Academic Journal of Science and Technology
ISSN: 2771-3032 | Vol. 2, No. 1, 2022

Feature-Fusion Parallel Decoding Transformer for Image Captioning

Chenhao Zhu*, Xia Ye, Qiduo Lu

Xi'an Research Institute of High-tech, Xi'an, 710025, China
*972778371@qq.com

Abstract: Image caption is an important research direction at the intersection of computer vision and natural language processing. It is based on object detection, enabling machines to describe image content in human language, generating sentences with correct grammar. Most of the existing methods employ a Transformer-based structure which achieve the cutting-edge performance. However, most methods focus on improving visual feature information extraction, optimizing and improving between grid features and region features, and improving the performance of the final model. In this paper, we tried to improve the final effect of the model from the perspective of the model structure and the visual features extraction. We proposed Feature-Fusion Parallel Decoding Transformer (FPDT) which adopts parallel decoding mode and uses both grid features and region features. We conducted a large number of experimental studies on the MSCOCO dataset. And FPDT's performance on MSCOCO datasets is also at the cutting edge.

Keywords: Enter key words or phrases in alphabetical order, separated by commas.

1. Introduction

Image caption has many applications in the real world, such as: augmented reality technology, helping visually impaired people, human-machine interaction technology for smart glasses, etc. It is an important technical issue in the process of realizing a high degree of intelligence in future machines.

In recent years, most image caption tasks [2-6] are based on encoder-decoder model which has shown good performance on this task. Among them, the CNN and RNN are used as encoders and decoders to extract image features and generate corresponding image descriptions, respectively. After the self-attention mechanism and Transformer model was proposed by Vaswani et al.[7], Transformer model and attention mechanism have been proved to have good mobility and adaptability to multiple downstream tasks, and perform better than CNN and RNN in image retrieval, natural language understanding and generation tasks. Since then, most efforts to improve image captioning model focus on optimizing visual feature extraction methods and improving model structure for feature processing. In terms of feature extraction, two main methods are region features [8] and grid features [4] respectively. At first, region features outperform grid features. Because region features can represent the object level information of the most significant region in the image, it is represented by feature vectors, which greatly reduces the difficulty of visual semantic embedding and thus improves the performance of the model. However, it does still have many shortcomings, such as the loss of fine granularity of detection, low coverage of image content and time-consuming. For these problems, grid features can be a good solution. Recently, Jiang et al. [9] demonstrated that grid features not only have the same accuracy as region features, but also have an order of magnitude faster inference speed. On the basis of previous work, we try to fuse grid features and region features to improve the final performance of our model.

The main contributions of our work are summarized as follows:

1. We propose the Feature Augment module to fuse the region features and region features. The correlation between features can be judged by calculating the similarity between them.

2. For the structure of the Transformer, we propose the Parallel Input Decoder module, which could decode multi-layer encoding vectors and improve the performance of the model.

2. Related Work

At first, grid features of fixed size extracted by CNN are used as the input of encoder. After encoding, the output of encoder is passed into the decoder composed of RNN or LSTM to get the image description. Vinyas et al. [4] proposed Neural Image Caption model (NIC). NIC is based on Encoder-Decoder structure. It uses CNN as encoder to extract visual features of the image and RNN as decoder to generate descriptions. Xu et al. [6] improved the work of Vinyas et al. [4]. They used the soft attention mechanism to weight the image features with the hidden layer state of LSTM at each time step, and input the weighted features and words into LSTM to get the results. Because the hidden layer state of LSTM contains context information, this weighting can select important information to the current time step to a certain extent. Mao et al. [13] proposed a multimodal recurrent neural network model (M-RNN), which fused hidden states of RNN layers, word sequence features, and visual features to generating captions. Chen et al. [11] proposed a SCA-CNN model, which makes full use of the spatial and channel characteristics of CNN to decode at each time step. Considering that the generated words do not always depend on visual information, Lu et al. [10] proposed an adaptive attention model with visual sentry, which can decide when to pay attention to visual information and language model.

