Open-Vocabulary Image Segmentation

Golnaz Ghiasi, Xiuye Gu, Yin Cui, Tsung-Yi Lin
Google Research
{golnazg,xiuyegu,yincui,tsungyi}@google.com

Arxiv, Dec 2021

Kunyang Han
2/19/2022

Open-Vocabulary

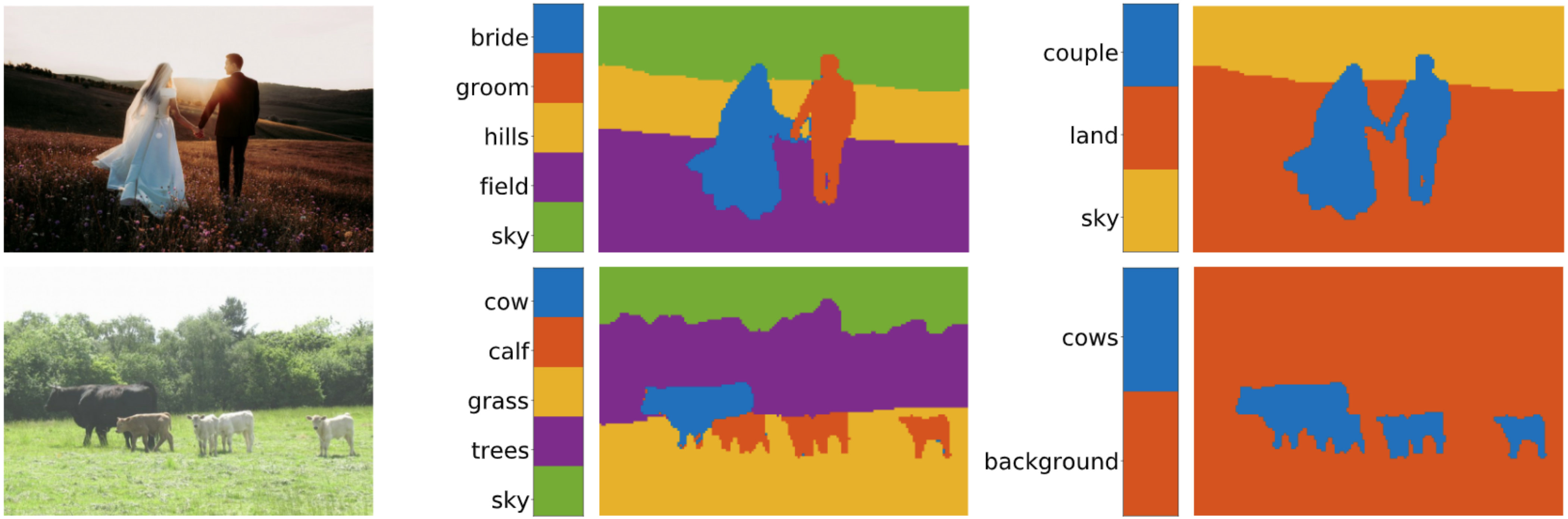
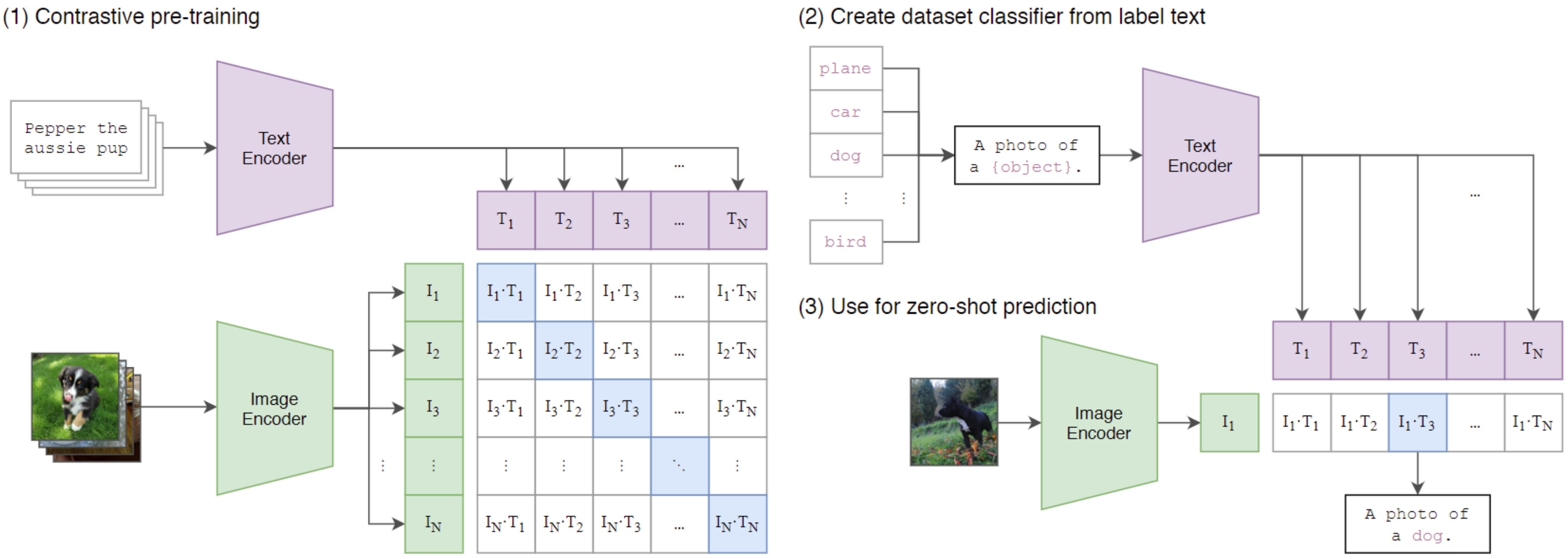
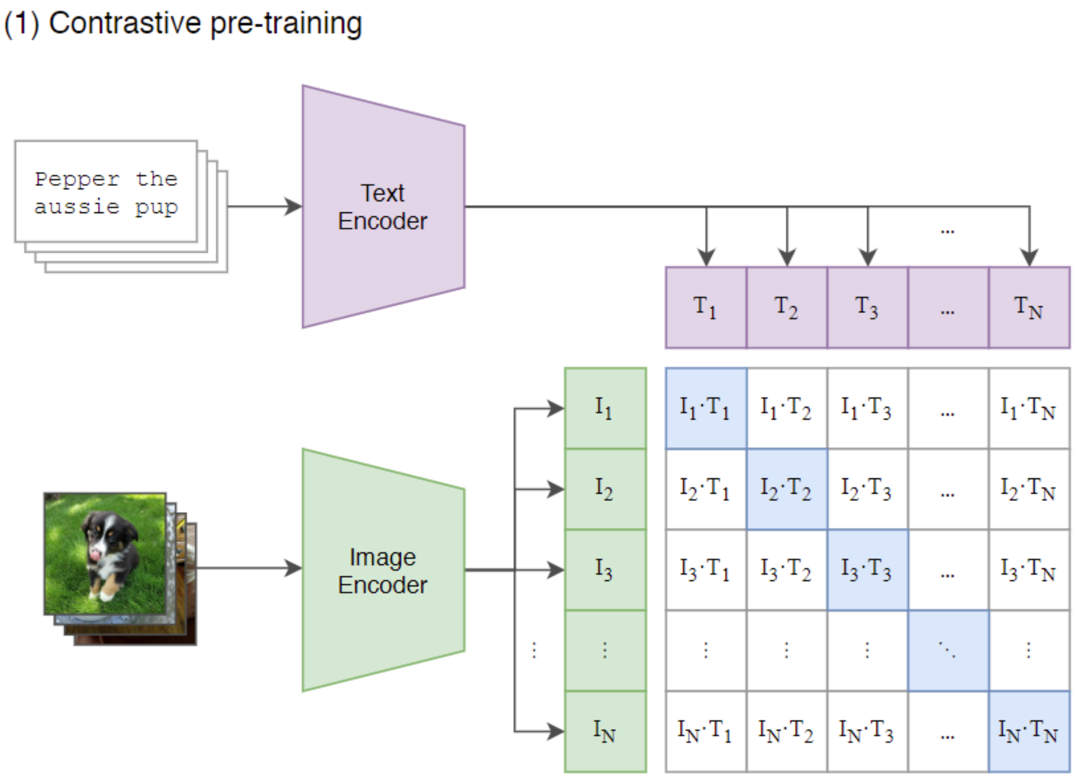


