EGTR: Extracting Graph from Transformer for Scene Graph Generation

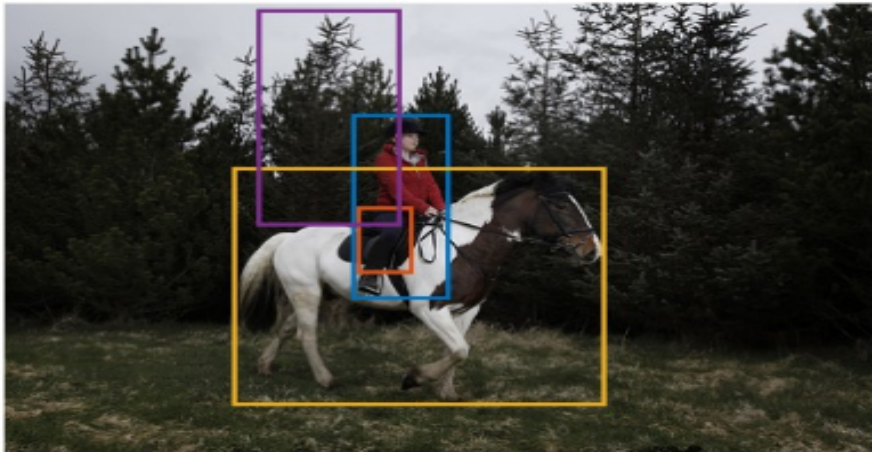
CVPR 2024

Jinbae Im, JeongYeon Nam, Nokyung Park, Hyungmin Lee, Seunghyun Park,

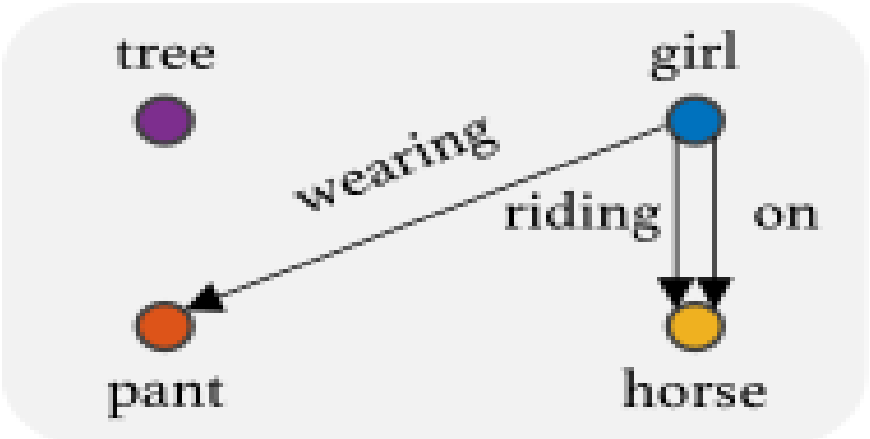
NAVER Cloud AI, NAVER, Korea University

October 26, 2025

What is a scene graph



What is a scene graph



Applications

Since a scene graph provides structural information of the image, it can be used for various vision tasks that require a higher level of understanding and reasoning about images

- Image Captioning
- Image Retrieval
- Visual Question Answering
- Conditional Image Generation

SGG Approaches

Two Main Approaches

- Two-Stage Methods
- One-Stage Methods
  - Object-Triplet Detection Models
  - Triplet Detection Models
  - Relation Extraction Models

Two-Stage Methods

1. Separate object detection model and relation prediction model are trained sequentially
2. They usually detect N objects from the off-the-shelf object detector such as Faster R-CNN
3. Then all possible combinations of the detected objects are fed into the relation prediction model to predict the relations between each object pair.

