Collaborative Transformers for Grounded Situation Recognition
— Supplementary Material —

Junhyeong Cho[1] Youngseok Yoon[1] Suha Kwak[1,2]
Department of CSE, POSTECH[1] Graduate School of AI, POSTECH[2]
{junhyeong99, yys8646, suha.kwak}@postech.ac.kr

This supplementary material provides method details (Section A), implementation details (Section B), qualitative evaluations (Section C), an application of this task (Section D), computational evaluations (Section E) and a limitation (Section F), which could not be included in the main paper due to the limited space.

A. Method Details

Transformer architectures in our CoFormer consist of common building blocks, encoder and decoder. The main components of these building blocks are attention layers. Section A.1 provides more details of the attention layers.

In Section 3.5 of the main paper, the losses to train our model are described: verb classification loss, noun classification losses, box existence prediction loss, and box regression losses. Section A.2 provides more details of the losses.

A.1. Attention Layer

Multi-Head Attention. The input of the multi-head attention layer is the sequence of query, key and value. The query sequence is denoted by $\mathbf{Q} \in \mathbb{R}^{L_Q \times d}$, where L_Q is the sequence length and d is the size of the hidden dimension. The key sequence is denoted by $\mathbf{K} \in \mathbb{R}^{L_{KV} \times d}$ and value sequence is denoted by $\mathbf{V} \in \mathbb{R}^{L_{KV} \times d}$, where L_{KV} is the sequence length. In the multi-head attention layer, we employ H attention heads; the hidden dimension of each attention head is $d_h = d/H$. For each attention head i, $\mathbf{Q}$, $\mathbf{K}$ and $\mathbf{V}$ are linearly projected via parameter matrices $\mathbf{W}_i^Q, \mathbf{W}_i^K, \mathbf{W}_i^V \in \mathbb{R}^{d \times d_h}$. In details,

$$\mathbf{Q}_i = \mathbf{Q}\mathbf{W}_i^Q \in \mathbb{R}^{L_Q \times d_h}, \quad \text{(A.1)}$$

$$\mathbf{K}_i = \mathbf{K}\mathbf{W}_i^K \in \mathbb{R}^{L_{KV} \times d_h}, \quad \text{(A.2)}$$

$$\mathbf{V}_i = \mathbf{V}\mathbf{W}_i^V \in \mathbb{R}^{L_{KV} \times d_h}. \quad \text{(A.3)}$$

The output of each attention head i is obtained by a weighted summation of the value $\mathbf{V}_i$, where the weights are computed by the scaled dot-product between the query $\mathbf{Q}_i$ and the key $\mathbf{K}_i$ followed by a softmax function. In details,

$$\text{Attention}(\mathbf{Q}_i, \mathbf{K}_i, \mathbf{V}_i) = \text{Softmax}(\frac{\mathbf{Q}_i \mathbf{K}_i^T}{\sqrt{d_h}})\mathbf{V}_i. \quad \text{(A.4)}$$

The output of each attention head i is concatenated along hidden dimension, then linearly projected via a parameter matrix $\mathbf{W}^O \in \mathbb{R}^{d \times d}$. In details,

$$\text{MultiHead}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = [\text{Head}_1; ...; \text{Head}_H]\mathbf{W}^O, \quad \text{(A.5)}$$

where $[;]$ is a concatenation along hidden dimension and $\text{Head}_i = \text{Attention}(\mathbf{Q}_i, \mathbf{K}_i, \mathbf{V}_i)$ for $i = 1, ..., H$.

Multi-Head Cross-Attention. This is the multi-head attention layer where the key sequence $\mathbf{K}$ is same with the value sequence $\mathbf{V}$, but the query sequence $\mathbf{Q}$ is different.

Multi-Head Self-Attention. This is the multi-head attention layer where the query sequence $\mathbf{Q}$, key sequence $\mathbf{K}$, and value sequence $\mathbf{V}$ are same, *i.e.*, $\mathbf{Q} = \mathbf{K} = \mathbf{V}$.