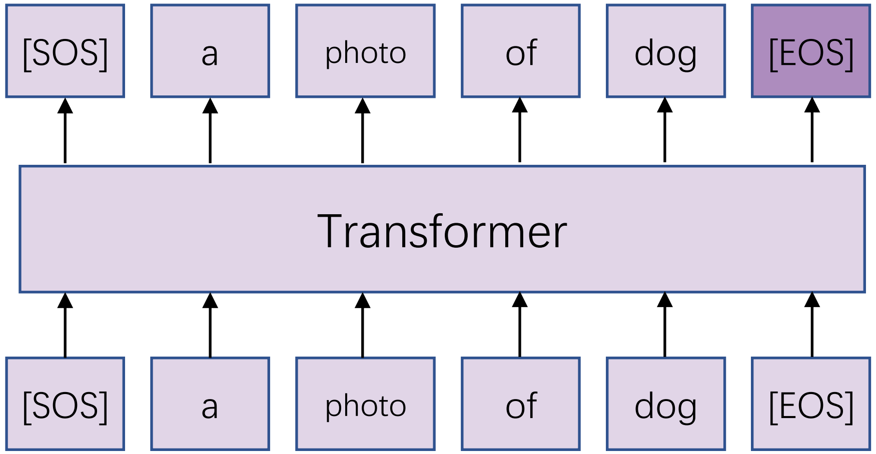
Figure 1. **Examples of image segmentation with arbitrary text queries.** We propose a model, called **OpenSeg**, that can organize pixels into meaningful regions indicated by texts. In contrast to segmentation models trained with close-vocabulary categories, OpenSeg can handle arbitrary text queries. For example, the model segments out a region for 'couple' and two regions for 'bride' and 'groom'.

