DL 526 Final Project: Image-text Matching with Stacked Cross Attention

Komal Krishna Mogilipalepu

February 8, 2025

Abstract

We have chosen the topic image-text matching for the problem of cross modal retrieval with the motivation of its applications in fake news detection and e-commerce. In cross modal retrieval, we retrieve a most suitable image given a sentence and retrieve a most suitable sentence given an image based on the similarity score between the image and sentence. To find out the most suitable image-text pair, first it is important to extract the best features from the image and a sentence, and map them to a common embedding space to compare both. Secondly, it is crucial to differentiate the most important regions and words while finding the similarity score. Towards this, we review the paper stacked cross attention network for image-text matching [1] and provide our understanding of the paper.

1 Introduction

Matching image and text is a foundation for cross modal retrieval, which retrieves an image for a given sentence or retrieves a sentence for a given image. To find the most suitable image-sentence pair, there needs to be a semantic alignment between image regions and words of a sentence. Inferring the latent semantic alignment between image regions and words is crucial to identify the most suitable image-sentence pair. The latent semantic alignment between words of a sentence and regions of an image results to a well defined similarity score. Towards this, previous works [2, 3] find the similarity score of an image and text by aggregating the similarity of image regions and words of the given pair. In particular, previous work [3] considered the maximum of region-word similarity scores overall the regions with respect to each word and average the results corresponding to all words in a sentence to get the resultant similarity score of an image and a sentence. However, it did not consider the importance of words from the visual context. The current work [1] attends differentially to each region in an image and each word in a sentence with each other as a context to generate the similarity score. Towards this, the authors of [1] introduced *Stacked Cross Attention* that enables attention with context from both image and sentence in two stages.

Image-text formulation: Each image region attends to every word of a sentence and compare with the attended sentence vector to know the importance of that image region.

Text-image formulation: Each word attends to all image regions of an image and compare with attended image vector to know the importance of that word.

2 Methodology

2.1 Encoding Image and Text

Given an image, first the paper finds the objects or regions of an image by using the faster R-CNN and the regions are mapped to feature vectors $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k\}$, where each region $\mathbf{v}_i \in \mathbb{R}^D$. Similarly, for each word in a sentence represented by bag of words vector and mapped to feature vector $\{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_n\}$, where each word $\mathbf{c}_i \in \mathbb{R}^D$.

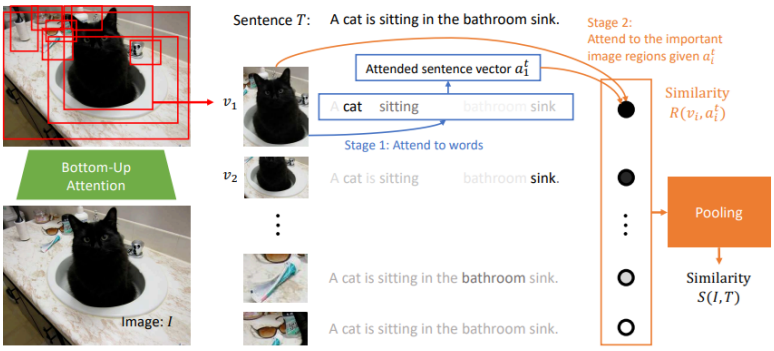

Figure 1: Illustration of Image-Text similarity calculation taken from [1]

2.2 Stacked Cross Attention Network

Given an image I and text T, the paper finds the similarity score between the pair I, T using the features $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k\}$ and $\{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_n\}$.

Image-Text Similarity: In image-text similarity, given an image, search for a suitable text. First, each image region attends to every word of a sentence to get the attended sentence vector and then each image region compared with the attended sentence vector to get the importance of that image region with respect to the sentence. The cosine similarity matrix s for all the regions i and for all the words j of an image-text pair is defined as

$$s_{ij} = \frac{\mathbf{v}_i^T \mathbf{c}_j}{\|\mathbf{v}_i\| \|\mathbf{c}_j\|} \mathbb{1}_{s_{ij}>0} + 0 \mathbb{1}_{s_{ij} \leq 0} \tag{1}$$

