Stacked Cross Attention for Image-Text Matching

Microsoft AI and Research

Objective

1. Finding similarity between a image and a sentence.

Stacked cross Attention

1. Two Input :
   a. a set of image features v = {v1, v2, …, vk}
   b. Word features encodes a word in sentence e = {e1, e2, …, en}
2. Output :
   a. Similarity score : measure similarity of image-sentence pair.

Stacked Cross Attention for Image-Text Matching

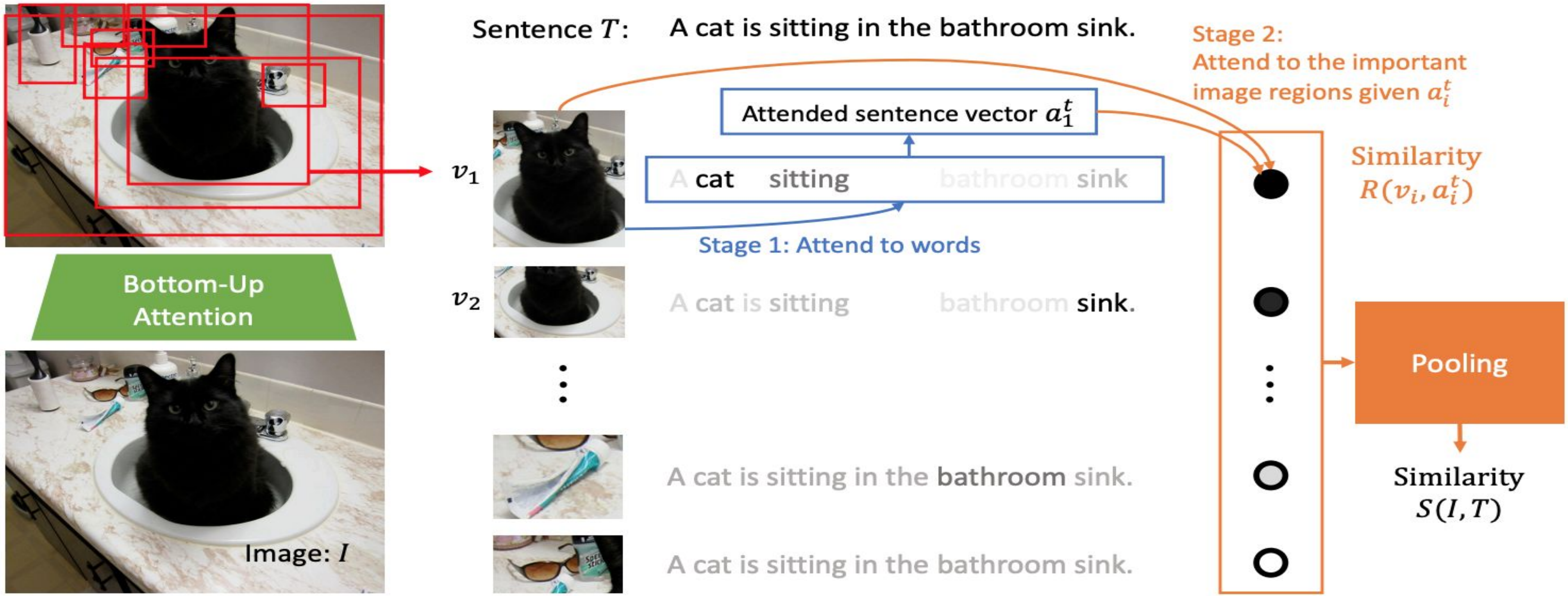

Sentence T: A cat is sitting in the bathroom sink.
Stage 2:
Attend to the important image regions given a_i^t
Attended sentence vector a_1^t
v_1
A cat sitting bathroom sink
Similarity $R(v_i, a_i^t)$
Stage 1: Attend to words
Bottom-Up Attention
v_2
A cat is sitting bathroom sink.
Pooling
A cat is sitting in the bathroom sink.
Similarity $S(I, T)$
A cat is sitting in the bathroom sink.
Image: I

Fig. 2. Image-Text Stacked Cross Attention: At stage 1, we first attend to words in the sentence with respect to each image region feature v_i to generate an attended sentence vector a_i^t for i-th image region. At stage 2, we compare a_i^t and v_i to determine the importance of each image region, and then compute the similarity score.

