and W. Liu, "Regularizing RNNs for caption generation by reconstructing the past with the present," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, June 2018.

[51] J. Yu, J. Li, and Z. Yu, "Multimodal transformer with multiview visual representation for image captioning," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, 2020.

[52] Y. Song and M. Soleymani, "Polysemous visualsemantic embedding for cross-modal retrieval," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, USA, July, 2019.

[53] C. Sun, A. Myers, and C. Vondrick, "Videobert: a joint model for video and language representation learning," in *Proceedings of the International Conference on Computer Vision*, Long Beach, CA, USA, July, 2019.

[54] P. Young, A. Lai, M. Hodosh, J. Hockenmaier, and M. Kringelbach, "From Image Descriptions to Visual Denotations: New Similarity Metrics for Semantic Inference over Event Descriptions," *Transactions of the Association for Computational Linguistics*, vol. 2, 2014.

[55] Y. Gong, L. Wang, and M. Hodosh, "Improving Image-Sentence Embeddings Using Large Weakly Annotated Photo Collections," in *Proceedings of the European Conference on Computer Vision*, Chapel Hill, NC, UK, September, 2014.

[56] K. Cho, Bart van Merrienboer, and D. Bahdanau, "On the properties of neural machine translation: encoder-decoder approaches," 2014, https://arxiv.org/abs/1409.1259.

[57] K. Cho, Bart van Merrienboer, and C. Gulcehre, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1724–1734, Doha, Qatar, 2014.

[58] S. Ding, S. Qu, Y. Xi, and S. Wan, "A long video caption generation algorithm for big video data retrieval," *Future Generation Computer Systems*, 2019.

[59] Yu Qin, J. Du, and Y. Zhang, "Look back and predict forward in image captioning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, July, 2019.

[60] L. Ke, W. Pei, R. Li, X. Shen, and Y. W. Tai, "Reflective decoding network for image captioning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Long Beach, CA, USA, July, 2019.

[61] S. Ding, S. Qu, Xi, Y. Wan, and Shaohua, "Stimulus-Driven and Concept-Driven Analysis for Image Caption Generation," *Neurocomputing,* vol. 398, pp. 520–530, 2020.

[62] S. Herdade, A. Kappeler, K. Boakye, and J. Soares, "Image captioning: transforming objects into words," 2019, https://arxiv.org/abs/1906.05963.

[63] L. Guo, J. Liu, X. Zhu, and P. Yao, "Normalized and geometry-aware self-attention network for image captioning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, June, 2020.

[64] A. Vaswani, N. Shazeer, N. Parmar et al., "Attention is all you need," in *Proceedings of the Neural Information Processing Systems (NIPS)*, Lake Tahoe Nevada, December, 2017.

[65] R. Girshick, J. Donahue, and Trevor Darrell, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Columbus, OH, USA, June, 2014.

[66] S. Liu, Z. Zhu, and N. Ye, "Improved image captioning via policy gradient optimization of SPIDEr," in *Proceedings of the International Conference on Computer Vision*, Honolulu, HI, USA, June, 2017.

[67] L. Zhou, P. Hamid, and L. Zhang, "Unified vision-languagepre-training for image captioning and vqa," in *Proceedings of the AAAI Conference on Artificial Intelligence*, Honolulu, HI, USA, January, 2020.

[68] X. Li, Xi Yin, and C. Li, "Oscar: object-semantics aligned pre-training for vision-language tasks," in *Proceedings of the European Conference on Computer Vision*, Glasgow, UK, September, 2020.

[69] Y. Li, Y. Pan, and T. Yao, "Scheduled sampling in vision-language pretraining with decoupled encoder-decoder network," in *Proceedings of the AAAI Conference on Artificial Intelligence*, Honolulu, HI, USA, February, 2021.

[70] Y. Li, Y. Pan, and T. Yao, "Comprehending and ordering semantics for image captioning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, New Orleans, LA, USA, June, 2022.