With the development of feature extraction methods, region features [8] bring improved performance to image subtitle models. Herdade et al. [12] proposed to integrate the geometric relationship information between regional features into the decoding process, which enhanced the interaction

between regional features. Cornia et al. [14] proposed the Meshed-Memory Transformer model, which uses region features to represent the images. It also meshed the encoder and decoder connectivity relationships.

Although the use of region features improves the performance of the image caption models, region features still have some problems such as time-consuming and low coverage of image details. However, these are precisely the advantages of the grid features of traditional methods. The RSTNet proposed by Zhang et al. [15] uses grid features to represent the images and achieves good results. On the basis of previous works, we try to fuse region features and grid features to enrich the visual feature information and give consideration to the subject object and detail content of the image, so as to achieve better results in the final description.

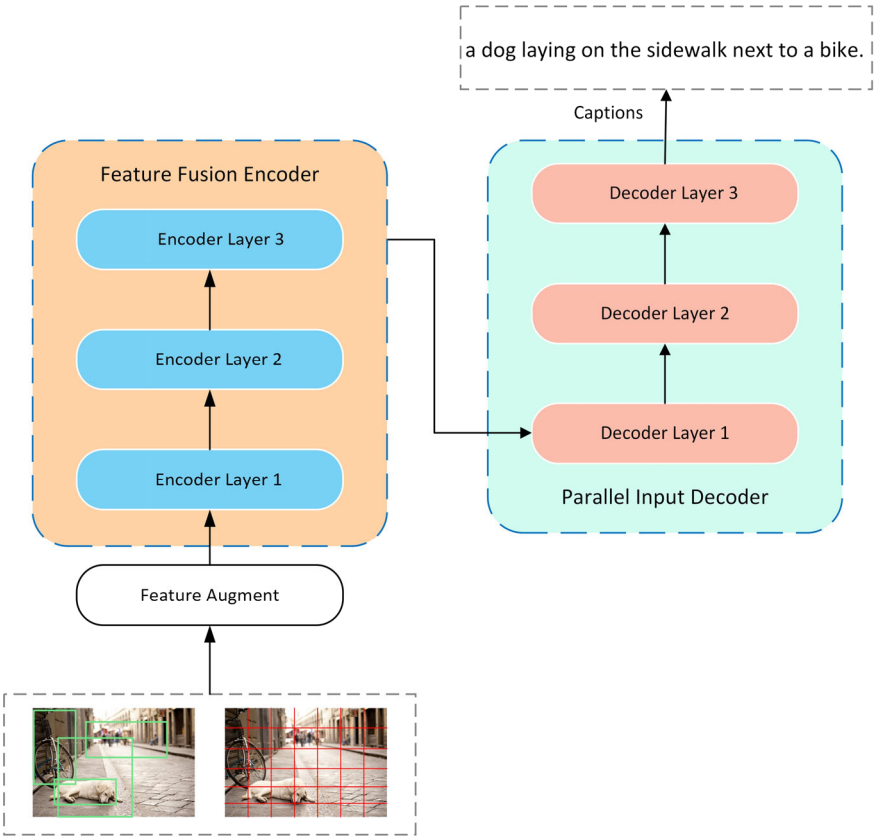

Figure 1. The overall architecture of FPDT model. The specific structure of the encoder layer is shown in Fig.2.

3. Model architecture

3.1. Overall Architecture

The overall structure of the model is shown in Figure 1. We take two features as inputs to the FPDT model's encoder. The structure and principles of each part are described next.

3.2. Feature-Fusion Encoder

Encoder Layer. The structure of the encoder layer is shown in Figure 2, where AddNorm represents the residual join and normalization operation. Before the features are fed into the encoder layer, we fuse the region features and grid features through the Feature Augment module. This module can fuse the related regional features and grid features, and retain the non-related features, so as to reduce noise and information repetition. And the complete operator is defined as follows:

$$S_{ij} = \max(\cos(< g_i, r_j >), t) \tag{1}$$

$$P = \begin{cases} g_i \oplus r_j & S_{ij} > t \\ g_i, r_j & S_{ij} < t \end{cases} \tag{2}$$

where $\oplus$ in Eq. (8) represents the addition of matrix elements. The max( ) is the maximum function, which takes the maximum between the similarity score S_{ij} and t. Note that t is a threshold value used to determine whether two features represent the same region in the image. If the similarity score S_{ij} is smaller than threshold value t, it will retain the region features and grid features. The cos() represents the cosine similarity function which used to measure the correlation between region features and grid features.