CLIP: Connecting Text and Images

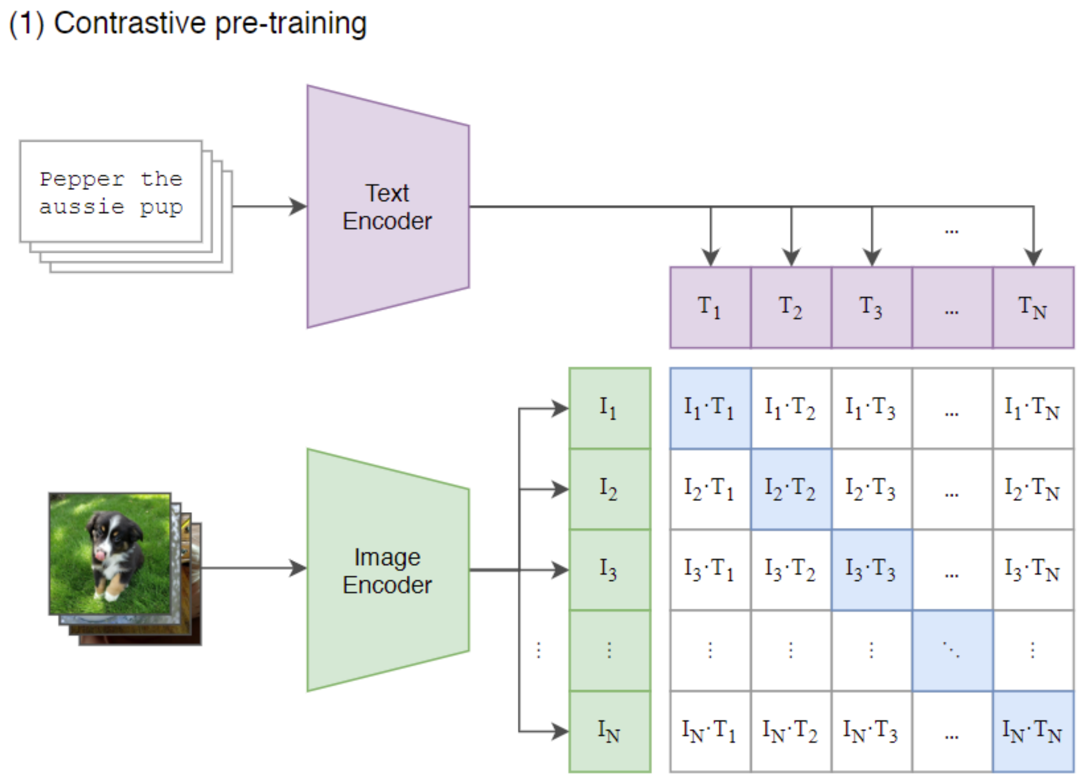


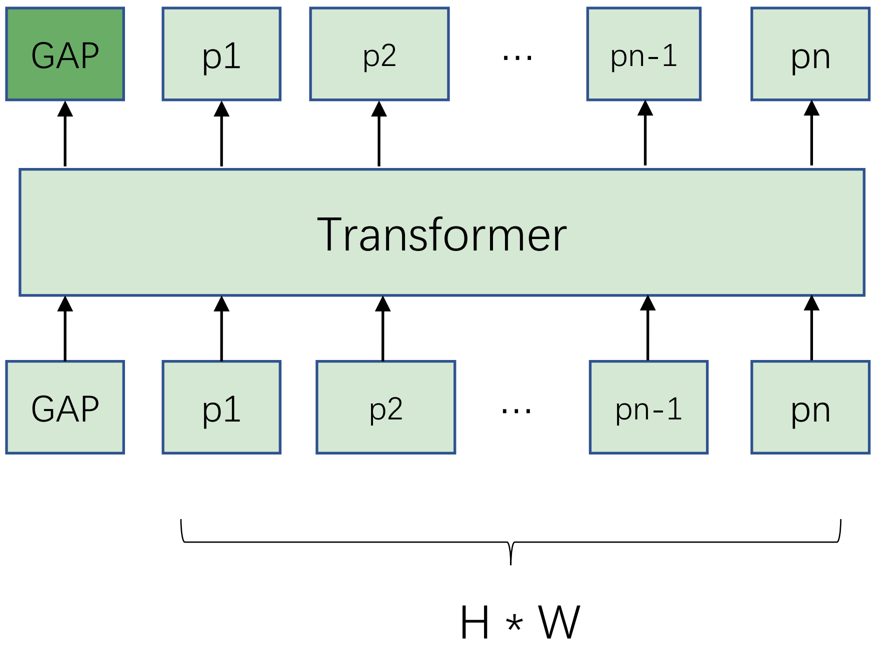
CLIP: Connecting Text and Images



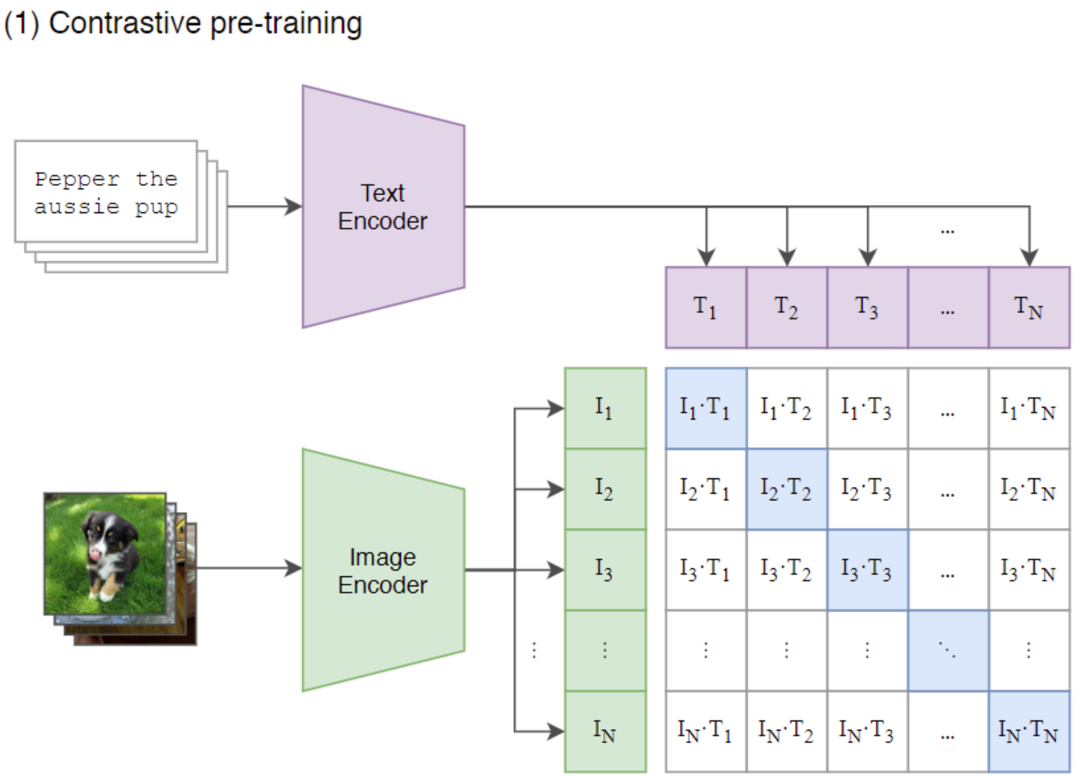


CLIP: Connecting Text and Images





CLIP: Connecting Text and Images




```
# image_encoder - ResNet or Vision Transformer
# text_encoder  - CBOW or Text Transformer
# I[n, h, w, c] - minibatch of aligned images
# T[n, l]       - minibatch of aligned texts
# W_i[d_i, d_e] - learned proj of image to embed
# W_t[d_t, d_e] - learned proj of text to embed
# t             - learned temperature parameter

# extract feature representations of each modality
I_f = image_encoder(I) #[n, d_i]
T_f = text_encoder(T)  #[n, d_t]

# joint multimodal embedding [n, d_e]
I_e = l2_normalize(np.dot(I_f, W_i), axis=1)
T_e = l2_normalize(np.dot(T_f, W_t), axis=1)