A.2. Loss

Figure A1 shows the losses to train CoFormer. The verb classification loss is denoted by $\mathcal{L}_{\text{Verb}}$. The noun classification loss from the classifier involved in the decoder of Gaze-S1 transformer is denoted by $\mathcal{L}^1_{\text{Noun}}$, the loss from the classifier involved in the encoder of Gaze-S1 transformer is denoted by $\mathcal{L}^2_{\text{Noun}}$, and the loss from the classifier involved in the decoder of Gaze-S2 transformer is denoted by $\mathcal{L}^3_{\text{Noun}}$. The box existence prediction loss is denoted by $\mathcal{L}_{\text{BoxExist}}$. The $L1$ box regression loss is denoted by $\mathcal{L}_{L1}$. The GIoU box regression loss is denoted by $\mathcal{L}_{\text{GIoU}}$.

The total training loss is the linear combination of $\mathcal{L}_{\text{Verb}}, \mathcal{L}^1_{\text{Noun}}, \mathcal{L}^2_{\text{Noun}}, \mathcal{L}^3_{\text{Noun}}, \mathcal{L}_{\text{BoxExist}}, \mathcal{L}_{L1}$, and $\mathcal{L}_{\text{GIoU}}$. In this total loss, the loss coefficients are as follows: $\lambda_{\text{Verb}}, \lambda^1_{\text{Noun}}, \lambda^2_{\text{Noun}}, \lambda^3_{\text{Noun}}, \lambda_{\text{BoxExist}}, \lambda_{L1}, \lambda_{\text{GIoU}} > 0$.

B. Implementation Details

In Section 4.2 of the main paper, some implementation details are described. For completeness, we describe more architecture details (Section B.1), loss details (Section B.2), augmentation details (Section B.3), and training details (Section B.4) of our CoFormer.

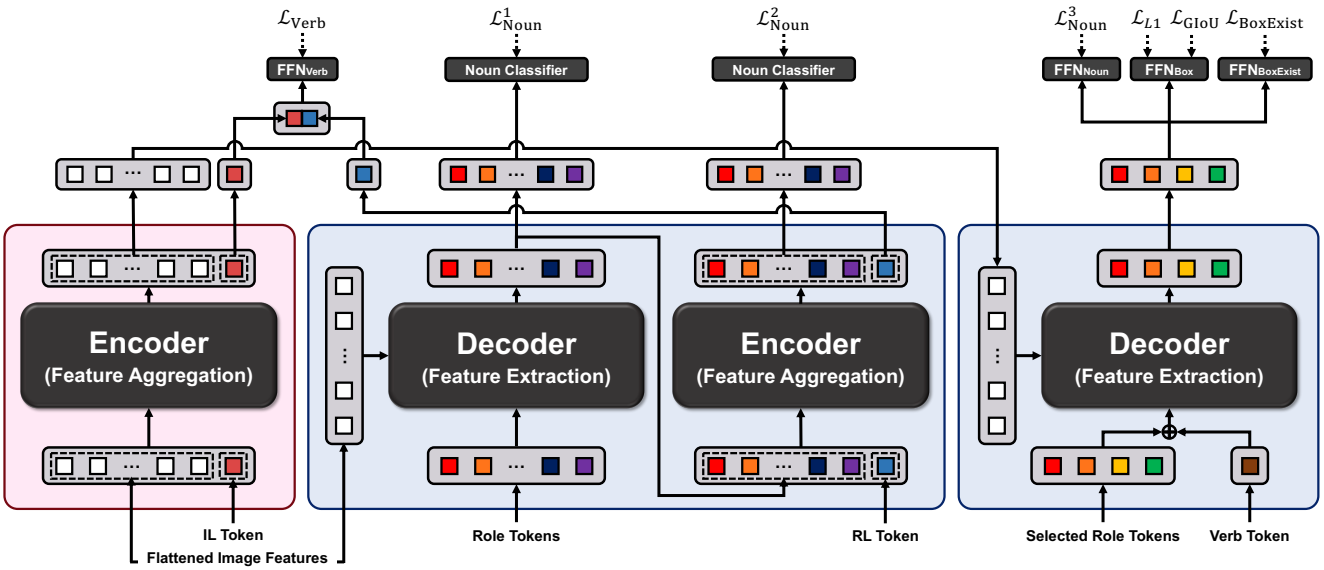


Figure A1. Transformer architectures in CoFormer including the losses to train our model. The losses for training our CoFormer are as follows: $\mathcal{L}_{\text{Verb}}, \mathcal{L}^1_{\text{Noun}}, \mathcal{L}^2_{\text{Noun}}, \mathcal{L}^3_{\text{Noun}}, \mathcal{L}_{\text{BoxExist}}, \mathcal{L}_{L1}, \mathcal{L}_{\text{GIoU}}$.