Image-Text Stacked Cross Attention

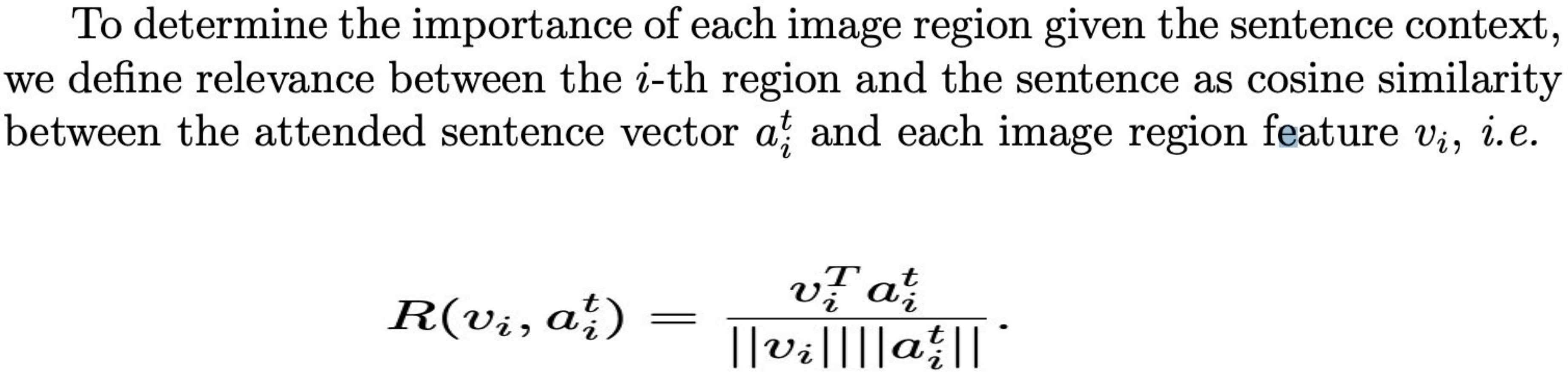
1. image features v = {v1, v2, …, vk}
2. Word features e = {e1, e2, …, en}
3. Cosine similarity matrix :

$$s_{ij} = \frac{v_i^T e_j}{||v_i||||e_j||}, i \in [1, k], j \in [1, n]$$

Stacked cross Attention

$$a_i^t = \sum_{j=1}^{n} \alpha_{ij} e_j,$$

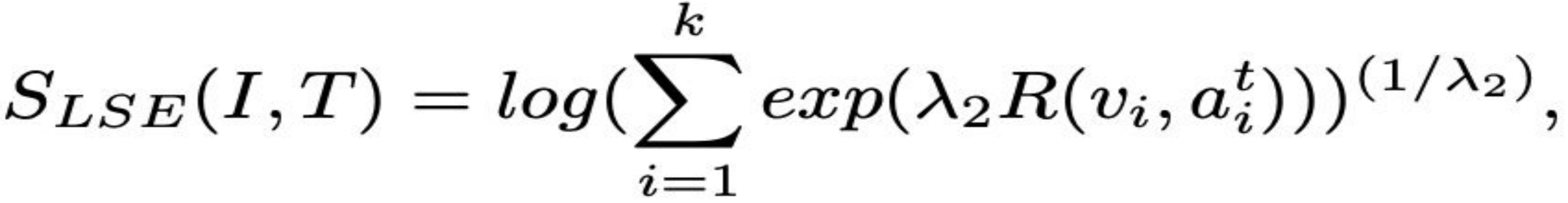
$$\alpha_{ij} = \frac{exp(\lambda_1 \bar{s}_{ij})}{\sum_{j=1}^{n} exp(\lambda_1 \bar{s}_{ij})},$$



To determine the importance of each image region given the sentence context, we define relevance between the i-th region and the sentence as cosine similarity between the attended sentence vector a_i^t and each image region feature v_i, *i.e.*

$$R(v_i, a_i^t) = \frac{v_i^T a_i^t}{||v_i|| \, ||a_i^t||}.$$

Inspired by the minimum classification error formulation in speech recognition, the similarity between image I and sentence T is calculated by LogSumExp pooling (LSE), i.e.



$$S_{LSE}(I,T) = log(\sum_{i=1}^{k} exp(\lambda_2 R(v_i, a_i^t)))^{(1/\lambda_2)},$$