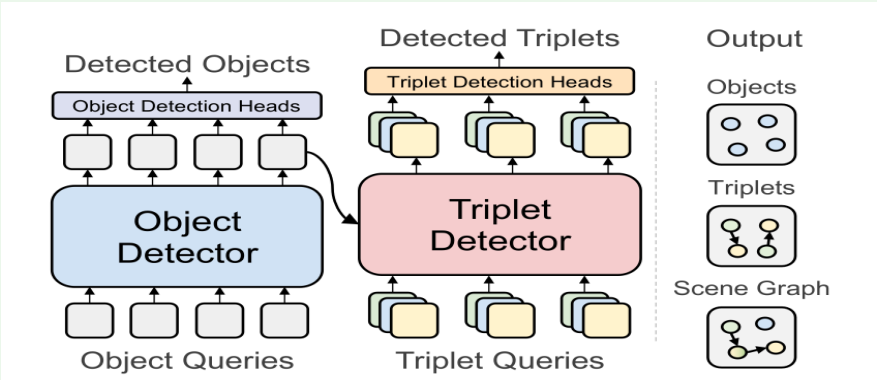
Shortcomings

They have the inherent limitation that the object detector that helps with relation extraction is trained separately, resulting in a significant increase in model complexity.

One-Stage Methods, Object-Triplet Detection Models

Object-triplet detection models are characterized by introducing additional triplet queries and building a triplet predictor on top of the object detector

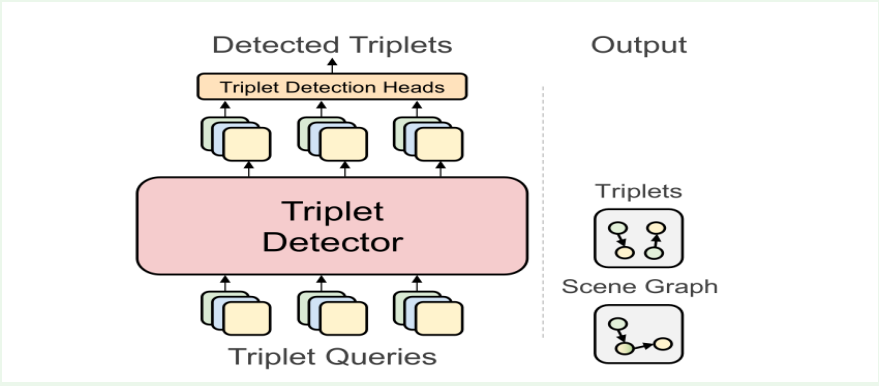
Example



One-Stage Methods, Triplet Detection Models

Triplet detection models directly detect triplets using triplet queries without an object detector

Example



One-Stage Methods, Triplet Detection Models

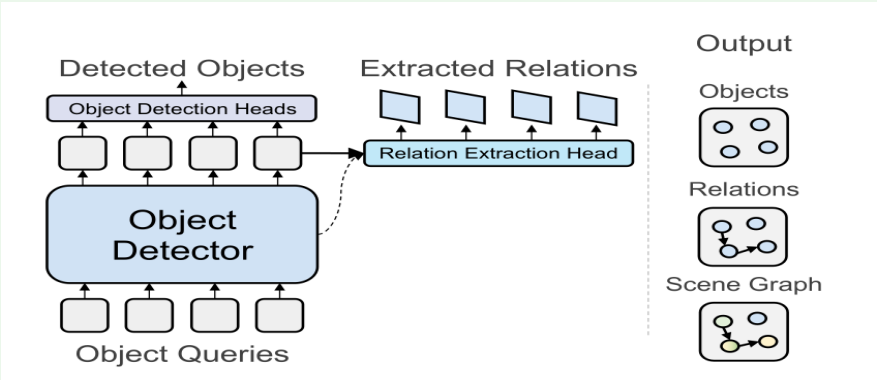
Shortcomings

- The model architecture became even more intricate to detect subjects and objects separately since an explicit object detector is not used.
- The triplet prediction models focus on detecting a sub-graph composed only of objects with relations, overlooking objects in an image that lack explicit relations.