For better results, normalize the cosine similarity matrix s as $\widehat{s}_{ij} = \frac{s_{ij}}{\sqrt{\sum_{i=1}^{k} s_{ij}^2}}$. The attended sentence vector with respect to i-th image region is

$$\mathbf{a}_i^t = \sum_{j=1}^{n} \alpha_{ij} \mathbf{c}_j \quad \text{where} \quad \alpha_{ij} = \frac{\exp\left(\lambda_1 \widehat{s}_{ij}\right)}{\sum_{j=1}^{n} \exp\left(\lambda_1 \widehat{s}_{ij}\right)}, \quad \lambda_1 - \text{parameter} \tag{2}$$

The relevance between the each image region $\mathbf{v}_i$ and the attended sentence vector $\mathbf{a}_i^t$ is

$$R(\mathbf{v}_i, \mathbf{a}_i^t) = \frac{\mathbf{v}_i^T \mathbf{a}_i^t}{\|\mathbf{v}_i\| \|\mathbf{a}_i^t\|} \tag{3}$$

Finally, the similarity score for the given image I, and the sentence T is

$$\mathbf{S}_{LSE}(I, T) = \log\left(\sum_{i=1}^{k} \exp\left(\lambda_2 R(\mathbf{v}_i, \mathbf{a}_i^t)\right)\right) \tag{4}$$

Alternatively, the authors also used average similarity score in addition to LogSumExponential one, that is

$$\mathbf{S}_{Avg}(I, T) = \frac{\sum_{i=1}^{k} R(\mathbf{v}_i, \mathbf{a}_i^t)}{k} \tag{5}$$

Text-Image Similarity: In text-image similarity, given a text, search for a suitable image. First, each word attends to all the image regions of an image to get the attended image vector and each word compare with the attended sentence vector to know the importance of that word with respect to the

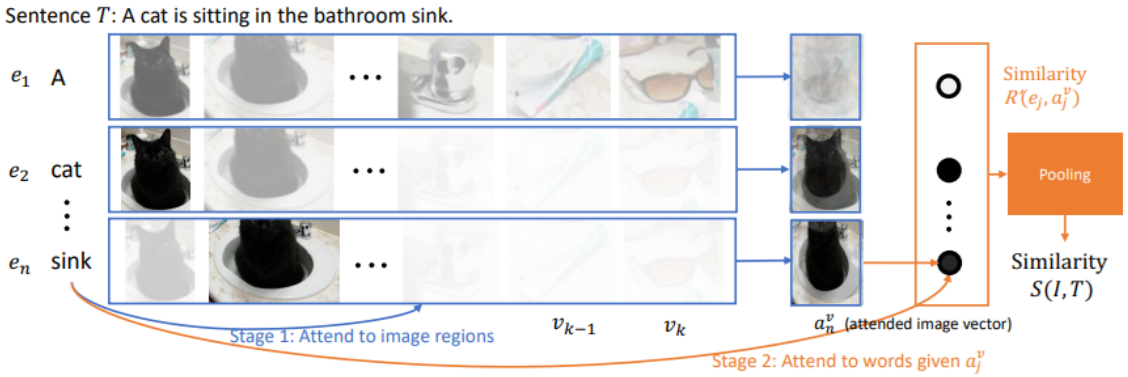

Figure 2: Illustration of Text-Image similarity calculation taken from [1]

image. The cosine similarity matrix $s^{'}$ for all the words j and for all the regions i of an text-image pair is defined as

$$s_{ij}^{'} = \frac{\mathbf{c}_j^T \mathbf{v}_i}{\|\mathbf{v}_i\| \|\mathbf{c}_j\|} \mathbb{1}_{s_{ij}^{'} > 0} + 0 \mathbb{1}_{s_{ij}^{'} \leq 0} \tag{6}$$

For better results, normalize the cosine similarity $s^{'}$ as $\widehat{s}_{ij}^{'} = \frac{s_{ij}^{'}}{\sqrt{\sum_{j=1}^{n} s_{ij}^{'2}}}$. The attended image vector with respect to j-th word is

$$\mathbf{a}_j^t = \sum_{i=1}^{k} \alpha_{ij} \mathbf{v}_i \quad \text{where} \quad \alpha_{ij} = \frac{\exp\left(\lambda_2 \widehat{s}_{ij}^{'}\right)}{\sum_{j=1}^{n} \exp\left(\lambda_2 \widehat{s}_{ij}^{'}\right)}, \quad \lambda_2 - \text{parameter} \tag{7}$$