B.1. Architecture Details

Following previous work [1, 7], we use ResNet-50 [4] pretrained on ImageNet [2] as a CNN backbone. Given an image, the CNN backbone produces image features of size $h \times w \times c$, where $h, w = 22$ and $c = 2048$. A 1×1 convolution followed by a flatten operation produces flattened image features $\mathbf{X}_F \in \mathbb{R}^{hw \times d}$, where $d = 512$. To retain spatial information, we employ positional encodings. We use learnable 2D embeddings for the positional encodings.

We initialize encoders and decoders using Xavier Initialization [3], and these modules are trained with the dropout rate of 0.15. The number of heads in the attention layers of these modules is 8. Each of feed forward networks in these modules is 2-fully connected layers with a ReLU activation function, whose hidden dimensions are $4d$ and dropout rate is 0.15. These modules take learnable tokens, and each embedding dimension of the tokens is d.

The verb classifier FFN_{Verb} is 2-fully connected layers with a ReLU activation function, whose hidden dimensions are $2d$ and dropout rate is 0.3. Each of the two noun classifiers placed on top of Gaze-S1 transformer is a linear layer. The noun classifier FFN_{Noun} is 2-fully connected layers with a ReLU activation function, whose hidden dimensions are $2d$ and dropout rate is 0.3. The bounding box estimator FFN_{Box} is 3-fully connected layers with two ReLU activation functions, whose hidden dimensions are $2d$ and dropout rate is 0.2. The box existence predictor $\text{FFN}_{\text{BoxExist}}$ is 2-fully connected layers with a ReLU activation function, whose hidden dimensions are $2d$ and dropout rate is 0.2.

B.2. Loss Details

Complete Details of Noun Losses. In the SWiG dataset, each image is associated with three noun annotations given by three different annotators for each role. For the noun classification losses $\mathcal{L}^1_{\text{Noun}}$, $\mathcal{L}^2_{\text{Noun}}$, $\mathcal{L}^3_{\text{Noun}}$, each noun loss is obtained by the summation of three classification losses corresponding to three different annotators.

Regularization. We employ label smoothing regularization [8] in the loss computation for verb classification loss $\mathcal{L}_{\text{Verb}}$ and noun classification losses $\mathcal{L}^1_{\text{Noun}}$, $\mathcal{L}^2_{\text{Noun}}$, $\mathcal{L}^3_{\text{Noun}}$. In details, the label smoothing factor in the computation of verb classification loss is 0.3, and the factor in the computation of noun classification losses is 0.2.

Loss Coefficients. Total loss to train CoFormer is a linear combination of losses. In our implementation, the loss coefficients are $\lambda_{\text{Verb}} = \lambda^3_{\text{Noun}} = 1$, $\lambda^1_{\text{Noun}} = \lambda^2_{\text{Noun}} = 2$, and $\lambda_{\text{BoxExist}} = \lambda_{L1} = \lambda_{\text{GIoU}} = 5$.

B.3. Augmentation Details

For data augmentation, we employ random scaling, random horizontal flipping, random color jittering, and random gray scaling. The input images are randomly scaled with the scaling factors of 0.5, 0.75, and 1.0. Also, the input images are horizontally flipped with the probability of 0.5. The brightness, saturation and hue of the input images are randomly changed with the factor of 0.1 for each change. The input images are randomly converted to grayscale with the probability of 0.3.