# scaled pairwise cosine similarities [n, n]
logits = np.dot(I_e, T_e.T) * np.exp(t)

# symmetric loss function
labels = np.arange(n)
loss_i = cross_entropy_loss(logits, labels, axis=0)
loss_t = cross_entropy_loss(logits, labels, axis=1)
loss   = (loss_i + loss_t)/2
```

CLIP: Connecting Text and Images

- Dataset
  - 400 million pairs (image, text)
  - Collected from Internet
- Training
  - 32 epochs
  - minibatch size of 32,768
  - 18 days on 592 V100 GPUs (RN50x64)
  - 12 days on 256 V100 GPUs (ViT-L/14)

ALIGN:A Large-scale ImaGe and Noisy-text embedding

- Leverage a noisy dataset of 1.8 billion image-text pairs
- Scale of our corpus can make up for its noise and leads to SOTA

Figure 2. Example image-text pairs randomly sampled from the training dataset of ALIGN. One clearly noisy text annotation is marked in *italics*.

		Flickr30K (1K test set)					
		image → text			text → image		
		R@1	R@5	R@10	R@1	R@5	R@10
Zero-shot	ImageBERT	70.7	90.2	94.0	54.3	79.6	87.5
	UNITER	83.6	95.7	97.7	68.7	89.2	93.9
	CLIP	88.0	98.7	99.4	68.7	90.6	95.2
	ALIGN	**88.6**	**98.7**	**99.7**	**75.7**	**93.8**	**96.8**
Fine-tuned	GPO	88.7	98.9	99.8	76.1	94.5	97.1
	UNITER	87.3	98.0	99.2	75.6	94.1	96.8
	ERNIE-ViL	88.1	98.0	99.2	76.7	93.6	96.4
	VILLA	87.9	97.5	98.8	76.3	94.2	96.8
	Oscar	-	-	-	-	-	-
	ALIGN	**95.3**	**99.8**	**100.0**	**84.9**	**97.4**	**98.6**

ALIGN

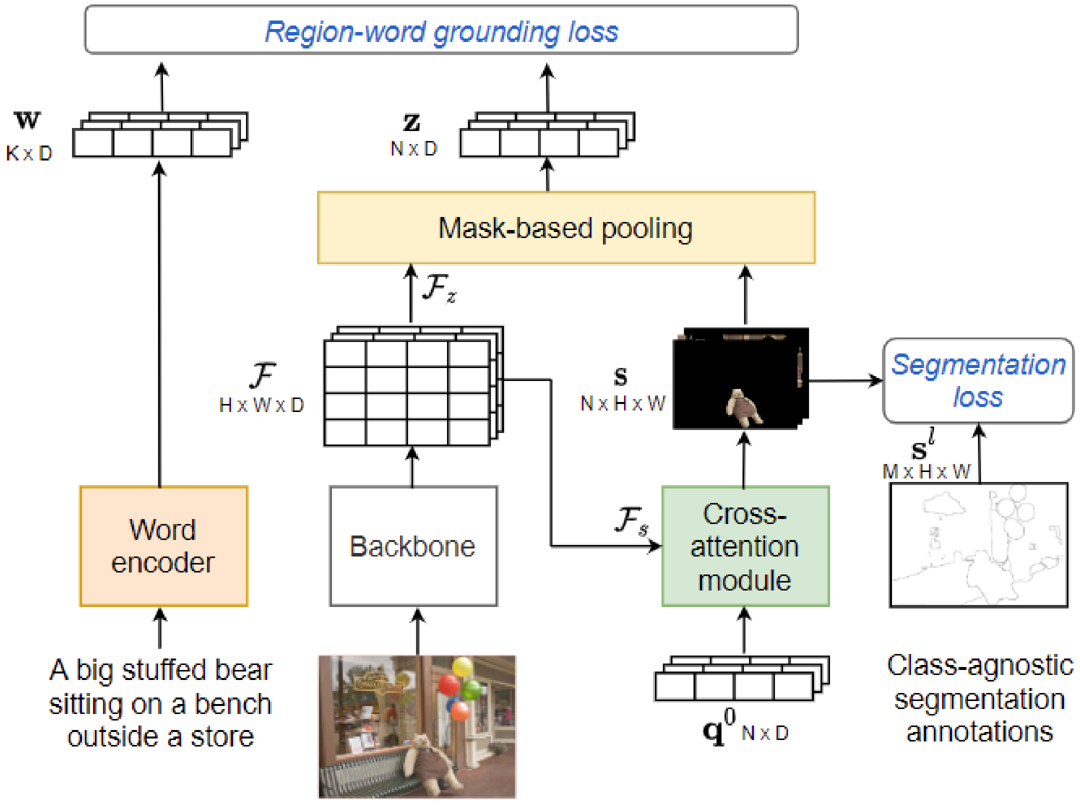
Table 1. Image-text retrieval results on Flickr30K and MSCOCO datasets (zero-shot and fine-tuned). ALIGN is compared with ImageBERT (Qi et al., 2020), UNITER (Chen et al., 2020c), CLIP (Radford et al., 2021), GPO (Chen et al., 2020a), ERNIE-ViL (Yu et al., 2020), VILLA (Gan et al., 2020), and Oscar (Li et al., 2020).