The relevance between the each word $\mathbf{c}_j$ and the attended image vector $\mathbf{a}_j^t$ is

$$R^{'}(\mathbf{c}_j, \mathbf{a}_j^t) = \frac{\mathbf{c}_j^T \mathbf{a}_j^t}{\|\mathbf{c}_j\| \|\mathbf{a}_j^t\|} \tag{8}$$

Finally, the similarity score for the given image I, and the sentence T is

$$\mathbf{S}_{LSE}^{'}(I,T) = \log\left(\sum_{j=1}^{n} \exp\left(\lambda_2 R^{'}(\mathbf{c}_j, \mathbf{a}_j^t)\right)\right) \tag{9}$$

Alternatively, the authors also used average similarity score in addition to LogSumExponential one, that is

$$\mathbf{S}_{Avg}^{'}(I,T) = \frac{\sum_{j=1}^{n} R^{'}(\mathbf{c}_j, \mathbf{a}_j^t)}{n} \tag{10}$$

2.3 Objective:

In image-text matching hinge based Triplet ranking loss is the commonly used, that is

$$l(I,T) = \sum_{\widehat{T}} [\alpha - \mathbf{S}(I,T) + \mathbf{S}(I,\widehat{T})]_+ + \sum_{\widehat{I}} [\alpha - \mathbf{S}(I,T) + \mathbf{S}(\widehat{I},T)]_+, \quad \text{where} \quad [x]_+ = \max(x,0).$$

$\widehat{T}$ is all the negative texts and $\widehat{I}$ is all the negative images. The hinge loss is zero if the correct image-text pair I, T is close by a margin of α. Instead of considering all the negatives, the paper consider the loss which contains only the hard negatives, that is

$$l_{hard}(I,T) = [\alpha - \mathbf{S}(I,T) + \mathbf{S}(I,\widehat{T}_h)]_+ + [\alpha - \mathbf{S}(I,T) + \mathbf{S}(\widehat{I}_h,T)]_+. \tag{11}$$

where, $\widehat{I}_h = \arg\max_{m \neq I} \mathbf{S}^{'}(m,T)$ and $\widehat{T}_h = \arg\max_{d \neq T} \mathbf{S}(I,d)$.

3 Experiments

To show the effectiveness of stacked cross attention, authors considered the MSCOCO and Flickr30K datasets. Flickr30K contains 31000 images with 5 sentences for each image. Out of which, 1000 images are used for validation and 1000 images are used for testing. MSCOCO contains 123,287 images and each image annotated by five sentences. Out of which, 82783 images are used for training, 5000 images are used for validation, and 5000 images are used for testing. To show the best retrieval authors used the metric recall R @ K, which is the fraction of ground truth in the top K retrievals. The stacked cross attention network outperform the baseline methods by a significant amount.

3.1 Dataset Preprocessing

Flickr30K and MSCOCO contains the raw images with five descriptions for each image. The preprocessing of an image I involves detecting the regions (i) of an image and representing each region in a fixed length vector $\mathbf{v}_i$. The image regions i are identified by following the same analogy of [4], bottom-up attention using Faster R-CNN [5] and encoded to region features $\mathbf{f}_i$. The each region feature $\mathbf{f}_i$ is transformed to a h-dimensional vector using a fully connected layer as $\mathbf{v}_i = \mathbf{c}_v\mathbf{f}_i + \mathbf{b}_v$ where $\mathbf{v}_i \in \mathbb{R}^D$. On the other hand, preprocessing of sentences involve word embedding followed by a bi-directional GRU to represent words along with their context. Each word i in a sentence embed to a 300-dimensional vector through an embedding matrix $\mathbf{c}_x$ as $\mathbf{x}_i = \mathbf{c}_x\mathbf{c}_i$, where $\mathbf{c}_i$ is a one-hot vector with 1 at the index of the given word in the vocabulary. Now, all the embedded words are passed through a bi-directional GRU to map into a sentence feature with context as $\mathbf{c}_i = \frac{\overrightarrow{GRU}(\mathbf{x}_i)+\overleftarrow{GRU}(\mathbf{x}_i)}{2}, \forall i \in [1, n]$, where $\mathbf{c}_i \in \mathbb{R}^D$ and n is the number of words in a sentence.