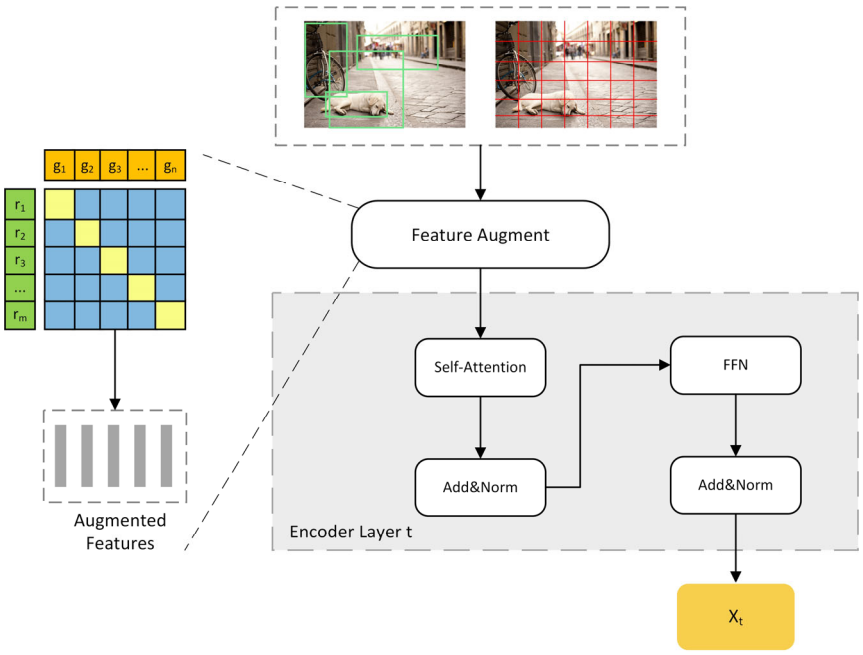

Figure 2. The structure of the encoder layer.

After Feature Augment module, we fed the augmented features into the self-attention layer and position-wise feedforward layer, and its whole operation is defined as follows:

$$Q = PW_q, K = PW_k, V = PW_v, \tag{3}$$

$$Z = soft\max(\frac{QK^T}{\sqrt{d_k}})V, \tag{4}$$

$$Y = AddNorm(Z), \tag{5}$$

$$X_t = AddNorm(\sigma(W_1 Y + b_1)W_2 + b_2), \tag{6}$$

where the W_q, W_k, W_v represents the learnable weight matrices and σ represents the RELU activation function. The b_1 and b_2 are the bias terms, W_1 and W_2 are the learnable weight matrices.

Full Encoder. In the above definition, we use 3 encoder layers and multi-head attention in the specific implementation. Through the feature augment operation to fuse the region features and grid features. So that the FPDT model can obtain good fine-grained details and sufficient image information to ensure the decoding effect.