[71] Y. Li, Y. Pan, and J. Chen, "X-Modaler: A Versatile and High-Performance Codebase for Cross-Modal Analytics," 2021, https://arxiv.org/abs/2108.08217.

[72] T. Yao, Y. Pan, Y. Li, and T. Mei, "Exploring visual relationship for image captioning," in *Proceedings of the European Conference on Computer Vision*, Glasgow, UK, August, 2018.

[73] Xu Yang, K. Tang, H. Zhang, and J. Cai, "Auto-encoding scene graphs for image captioning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, June, 2019.

[74] S. Chen, Q. Jin, and P. Wang, "Say as you wish: fine-grained control of image caption generation with abstract scene graphs," in *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, June, 2020.

[75] S. Schuster, R. Krishna, A. Chang, L. Fei-Fei, and C. D. Manning, "Generating semantically precise scene graphs from textual descriptions for improved image retrieval," in *Proceedings of the fourth workshop on vision and language*, pp. 70–80, Lisbon, Portugal, September, 2015.

[76] M. Defferrard and Bresson, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, March, 2016.

[77] Y. Luo, J. Ji, and X. Sun, "Dual-level collaborative transformer for image captioning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Association for the Advance of Artificial Intelligence, Nashville, TN, USA, July, 2021.

[78] R. Kiros, R. Salakhutdinov, and R. S. Zemel, "Unifying visual-semantic embeddings with multimodal neural language models," 2014.

[79] Q. You, Z. Zhang, and J. Luo, "End-to-end convolutional semantic embeddings," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5735–5744, Salt Lake City, UT, USA, June 2018.

[80] W. Hu, H. Zhao, Li Jiang, J. Jia, and T.-T. Wong, "Bi-directional projection network for cross dimension scene understanding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Nashville, TN, USA, June, 2021.

[81] X. Li, A. You, Z. Zhu, and H. Zhao, "Semantic Flow for Fast and Accurate Scene Parsing," in *Proceedings of the European Conference on Computer Vision*, Tel Aviv, Israel, October, 2020.

[82] H. Zhang, Z. Kyaw, S. F. Chang, and T. S. Chua, "Visual translation embedding network for visual relation detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, July, 2017.

[83] D. Klein and C. D. Manning, "Accurate unlexicalized parsing," in *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pp. 423–430, Association for Computational Linguistics, Japan, July, 2003.

[84] R. Krishna, Y. Zhu, O. Groth et al., "Visual genome: connecting language and vision using crowdsourced dense image annotations," *International Journal of Computer Vision*, vol. 123, 2017.

[85] X. Chen, H. Fang, T. Y. Lin et al., "Microsoft COCO captions: data collection and evaluation server," 2015, https://arxiv.org/abs/1504.00325.

[86] C. Rashtchian, P. Young, M. Hodosh, and J. Hockenmaier, "Collecting image annotations using amazon's mechanical turk," in *Proceedings of the NAACL HLT Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pp. 139–147, Los Angeles, CA, USA, June, 2010.

[87] P. Young, A. Lai, M. Hodosh, and J. Hockenmaier, "From image descriptions to visual denotations: new similarity metrics for semantic inference over event descriptions," *Transactions of the Association for Computational Linguistics*, vol. 2, 2014.

[88] P. Kishore, 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*, Philadelphia, Pennsylvania, July, 2002.

[89] S. Banerjee and L. Alon, "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*, Michigan, USA, March, 2005.

[90] C.-Y. Lin, "Rouge: a package for automatic evaluation of summaries," in *Proceedings of the ACL Workshop on Text summarization branches out*, Barcelona, Spain, February, 2004.

[91] R. Vedantam, C. Lawrence Zitnick, and D. Parikh, "Cider: consensusbased image description evaluation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Boston, MA, USA, June, 2015.

[92] P. Anderson, B. Fernando, M. Johnson, and S. Gould, "Spice: semantic propositional image caption evaluation," in *Proceedings of the European Conference on Computer Vision*, Glasgow, UK, August, 2016.