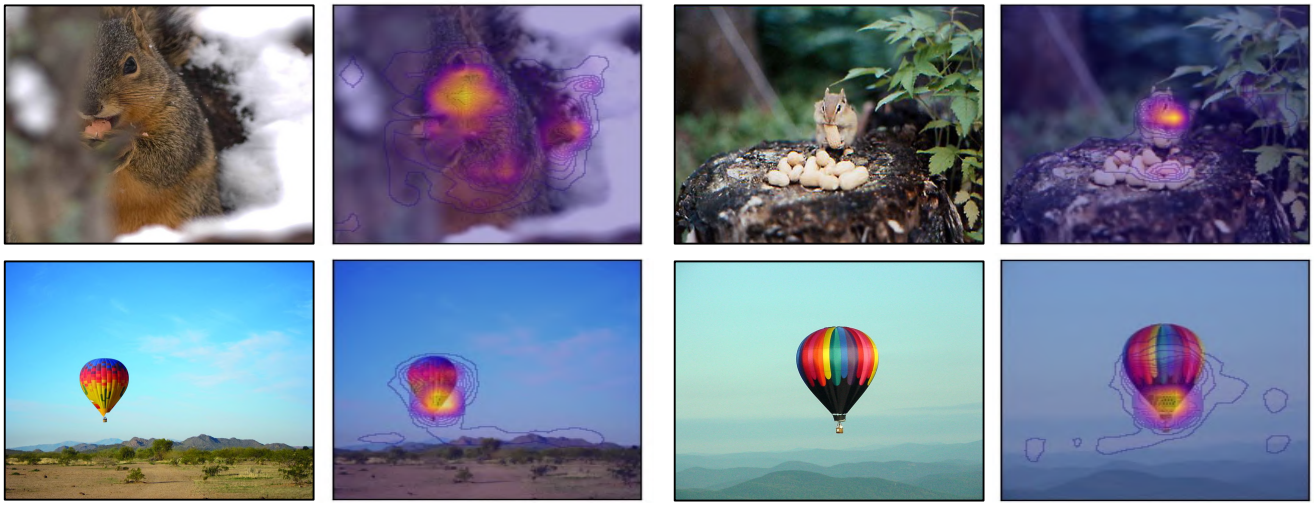


Figure C2. Attention scores from IL token to image features. We visualize the attention scores computed from the last self-attention layer of the encoder in Glance transformer. Higher attention scores are highlighted in red color on images.

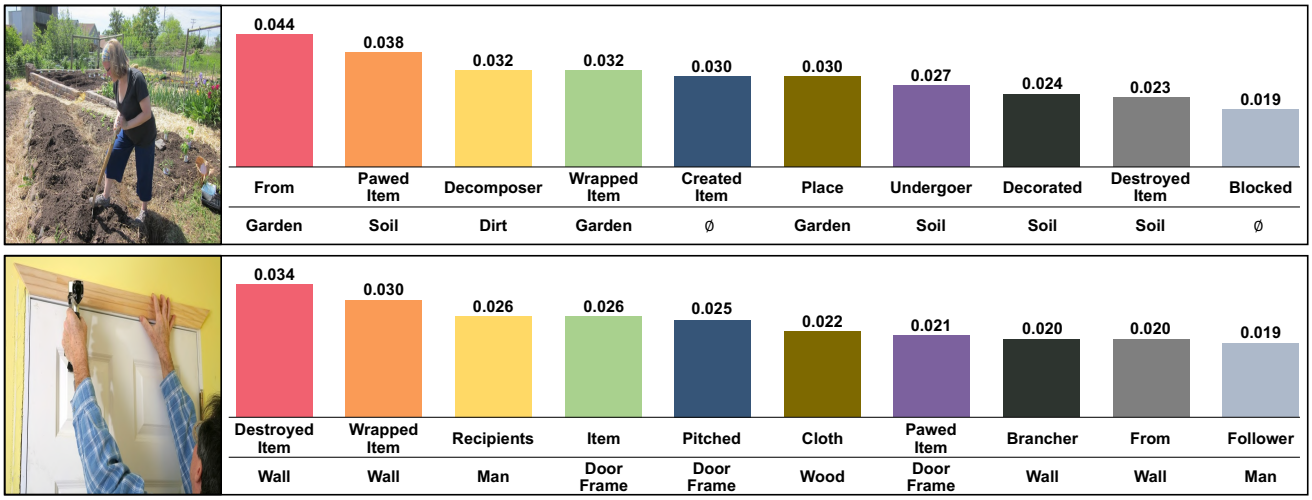


Figure C3. Attention scores from RL token to role features. We visualize the attention scores computed from the last self-attention layer of the encoder in Gaze-S1 transformer. Note that we show the roles where RL token has top-10 attentions scores. In Figure 7(b) of the main paper, we show the results corresponding to the roles in the frame of the ground-truth verb.

B.4. Training Details

We employ AdamW Optimizer [6] with the weight decay of 10^{-4}, $\beta_1 = 0.9$, and $\beta_2 = 0.999$. For stable training, we apply gradient clipping with the maximal gradient norm of 0.1. The transformers, classifiers and learnable embeddings are trained with the learning rate of 10^{-4}. The CNN backbone is fine-tuned with the learning rate of 10^{-5}. Note that we have a learning rate scheduler and the learning rates are divided by 10 at epoch 30. For batch training, we set the batch size to 16. We train CoFormer for 40 epochs, which takes about 30 hours on four RTX 3090 GPUs.