3.3. Parallel Input Decoder

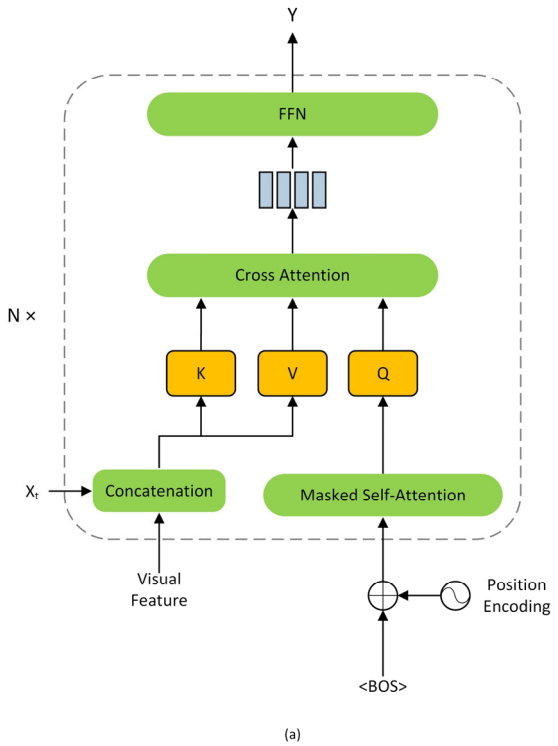

Figure 3. (a) The structure of decoder layer

The encoder focuses on understanding and encoding the information of significant regions, while the decoder focuses on generating the final caption result based on the output of the encoder. In decoding stage, the effect of final image description largely depends on the information passed in the decoding stage. In order to make the decoder decode better, we try to enhance the input encoding vector. The structure of decoder layer is shown in Figure 3.

Decoder Layer. As shown in the Figure 3, the word sequence is processed by word embedding and the position encoding. We use sinusoidal positional encoding [7] to represents the words' positions in the sequence. In order to improve the final decoding effect, we introduce the output X_t of different encoder layers t into the decoder layer t as additional visual information. The X_t is concatenated with the visual features V as a parallel input to the decoder layer. The operators of Parallel Input Cross Attention are defined as follows:

$$Z = [V, X_t] \tag{7}$$

$$S = Attention(W_q Y, W_k Z, W_v Z) \tag{8}$$

where the [,] in Eq. (7) represents concatenation operation. In Eq. (8), W_k and $W_v \in R^{2d \times d}$ are learnable weight matrices.

Full Decoder. In the above definition, we use 3 decoder layers which are stacked in sequence in the specific implementation.

3.4. Training Details

Refer to the training methods commonly used in image caption tasks, we optimize the FPDT model with the cross-entropy loss (XE):

$$L_{XE}(\theta) = -\sum_{t=1}^{T} \log\left(p_\theta\left(y_t^* \mid y_{1:t-1}^*\right)\right) \tag{9}$$

where the $y_{1:T}^*$ is ground truth sequence and the θ is the parameters of our model which needs to be optimized.

After that, we continually optimize the non-differentiable CIDER-D score by Self-Critical Sequence Training (SCST):

$$\nabla_\theta L_{RL}(\theta) = -\frac{1}{k}\sum_{i=1}^{k}\left(r\left(y_{1:T}^i\right) - b\right)\nabla_\theta \log p_\theta\left(y_{1:T}^i\right) \tag{10}$$

where the k is the beam size, $y_{1:T}^i$ is the i-th sentence in the beam. $r(\ )$ is the CIDEr-D function, and $b = (\sum_i r(y_{1:T}^i))/k$ is the baseline.

4. Experiment and Analysis

4.1. Dataset

The data set used in our experiment is MSCOCO. MSCOCO is a dataset provided by the Microsoft team. It contains more than 120,000 images. Each image has 5 different annotations. In addition to this, Karpathy *et al*. provide another split where 5,000 images are used for evaluation, 5,000 for testing and the rest for training.

4.2. Experimental Settings

Evaluation Metrics. Following the standard evaluation protocol, we use the full set of captioning metrics to evaluate the quality of our IEAT model, including BLEU [22], METEOR [33], ROUGE [34], CIDEr [35], and SPICE [36].

Implementation details. In terms of the extraction of visual features, we utilize the region features and the grid features provided by Jiang et al. [9] where the grid size is 7 × 7 and the dimension of image features is 2048. In our implementation, we set the number of heads equal to 8, the batch size equal to 50 and the number of memory slots is 40. The d_{model} of each layer is 512 and the inner dimension of FFN is 2048. The threshold value t is set to 0.6. The dropout probability is set to 0.1 and we set the beam size to 5.

We adopt Adam optimizer to train our model. In the XE pre-training stage, our learning rate strategy is defined as follows:

$$lambda\_lr = \begin{cases} base\_lr * e / 3, & e \le 3 \\ base\_lr, & 4 \le e \le 10 \\ base\_lr * 0.2, & 11 \le e \le 12 \\ base\_lr * 0.2 * 0.2, & 13 \le e \end{cases} \tag{11}$$

where the $base\_lr$ is set to 1×10^{-4} and the e represents the current epoch number. During the XE pre-training stage, if the CIDEr value drops for 5 consecutive epochs, we will switch it to self-critical sequence training and the learning rate is set to a fixed value of 5×10^{-6}. The training process will stop when the CIDEr value drops for 5 consecutive epochs in self-critical sequence training.