		Flickr30K (1K test set)						MSCOCO (5K test set)					
		image → text			text → image			image → text			text → image		
		R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
Zero-shot	ImageBERT	70.7	90.2	94.0	54.3	79.6	87.5	44.0	71.2	80.4	32.3	59.0	70.2
	UNITER	83.6	95.7	97.7	68.7	89.2	93.9	-	-	-	-	-	-
	CLIP	88.0	98.7	99.4	68.7	90.6	95.2	58.4	81.5	88.1	37.8	62.4	72.2
	ALIGN	**88.6**	**98.7**	**99.7**	**75.7**	**93.8**	**96.8**	**58.6**	**83.0**	**89.7**	**45.6**	**69.8**	**78.6**
Fine-tuned	GPO	88.7	98.9	99.8	76.1	94.5	97.1	68.1	90.2	-	52.7	80.2	-
	UNITER	87.3	98.0	99.2	75.6	94.1	96.8	65.7	88.6	93.8	52.9	79.9	88.0
	ERNIE-ViL	88.1	98.0	99.2	76.7	93.6	96.4	-	-	-	-	-	-
	VILLA	87.9	97.5	98.8	76.3	94.2	96.8	-	-	-	-	-	-
	Oscar	-	-	-	-	-	-	73.5	92.2	96.0	57.5	82.8	**89.8**
	ALIGN	**95.3**	**99.8**	**100.0**	**84.9**	**97.4**	**98.6**	**77.0**	**93.5**	**96.9**	**59.9**	**83.3**	**89.8**

Open-vocabulary Image Segmentation

- N object masks → N vision embed
- M words → M text embed

- Similarity matrix, (N, M)



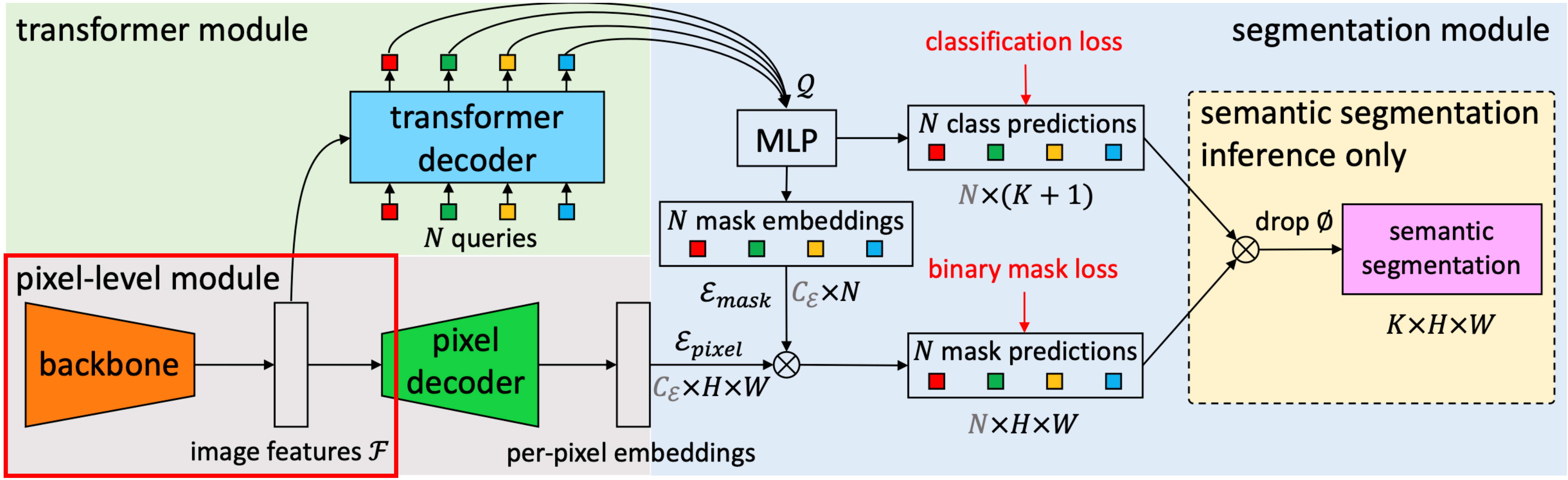
(c) OpenSeg (ours)