C. Qualitative Evaluations

We visualize the attention scores computed in the attention layers of the transformers in our CoFormer. Figure C2 shows that IL token captures the essential features to estimate the main activities for two *Cramming* images and two *Ballooning* images. Figure C3 shows the roles where RL token has top-10 attention scores, and the classification results from the noun classifier placed on top of the encoder in Gaze-S1 transformer; attention scores among 190 roles sum to 1. Note that several roles where RL token has high attention scores are not relevant to the main activity, but the noun classification results corresponding to those roles are

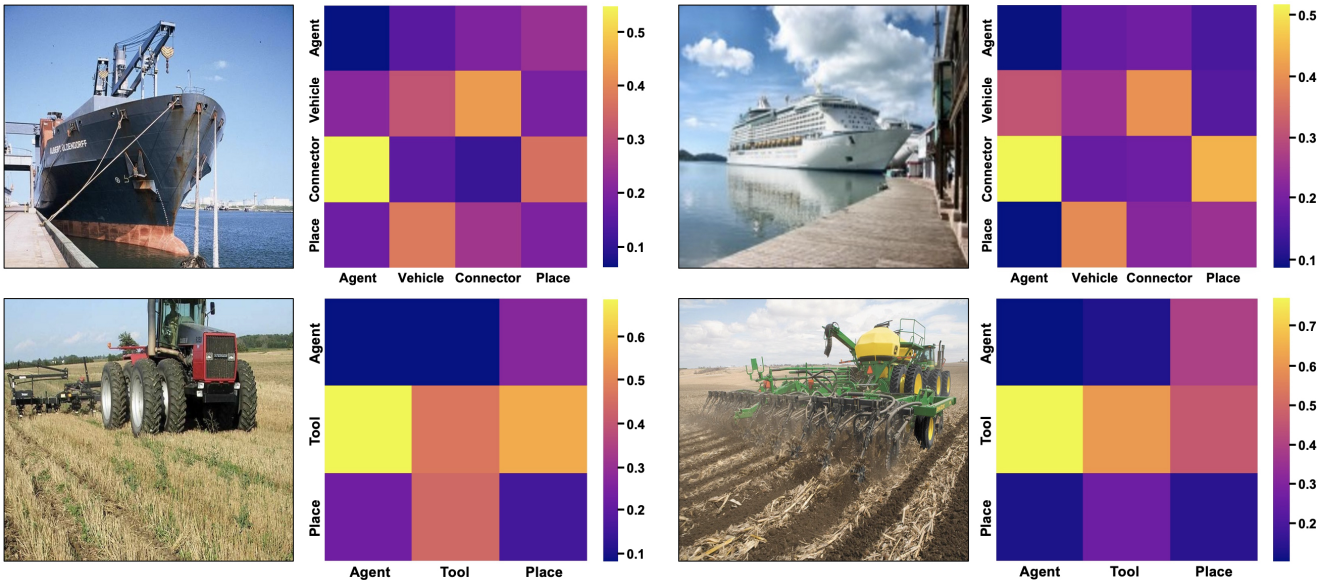


Figure C4. Attention scores on frame-role queries. We visualize the attention scores computed from the last self-attention layer of the decoder in Gaze-S2 transformer.