4.3. Comparisons with State-of-the-art Methods

Offline Evaluation. We report the performance of the FPDT model with models that have achieved cutting-edge results in recent years in the same experimental settings, including SCST [16], Up-Down [1], RFNet [17], GCN-LSTM [18], SGAE [19], ORT [20], AoANet [21], M^2Transformer [23], X-Transformer [22]. The results are shown in Table 2.

Table 1. Comparison with the state of art models published on the MSCOCO online leaderboard. C5 means there are 5 standard captions per image while c40 means there are 40 captions per image.

	B-1		B-2		B-3		B-4		M		R		C	
	C5	C40	C5	C40	C5	C40	C5	C40	C5	C40	C5	C40	C5	C40
SCST	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	116.7
Up-Down	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	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	80.8	95.9	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	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	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
M^2Transformer	81.6	96.0	66.4	90.8	51.8	82.7	39.7	72.8	29.4	39.0	59.2	74.8	129.3	132.1
X-Transformer	81.3	95.4	66.3	90.0	51.9	81.7	39.9	71.8	29.5	39.0	59.3	74.9	129.3	131.4
FPDT	81.5	96.2	66.5	91.0	51.7	81.9	40.3	72.9	29.6	39.2	59.3	75.1	129.5	131.8

Table 2. Comparison with the state of the art on the "Karpathy" test split, in single-model setting. The B-1, B-4, M, R, C, S are the abbreviation for BLEU-1, BLEU-4, METEOR, ROUGE, CIDEr and SPICE scores, and all values are reported as a percentage.

	B-1	B-4	M	R	C	S
SCST	-	34.2	26.7	57.7	114.0	-
Up-Down	79.8	36.3	27.7	56.9	120.1	21.4
RFNet	79.1	36.5	27.7	57.3	121.9	21.2
GCN-LSTM	80.5	38.2	28.5	58.3	127.6	22.0
SGAE	80.8	38.4	28.4	58.6	127.8	22.1
ORT	80.5	38.6	28.7	58.4	128.3	22.6
AoANet	80.2	38.9	29.2	58.8	129.8	22.4
M^2Transformer	80.8	39.1	29.2	58.6	131.2	22.6
X-Transformer	80.9	39.7	29.5	59.1	132.8	23.4
FPDT	81.1	39.7	29.6	59.2	133.2	23.1

Table 3. Comparison with the state of art models on the "Karpathy" test split, 4 models are used for ensemble.

	B-1	B-4	M	R	C	S
SCST	-	35.4	27.1	56.6	117.5	-
GCN-LSTM	80.9	38.3	28.6	58.5	128.7	22.1
SGAE	81.1	39.0	28.4	58.9	129.1	22.2
AoANet	81.6	40.2	29.3	59.4	132.0	22.8
M^2Transformer	82.0	40.5	29.7	59.5	134.5	23.5
X-Transformer	81.7	40.7	29.9	59.7	135.3	23.8
FPDT	81.8	40.9	29.9	59.8	135.6	23.5

As shown in the Table 2, except the Spice score, our model FPDT outperforms the other models in BLUE-1，BLUE-4, Meteor, Rouge and CIDEr scores. And in CIDEr score our model achieves a 133.2% performance ahead of the X-Transformer model, the best model in the nine models, by about 1%. The outstanding performance in the scores verifies the effectiveness of our proposed model. In Table 3, we report the FPDT's performance compared with other models, using ensembles of four models. Our ensembled model has improved to some extent in all scores. Except for the Spice score, which is slightly lower than the X-Transformer model, the other five scores are still the best.

Online Evaluation. We also report the performance of the model on the online COCO test server. The COCO test split used in the online test is not publicly available. The results are shown in Table 1.

From the test results, compared to X-Transformer, the second model in the ranking, our model has improved in most evaluation indicators.

Qualitative results and visualization. To make the comparison of model effects more intuitive, we showed four images of the test results. We compared the results with the results of different models.

GT1: a bottle of wine and a glass on a table.