Dataset

- COCO Panoptic
- COCO Caption

a group of people riding horses down a small road.
a group of people are riding horses at a park.
people are riding their horses in the parade.
a group of riders on horses in a field.
a large crowd of people riding horses walks along a trail.

Vision Embed- Segmentation mask



Vision Embed- Mask based pooling

Pool(Feature [C, H, W] * Mask [H, W]) [C]

7.56	8.23	6.19	8.31
6.95	9.13	9.27	1.89
0.06	6.61	6.59	5.13
5.97	9.29	7.96	2.82

C_1

*

0	0	0	0
0	0	1	1
0	1	1	1
0	0	0	0

Mask

=

0	0	0	0
0	0	9.27	1.89
0	6.61	6.59	5.13
0	0	0	0

Mask_C_1

Vector_1 = (9.27 + 1.89 + 6.61 + 6.59 + 5.13) / 5

Vision Embed

1. N maskformer query
2. Maskformer predict
   - N object mask. (Bsz, N, H, W)
3. Mask based pooling
   - N vision embedding. (Bsz, N, dim)

Text Embed

1. Image Caption.
    - “a group of people riding horses down a small road.”. str
2. Extract Noun.
    - [‘people’, ‘horses’, ‘road’]. List[str]
3. Tokenize.
    - Token. List[List[Float]] (num_word, context_length)
4. Forward, ALIGN text encoder.
    - Embedding. Tensor. (Bsz, M, dim)

Grounding Loss - Similarity

- Similarity $\langle z_i, w_j \rangle = \frac{z_i \cdot w_j}{\|z_i\| \|w_j\|}$
- All regions to one word $g(\mathbf{z}, w_j) = [\langle z_1, w_j \rangle, ..., \langle z_N, w_j \rangle] \in \mathbb{R}^{N \times 1}$
- Softmax at i-th element $\sigma(\mathbf{x})_i = \frac{e^{x_i/\tau}}{\sum_j e^{x_j/\tau}}$
- Image-Caption score

$$G(I_b, C_b) = \frac{1}{K} \sum_{j=1}^{K} \sum_{i=1}^{N} \underbrace{\sigma(g(\mathbf{z}, w_j))_i}_{\text{weight}} \cdot \underbrace{\langle z_i, w_j \rangle}_{\text{similarity}}$$

How j similar with i, compared with others

Grounding Loss

- Image-Caption score
  - Encourages each word to be grounded to one or a few region
- All images in a batch I to a caption C_b

$$G(\mathbf{I}, C_b) = [G(I_1, C_b), ..., G(I_{|B|}, C_b)] \in \mathbb{R}^{|B| \times 1}$$

- Caption to I_b $\quad G(I_b, \mathbf{C}) = [G(I_b, C_1), ..., G(I_b, C_{|B|})] \in \mathbb{R}^{|B| \times 1}$
- Grounding Loss:

$$\mathcal{L}_{\mathcal{G}} = -\frac{1}{|B|} \sum_{b=1}^{|B|} \Big(\log \sigma \big(G(\mathbf{I}, C_b)\big)_b + \log \sigma \big(G(I_b, \mathbf{C})\big)_b \Big)$$