One-Stage Methods, Relation Extraction Models

Relation extraction models extract a scene graph using a lightweight relation predictor without separate triplet queries or triplet detector

Example



Preliminary : DETR

Steps

1. Backbone : ResNet-50
   - Input : $x_{img} \in \mathbb{R}^{3 \times H \times W}$
   - Output : $F \in \mathbb{R}^{C \times H_F \times W_F}$
2. Transformer Encoder
   - 1×1 convolution applied to F to reduce dimension from C to d_{model}
   - Flatten to get sequence with length $H_F W_F$
3. Transformer Decoder
   - N object queries - Learnable and usually N is a large number to cover all objects inside the image
   - Alternating self-attention and cross attention layers
   - $Q_h^l \in \mathbb{R}^{N \times d_{head}}$, $K_h^l \in \mathbb{R}^{N \times d_{head}}$ are query and key respectively
   - $A_h^l = softmax(\frac{Q_h^l K_h^l}{\sqrt{d_{head}}}) \in \mathbb{R}^{N \times N}$

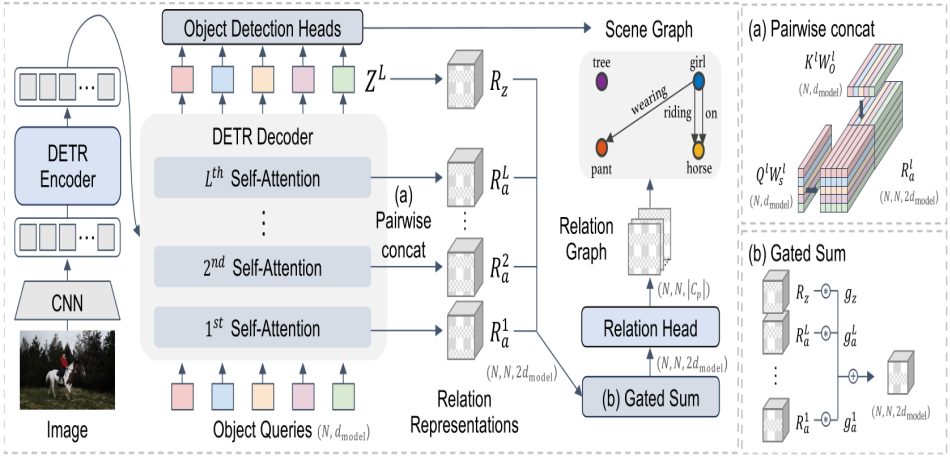
Preliminary : DETR

Steps

4. Detection Head : Three layer perceptron with ReLU To obtain bbox and category label
5. OD loss(hungaraian algorithm)

$$\mathcal{L}_{od} = \sum_{i=1}^{N} [\lambda_c \mathcal{L}_c(\hat{v}_i'.c, v_i.c) + \mathbb{I}_{v_i.c \neq \phi}(\lambda_b \mathcal{L}_b(\hat{v}_i'.b, v_i.b))]$$

EGTR


Object Detection Heads
Scene Graph
DETR Encoder
CNN
Image
DETR Decoder
L^{th} Self-Attention
2^{nd} Self-Attention
1^{st} Self-Attention
Object Queries (N, d_{model})
Z^L
R_z
R_a^L
R_a^2
R_a^1
(a) Pairwise concat
Relation Representations
$(N, N, 2d_{\text{model}})$
tree
girl
wearing
riding
on
pant
horse
Relation Graph
$(N, N, |C_p|)$
Relation Head
(b) Gated Sum
(a) Pairwise concat
$K^l W_o^l$
(N, d_{model})
$Q^l W_s^l$
R_a^l
$(N, N, 2d_{\text{model}})$
(b) Gated Sum
g_z
g_a^L
g_a^1

Extract the predicate information from the self-attention weights in the entire L layers, by considering the attention queries and keys as subjects and objects, respectively.