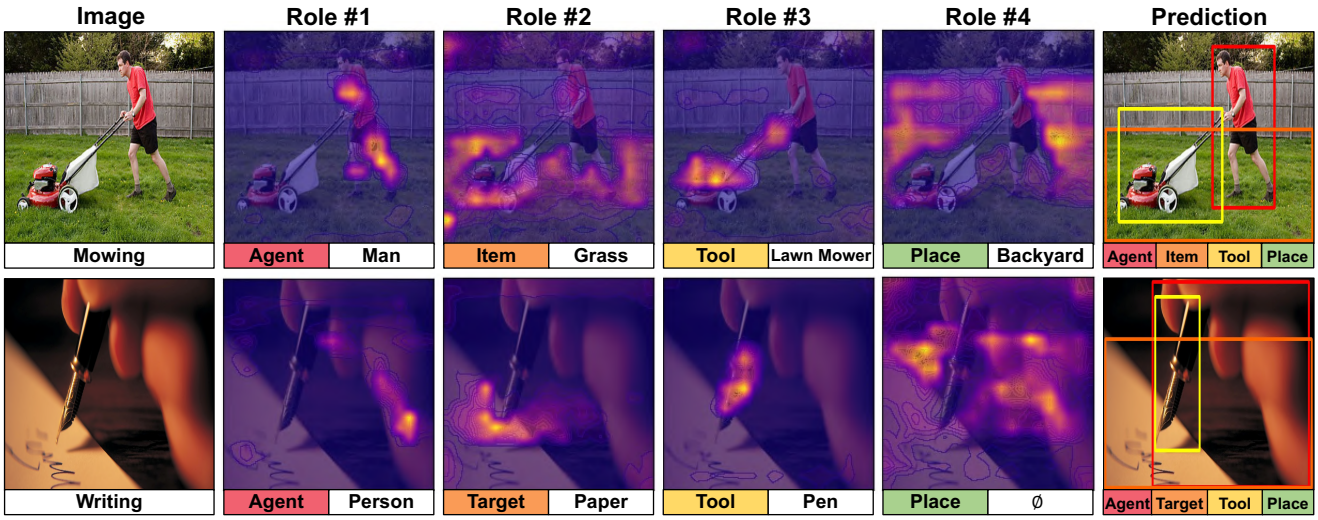
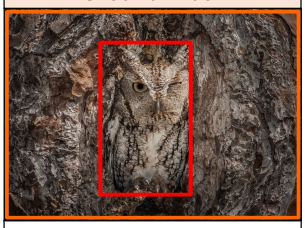


Figure C5. Attentions scores from frame-role queries to image features. We visualize the attention scores computed from the last cross-attention layer of the decoder in Gaze-S2 transformer. Higher attention scores are highlighted in red color on images.

highly relevant to the activity. Since RL token leverages the role features which are fed as input to the noun classifier, it is reasonable to aggregate those role features for accurate verb prediction; the role features are aware of involved nouns and their relations. Figure C3 demonstrates that RL token can effectively capture involved nouns and their relations for verb prediction through self-attentions on the role features in Gaze-S1 transformer. Figure C4 shows how role relations are captured through self-attentions on frame-role queries, which demonstrates that CoFormer similarly captures the relations if the situations in images are similar; attention scores sum to 1 in each column. Figure C6 shows the prediction results of CoFormer on the SWiG test set. The first and second row show correct prediction results. The third and fourth row show incorrect prediction results. As shown in Figure C6, three noun annotations are given for each role in the SWiG dataset. Note that the noun prediction is considered correct if the predicted noun matches any of the three noun annotations. The grounded noun prediction is considered correct if a noun, a bounding box, and box existence are correctly predicted for a role.

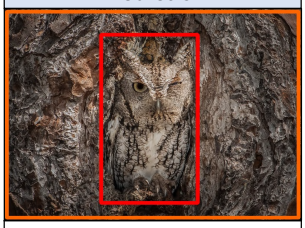
Ground-Truth



Camouflaging

Agent	Hiding Item	Place
Owl	Tree	Outdoors
Owl	Tree	Outdoors
Owl	Tree	Outdoors

Prediction



Camouflaging

Agent	Hiding Item	Place
Owl	Tree	Outdoors

Ground-Truth

Dusting

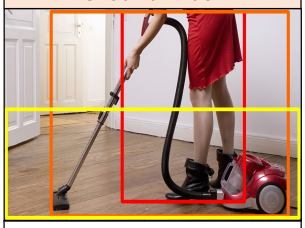
Agent	Source	Tool	Place
Woman	Table	Dust Cloth	Room
Person	Tabletop	Towel	House
Person	Table	Towel	Room

Prediction

Dusting

Agent	Source	Tool	Place
Person	Table	Dust Cloth	Room

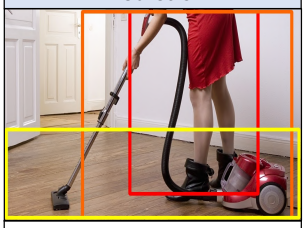
Ground-Truth



Vacuuming

Agent	Tool	Surface	Place
Female	Vacuum	Floor	Room
Woman	Vacuum	Floor	Inside
Woman	Vacuum	Floor	Room