Scale up the training data

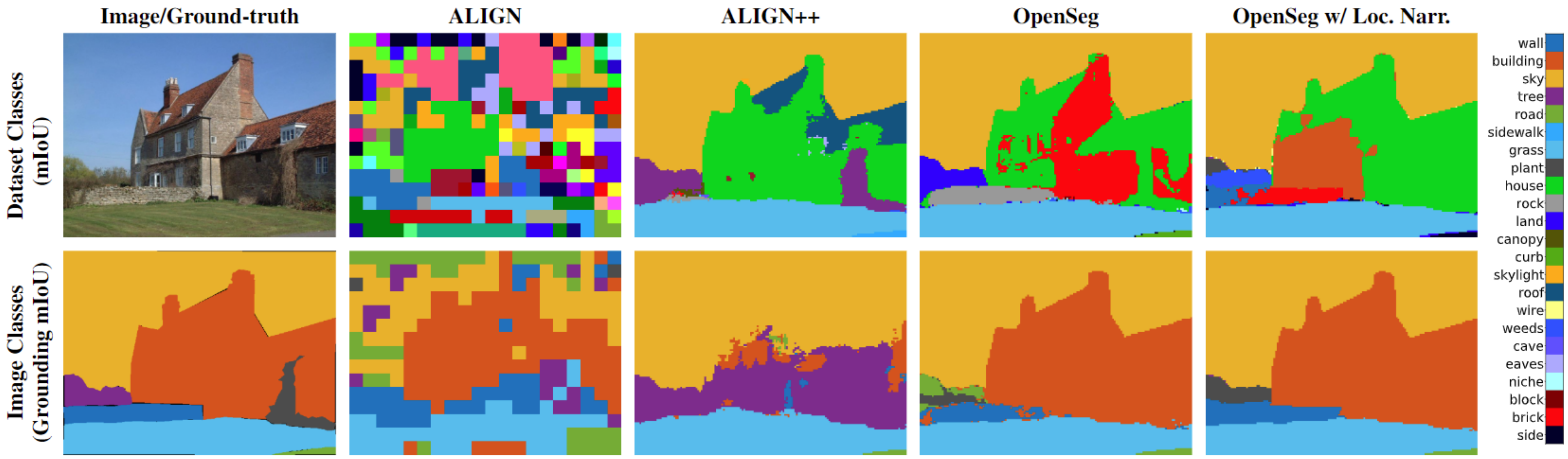
1. Train a teacher model on a segmentation dataset
2. Annotate with pseudo segmentation labels
3. Train with the mix of human and pseudo labels

- COCO
- Localized Narrative.
  - Contains detailed natural language descriptions along with mouse traces.
  - COCO, Flickr, Open Images, ADE20k.
  - 652k images for training

Training

- Batch size
  - 1024
- Max iter
  - 30k for COCO
  - 60k for COCO and Loc. Narr.
- Loss weight
  - 4 : 1, segmentation loss : grounding loss
- Keep probability of words extracted from captions
  - 0.75

Result



	COCO Train Data			mIoU					Grounding mIoU				
	label	mask	cap.	A-847	PC-459	A-150	PC-59	COCO	A-847	PC-459	A-150	PC-59	COCO
ALIGN	✗	✗	✗	4.1	3.7	10.7	15.7	20.0	18.2	22.5	28.0	37.6	28.2
ALIGN++	✓	✓	✗	3.8	7.8	18.0	**46.5**	55.1	10.5	17.1	30.8	56.7	60.8
OpenSeg	✗	✓	✓	6.3	9.0	21.1	42.1	36.1	21.8	32.1	41.0	57.2	48.2
OpenSeg w/ Loc. Narr.	✗	✓	✓	**6.8**	**11.2**	**24.8**	45.9	38.1	**25.4**	**39.0**	**45.5**	**61.5**	48.2

ALIGN++, add FPN, high resolution, per-pixel supervision.

Grounding mIoU, only uses the ground-truth classes in an image.

Ablation Study

	A-847	PC-459	A-150	PC-59
OpenSeg	6.3	9.0	21.1	42.1
- pred. masks	(-1.7) 4.6	(-3.1) 5.9	(-4.7) 16.4	(-10.0) 32.1
+ gt. masks	(+2.8) 9.1	(+3.3) 12.3	(+6.4) 27.5	(+7.2) 49.3

Table 4. **Incorporating predicted masks at inference improves mIoU accuracy.** Using the ground-truth masks can be seen as the performance upper bound when segmentation masks are perfectly predicted. The model is trained on COCO.

caption filter	A-847	PC-459	A-150	PC-59
all words	5.3	8.8	20.0	41.3
noun + adj. + verb	6.0	8.8	20.9	41.7
noun	**6.3**	**9.0**	**21.1**	**42.1**

Table 5. **Using all words in training captions hurts performance.** We show the mIoU performances with different text filtering to break a training caption into words. Using nouns only for training achieves the best results. The model is trained on COCO.