GT2: a bottle of wine sitting on a table next to a glass.

Transformer: a glass and a bottle of wine on the table.

FPDT: a bottle of wine with a glass on a glass table.

GT1: a man is standing in the kitchen with a pan.

GT2: a man standing in a kitchen with pans.

Transformer: a man pointing at pans.

FPDT: a man standing in a kitchen with a bunch of pots and pans.

GT1: a cat standing on a bed.

GT2: a cat standing on a bed with a cup.

Transformer: a cat is standing on the food counter.

FPDT: a black and white cat standing on a bed next to a cup.

GT1: a woman with a child is carrying a toy car.

GT2: a woman holding a baby in a bathroom.

Transformer: a man and a woman pose for a picture.

FPDT: a woman photoing a baby brushing her teeth in the bathroom.

Figure 4. The visualization results compared with different models.

As we can see, our model is better than the other three models in terms of coverage of image content and the fine-grainedness of the image caption. For example, in the fourth image, FPDT gives a good description of the image content, such as photoing, brushing teeth, bathroom, etc. Other models don't do a very good job of describing it.

5. Conclusion

In this paper, we propose a new Transformer-based model, FPDT, to improve the model by fusing grid features and region features. It also achieves better decoding effect by improving the input information in the decoding phase which allows the model have more information about the test images. The final test results demonstrate that our proposed module can effectively improve the model's performance.

References

[1] Anderson, Peter, et al. "Bottom-up and top-down attention for image captioning and visual question answering." Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.

[2] Cornia, Marcella, Lorenzo Baraldi, and Rita Cucchiara. "Show, control and tell: A framework for generating controllable and grounded captions." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019.

[3] Junhua Mao, Wei Xu, Yi Y ang, Jiang Wang, and Alan LY uille. Explain images with multimodal recurrent neural networks. arXiv preprint arXiv:1410.1090, 2014. 1, 2

[4] Vinyals, Oriol, et al. "Show and tell: A neural image caption generator." Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.

[5] Vinyals, Oriol, et al. "Show and tell: Lessons learned from the 2015 mscoco image captioning challenge." IEEE transactions on pattern analysis and machine intelligence 39.4 (2016): 652-663.

[6] Xu, Kelvin, et al. "Show, attend and tell: Neural image caption generation with visual attention." International conference on machine learning. PMLR, 2015.

[7] Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems 30 (2017).

[8] Ren, Shaoqing, et al. "Faster r-cnn: Towards real-time object detection with region proposal networks." Advances in neural information processing systems 28 (2015).

[9] Jiang, Huaizu, et al. "In defense of grid features for visual question answering." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020.

[10] Lu, Jiasen, et al. "Knowing when to look: Adaptive attention via a visual sentinel for image captioning." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

[11] Chen, Long, et al. "Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

[12] Herdade, Simao, et al. "Image captioning: Transforming objects into words." Advances in Neural Information Processing Systems 32 (2019).

[13] Mao, Junhua, et al. "Deep captioning with multimodal recurrent neural networks (m-rnn)." arXiv preprint arXiv:1412.6632 (2014).

[14] Cornia, Marcella, et al. "Meshed-memory transformer for image captioning." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020.

[15] Zhang, Xuying, et al. "RSTNet: Captioning with adaptive attention on visual and non-visual words." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021.

[16] Rennie, Steven J., et al. "Self-critical sequence training for image captioning." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

[17] Jiang, Wenhao, et al. "Recurrent fusion network for image captioning." Proceedings of the European Conference on Computer Vision (ECCV). 2018.

[18] Yao, Ting, et al. "Exploring visual relationship for image captioning." Proceedings of the European conference on computer vision (ECCV). 2018.

[19] Yang, Xu, et al. "Auto-encoding scene graphs for image captioning." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019.

[20] Herdade, Simao, et al. "Image captioning: Transforming objects into words." Advances in Neural Information Processing Systems 32 (2019).

[21] Huang, Lun, et al. "Attention on attention for image captioning." Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019.

[22] Pan, Yingwei, et al. "X-linear attention networks for image captioning." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020.

[23] Cornia, Marcella, et al. "Meshed-memory transformer for image captioning." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020.