Prediction



Vacuuming

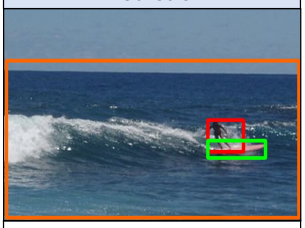
Agent	Tool	Surface	Place
Woman	Vacuum	Floor	Room

Ground-Truth

Surfing

Agent	Path	Place	Tool
Surfer	Water	Ocean	Surfboard
Person	Water	Ocean	Surfboard
Surfer	Wave	Ocean	Surfboard

Prediction



Surfing

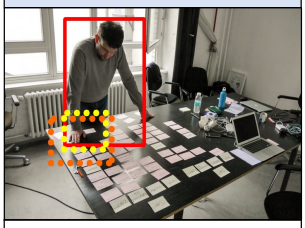
Agent	Path	Place	Tool
Surfer	Water	Ocean	Surfboard

Ground-Truth

Arranging

Agent	Item	Tool	Place
Man	Paper	Hand	Inside
Man	Paper	Hand	Office
Man	Card	∅	Table

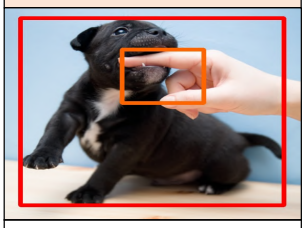
Prediction



Arranging

Agent	Item	Tool	Place
Man	Paper	Hand	Office

Ground-Truth



Nipping

Agent	Item	Place
Dog	Finger	∅
Puppy	Finger	Inside
Puppy	Finger	∅

Prediction

Nipping

Agent	Item	Place
Puppy	Finger	∅

Ground-Truth

Camping

Agent	Shelter	Place
Man	Tent	Mountain
Man	Tent	Hill
Man	Tent	Valley

Prediction

Camping

Agent	Shelter	Place
Man	Tent	Field

Ground-Truth

Picking

Agent	Crop	Source	Place
Female Child	Berry	Shrub	Outside
Child	Berry	Shrub	Outdoors
Girl	Berry	Leaf	Outdoors

Prediction

Picking

Agent	Crop	Source	Place
Girl	Berry	Plant	Outside

Figure C6. Prediction results of our CoFormer on the SWiG test set. Dashed boxes denote incorrect grounding predictions. Incorrect noun predictions are highlighted in gray color.

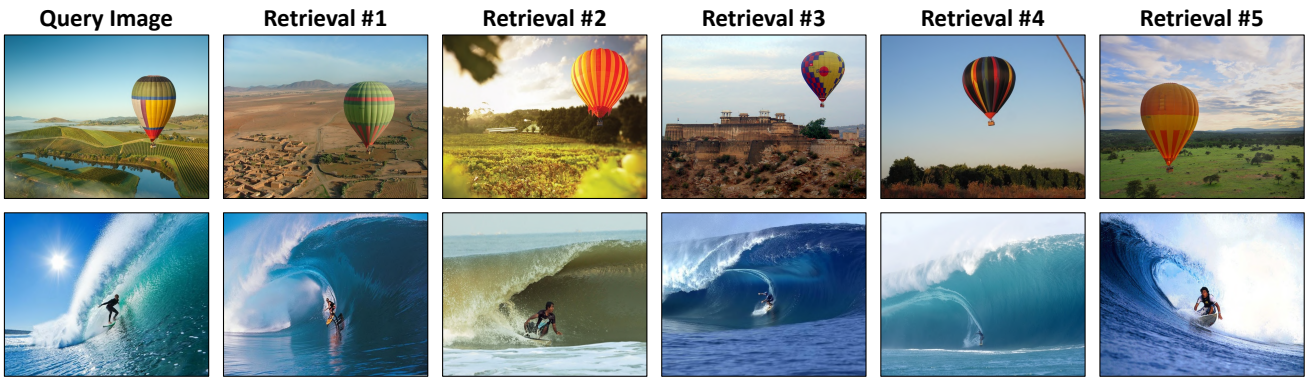


Figure D7. Grounded semantic aware image retrieval on the SWiG dev set. For each query image, we show the retrieval results which have top-5 similarity scores computed by $\mathrm{GrSitSim}(\cdot)$ [7]. This retrieval computes the similarity between two images considering the predicted verbs, nouns, and bounding-box groundings of the nouns.