Pairwise concatenation

$$R_a^l = [Q^l W_S^l; K^l W_O^l]$$

Where : W_S^l and W_O^l are linear weights of shape $\mathbb{R}^{d_{model} \times d_{model}}$

Gating Mechanism

- Last layer representation of object queries : $Z^L \in \mathbb{R}^{N \times d_{model}}$
- $R_z = [Z^L W_S; Z^L W_O]$
- $g_a^l = \sigma(R_a^l W_G)$, $g_z = \sigma(R_z W_G)$ where g_z and $g_a^l \in \mathbb{R}^{N \times N \times 1}$ and $W_G \in R^{2d_{model} \times 1}$

Relation Extractor

$$\hat{G} = \sigma(MLP(\sum_{l=1}^{L}(g_a^l * R_a^l) + g_z * R_z))$$

Where $\hat{G} \in \mathbb{R}^{N \times N \times C}$ and MLP is a three-layer perceptron with ReLU activation

Training Loss

$$\mathcal{L} = \mathcal{L}_{od} + \lambda_{rel}\mathcal{L}_{rel} + \lambda_{con}\mathcal{L}_{con}$$

Adaptive Smoothing

For object candidate $\hat{v}_i'$, uncertainty is defined as follows:

$$u_i = \sigma(cost_i - cost_{min} + \sigma^{-1}(\alpha))$$

And:

$$G_{ijk} = (1 - u_i)(1 - u_j)$$

Results

	Model	# params (M)	FPS	AP50	R@20	R@50	R@100	mR@20	mR@50	mR@100
two-stage	IMP (EBM) [34, 42]	*322.2*	*2.0*	28.1	18.1	25.9	31.2	2.8	4.2	5.4
	VTransE [47]	*312.3*	*3.5*	-	24.5	31.3	35.5	5.1	6.8	8.0
	Motifs [45]	*369.9*	*1.9*	28.1	25.1	32.1	36.9	4.1	5.5	6.8
	VCTree [36]	*361.5*	*0.8*	28.1	24.8	31.8	36.1	4.9	6.6	7.7
	VCTree (TDE) [36, 37]	*361.3*	*0.8*	28.1	14.0	19.4	23.2	6.9	9.3	11.1
	VCTree (EBM) [34, 36]	*372.5*	-	28.1	24.2	31.4	35.9	5.7	7.7	9.1
	GPS-Net [20]	-	-	-	-	31.1	35.9	-	6.7	8.6
	BGNN [16]	341.9	*1.7*	29.0	23.3	31.0	35.8	7.5	10.7	12.6
one-stage	FCSGG [21]	87.1	*6.0*	28.5	16.1	21.3	25.1	2.7	3.6	4.2
	RelTR [7]	63.7	*13.4*	26.4	21.2	27.5	-	6.8	10.8	-
	SGTR [17]	*117.1*	*6.2*	25.4	-	24.6	28.4	-	12.0	15.2
	Relationformer [32]	92.9	*8.5*	26.3	22.2	28.4	31.3	4.6	9.3	10.7
	Iterative SGG [9]	*93.5*	*6.0*	*27.7*†	-	29.7	32.1	-	8.0	8.8
	SSR-CNN [38]	*274.3*	*4.0*	*23.8*†	**25.8**	**32.7**	**36.9**	6.1	8.4	10.0
	SSR-CNN [38] $_{\text{LA},\tau=0.3}$	*274.3*	*4.0*	*23.8*†	18.4	23.3	26.5	**13.5**	**17.9**	21.4
	EGTR (Ours)	**42.5**	**14.7**	**30.8**	23.5	30.2	34.3	5.5	7.9	10.1
	EGTR (Ours) $_{\text{LA},\tau=0.7}$	**42.5**	**14.7**	**30.8**	15.7	18.7	20.5	12.1	17.8	**21.7**
	EGTR (Ours) $_{\text{LA},\tau=0.5}$	**42.5**	**14.7**	**30.8**	19.7	24.2	26.7	11.0	17.1	21.4
	EGTR (Ours) $_{\text{LA},\tau=0.3}$	**42.5**	**14.7**	**30.8**	22.4	28.2	31.7	8.8	14.0	18.3