Hard Negative Sampling Strategy for Referring Image Segmentation

---

Multi-View Cluster Generation

Key idea: Given an image and a short, under-informative referring expression (e.g., "man" or "lady"), augment the expressions by describing the object from multiple semantic perspectives (e.g., appearance, spatial location, action).

We introduce a new concept "*cluster*", which is the group of text embeddings for each target instance. We assume the cluster represents a comprehensive semantic boundary such that, if a new expression's embedding falls outside this cluster, it's treated as a negative sample.

Example: (ref_coco)

Original text descriptions:

```
[[b'left', b'man', b'Man'], [b'lady', b'Woman on right', b'woman']]
```

Expected text descriptions after *multi-view cluster generation* (for the man):

- **Appearance:** the young man with glasses and a white shirt
- **Action:** the man looking at the woman
- **Spatial:** the man on the left, facing the woman

Input

An image with raw referring expressions

Output

An updated cluster for each instance in the image. Each cluster contains 3 diverse referring expressions from 3 perspectives (another expression from the perspective "Appearance" is generated for the case where not all perspectives are applicable)

Prompt

```
You will be shown an image and a list of lists of short
referring expressions that identifies a specific instance,
for each instance in the image.

Your task is to generate 3 diverse referring expressions
for each instance, based on the following perspectives:
1.  Appearance (e.g., color, shape, texture)
2.  Action (e.g., doing something or interacting)
3.  Spatial (e.g., position or relation to
others/background)

Note:
- If a perspective is not applicable (e.g., a tree is
static and has no action), provide a second description
from another applicable view.

Example 1:
Image Description:  "A woman is adjusting a man's tie, the
man is on the left side of the image."
Original Referring Expression:  "woman"
Multi-view expressions:
- Appearance:  "the woman with short gray hair and glasses"
- Action:  "the woman adjusting the man's tie"
- Spatial:  "the woman on the right side of the image"

Example 2:
Image Description:  "A large tree standing alone in a
field, the tree is tall with green leaves"
Original Referring Expression:  "tree"
Multi-view expressions:
- Appearance:  "the tall tree with a thick trunk"
- Action:  (none - object is static)
- Spatial:  "the tree in the center of the field"
- Alternative from "Appearance":  "the tree with green
leaves"

Example 3:
Image Description:  "A red car parked beside a black SUV."
Original Referring Expression:  "red car"
Multi-view expressions:
- Appearance:  "the shiny red car with silver rims"
- Action:  (none - car is parked)
- Spatial:  "the car next to the black SUV"
- Alternative from "Appearance":  "the car that is smaller
between the two"
```

Metric for Negative Classification

During training, need to ensure that the embeddings from the same cluster lie closely so that the cluster won't cover a large space

- **Method 1:** Compute the mean

$$\mu = \frac{1}{3}\sum_{i=1}^{3} e_i$$

  where e_i denotes the embedding of multi-view expressions. For a new expression e, compute the cosine similarity:

$$\text{sim}(e, \mu) = \frac{e \cdot \mu}{\|e\| \cdot \|\mu\|}$$

  and treat it as negative if the score lies below a threshold.

- **Method 2:** *(To be defined)*

Hard Negative Sampling

Key idea: Use expressions from different but semantically similar instances to push the model to learn fine-grained visual-textual discrimination.

1. Image-Text Matching

1. **Texts describing different instances from the same image**: Given an image with its corresponding referring expressions (several for each instance), choose one text as positive, then sample texts that are not from the same list of the "positive" text (i.e., not describing the same instance) as negative.

2. **Similar texts from different images**:

   - **Method 1:** Within each cluster of text embeddings, use the classification metric to identify referring expressions *outside* the cluster that are *closest* in embedding space. These are treated as hard negatives due to embedding-level similarity but differing semantics.
   - **Method 2 (with preprocessing):** First, use a large language model (LLM) to extract the object class (e.g., one of the 80 COCO categories) for each referring expression. Then apply Method 1, restricting the pool to expressions of the *same object class*.
     Note: Method 1 relies solely on embedding similarity, which may not reflect true semantics. Explicit class labels in Method 2 yield more robust hard negatives.

2. Text Augmentation

Use LLM-based augmentation of referring expressions to generate semantic distractors.

- **Attribute Substitution** (Replace color, spatial keyword, count)
  e.g., "the man in red" → "the man in blue"
- **Negation**
  e.g., "the man holding an umbrella" → "the man not holding an umbrella"
- **Role Reversal** (Swap subject-object)
  e.g., "the cat chasing the dog" → "the dog chasing the cat"

Implementation

We provide further details of implementation here.

Multi-view Expression Generation

The original referring expressions are kept under `external/refer/refcoco/`. We load the file and get a list of referring object,

We feed the image and the original referring expressions into an LLM. Here we choose `llava-hf/llava-1.5-7b-hf` as our model.

We ask the LLM to generate 3 expressions, each from a unique perspective: appearance, action and spatial.

After multi-view expression generation, we save our expressions back to `external/refer/refcoco/`.

Hard Negative Sampling

We first implement the first part, i.e., taking expressions of different objects from the same image as negative examples.

Take SaG as example. The original loss (C^3) is defined as

$$\mathcal{L}_{C^3} = -\frac{1}{B}\sum_{j=1}^{B}\log\frac{e^{\langle \mathbf{z}_{jj}, \mathrm{sg}(\mathbf{x}_j^{\mathcal{T}})\rangle}}{\sum_{i=1}^{B} e^{\langle \mathbf{z}_{ij}, \mathrm{sg}(\mathbf{x}_j^{\mathcal{T}})\rangle}}$$

In the hard-negative sampling stage, it should be

$$\mathcal{L}^{*}_{C^3} = -\frac{1}{B}\sum_{i=1}^{B}\log\frac{e^{\langle \mathbf{z}_{ii_0}, \mathrm{sg}(\mathbf{x}_{i_0}^{\mathcal{T}})\rangle}}{\sum_{j=0}^{N} e^{\langle \mathbf{z}_{ii_0}, \mathrm{sg}(\mathbf{x}_{i_j}^{\mathcal{T}})\rangle}}$$

where N denotes the number of hard-negative samples, i_j denotes the j^{th} text description (both positive and hard-negative) of the i^{th} image, with i_0 representing the positive pair.

Another version:

$$\mathcal{L}^{*}_{\mathrm{C}^3} = -\frac{1}{B}\sum_{i=1}^{B}\log\frac{e^{\langle \mathbf{z}_{ii_0}, \mathrm{sg}(\mathbf{x}^{\mathcal{T}}_{i_0})\rangle}}{\sum_{j=0}^{N} e^{\langle \mathbf{z}_{ii_j}, \mathrm{sg}(\mathbf{x}^{\mathcal{T}}_{i_0})\rangle}}$$

In addition, we apply a curriculum learning strategy. In the first stage, we train the model in the original configuration (without any augmentation on the training data). After several epochs, we move on to the second stage (hard-negative sampling) to let the model learn some more refined features by applying our method.