$$\mathrm{GrSitSim}(I,J) = \max\left\{ \frac{\mathbb{1}_{[\hat{v}_i^I = \hat{v}_j^J]}}{i \cdot j \cdot |\mathcal{R}_{\hat{v}_i^I}|} \sum_{k=1}^{|\mathcal{R}_{\hat{v}_i^I}|} \mathbb{1}_{[\hat{n}_{i,k}^I = \hat{n}_{j,k}^J]} \cdot \left(1 + \mathrm{IoU}(\hat{\mathbf{b}'}_{i,k}^I, \hat{\mathbf{b}'}_{j,k}^J)\right) \middle| 1 \le i,j \le 5 \right\}. \tag{D.6}$$

D. Application

As shown in Figure D7, we can apply GSR models to grounded semantic aware image retrieval. This image retrieval computes the similarity between two images considering their grounded situations. In details, a similarity score between an image I and an image J is computed by $\mathrm{GrSitSim}(I,J)$ (Eq. D.6). Given an image I, a GSR model predicts the top-5 most probable verbs $\hat{v}_1^I, ..., \hat{v}_5^I$. For each predicted verb $\hat{v}_i^I$, the model predicts nouns $\hat{n}_{i,1}^I, ..., \hat{n}_{i,|\mathcal{R}_{\hat{v}_i^I}|}^I$ and bounding boxes $\hat{\mathbf{b}'}_{i,1}^I$, ..., $\hat{\mathbf{b}'}_{i,|\mathcal{R}_{\hat{v}_i^I}|}^I$. These prediction results are used in the computation of $\mathrm{GrSitSim}(I,J)$. By this score function, the similarity score is maximized if the top-1 predicted verb and the predicted grounded nouns are same for the two images I and J. Using this retrieval, we can retrieve images which have similar grounded situations with the situation of a query image.

E. Computational Evaluations

The number of parameters and inference time of our CoFormer are shown in Table E1. We also evaluate JSL [7] and GSRTR [1] on the SWiG test set using a single 2080Ti GPU with a batch size of 1. JSL uses two ResNet-50 [4] and a feature pyramid network (FPN) [5] in the CNN backbone, while GSRTR and our CoFormer only employ a single ResNet-50 in the backbone; these two models demand much shorter inference time than JSL, which is crucial for real-world applications. GSRTR and CoFormer are trained in an end-to-end manner, but JSL is trained separately in terms of verb model and grounded noun model.

Method	Backbone	#Params	Inference Time
JSL [7]	R50, R50-FPN	108 M	80.23 ms (12.46 FPS)
GSRTR [1]	R50	83 M	21.69 ms (46.10 FPS)
CoFormer (Ours)	R50	93 M	30.62 ms (32.66 FPS)

Table E1. Number of parameters and inference time. Inference time was measured on the SWiG test set using one 2080Ti GPU.

	Area (width × height)			Aspect Ratio (width/height)		
Metric	0-10%	10-20%	20-100%	0-5%	5-95%	95-100%
value	66.82	69.68	78.64	72.75	76.24	71.88
grnd value	7.42	25.38	65.49	36.88	62.62	31.01

Table F2. Quantitative analysis of our CoFormer on the SWiG dev set in Ground-Truth Verb evaluation setting. The effects of box scales and aspect ratios are evaluated. Each range denotes the ratio of ground-truth boxes when sorted by the value of area or aspect ratio in ascending order.

F. Limitation

As shown in Table F2, CoFormer suffers from estimating the noun labels and boxes for objects which have small scales (Area 0-10% and 10-20%) or extreme aspect ratios (Aspect Ratio 0-5% and 95-100%). To overcome such limitation, one may leverage multi-scale image features.

References

[1] Junhyeong Cho, Youngseok Yoon, Hyeonjun Lee, and Suha Kwak. Grounded Situation Recognition with Transformers. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2021. 2, 6

[2] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 2

[3] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 249–256, 2010. 2

[4] He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 2, 6

[5] Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature Pyramid Networks for Object Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2117–2125, 2017. 6

[6] Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations (ICLR)*, 2019. 3

[7] Sarah Pratt, Mark Yatskar, Luca Weihs, Ali Farhadi, and Aniruddha Kembhavi. Grounded Situation Recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 314–332, 2020. 2, 6

[8] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the Inception Architecture for Computer Vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826, 2016. 2