3.2 Object Detection

Given an image I, it is crucial to extract all regions of the image to capture the semantic information. Classifying the objects in an image is one way to extract the image regions and has a significant role in image-text matching. In object detection, the image fed into a pre-trained convolutional network (CNN) to generate the features and the features are passed to two different fully connected layers with classification and regression losses to classify the object and predict the bounding box coordinates. Fine-tuning the CNN weights and fully connected layer weights by minimizing the loss functions result to an object detection network. Unfortunately, this process work only for single object in an image. For N objects, need to re-run the regression model N times to generate 4N coordinates. To mitigate this, one apply convolutional kernels on the cropped image regions; however the number of cropped regions grow with the image size. This limitation is addressed by considering only the regions of interest using the selective search method [6]. The selective search still keep multiple boxes on a single object and this can be eliminated by non-maximum suppression. Non-maximum suppression first sort all the bounding boxes confidence scores and greedily pick a highest confidence score bounding box. After that calculate the Intersection over Union (IoU) of all the other boxes with the selected bounding box and remove the bounding boxes with IoU score greater than a threshold value. The non-maximum suppression removes the redundancy of bounding boxes on a single object.

The process of selectively proposing a region and applying convolutional network on top of each region is named as R-CNN [7] but it consume lot of memory and takes more time. Instead of applying CNN for each region, Fast R-CNN apply the CNN on the entire image and the region proposal is performed on the features of the image to get the category and bounding box coordinates of an object. Faster R-CNN [8] improves the speed of Fast R-CNN by introducing a region proposal network inplace of selective search based proposals and using shared weights in the colvolutional blocks of generating image features and classifying the objects. Finally, Mask R-CNN is an improved version of Faster R-CNN, which predicts the instances of an object in addition to the object class and bounding box.

3.3 Bottom-Up Attention Using Faster R-CNN

Faster R-CNN consists of region proposal network followed by a region of interest pooling for detecting objects in an image by identifying the object regions, which belong to certain classes and localizing them with bounding boxes. First, the region proposal network predict the objectness score of anchor boxes of multiple scales and aspect ratios, and apply the greedy non-maximum suppression with an

Table 1: Stacked Cross-Attention Network Training Performance.

Image to Text	E = 1	E = 5	E = 10	E = 15	E = 20
R@1	14.1	25.6	28.2	30.8	30.8
R@5	28.2	56.4	57.7	59.0	57.7
R@10	50.0	71.8	73.1	74.4	74.4

Table 2: Stacked Cross-Attention Network Training Performance.

Text to Image	E = 1	E = 5	E = 10	E = 15	E = 20
R@1	5.4	8.7	9.7	9.5	9.5
R@5	15.6	21.0	21.3	21.8	21.0
R@10	24.1	30.3	30.0	30.3	30.5

Table 3: Stacked Cross-Attention Network Test Performance.

Retrieval	R@1	R@5	R@10
Image to Text	16.3	41.3	55.9
Text to Image	13.6	36.4	49.2

Intersection-over-Union (IoU) threshold to keep the top box object score and ignore the non-maximum ones. The top box proposals are fed to the second stage of Region of Interest (RoI) pooling to extract the small feature map from each proposal. The feature maps extracted from all the object proposals are combined and fed to a convolutional neural network followed by a softmax distribution over the class labels and class-specific bounding box refinements for each box proposal.

4 Reproduced Results

We train the model on the subset of Flickr30K train images and test the trained model on the Flickr30K test data. Each image contain five captions. First, the images and corresponding captions are loaded using PyTorch data loaders and then fed to an image encoder containing a fully connected layer to get the image embeddings. Similarly, the captions are passed through an text embedding layer that can handle variable size captions and uses bidirectional Gated Recurrent Units (GRUs) to get the textual embeddings. The model minimize the triplet loss that calculates the difference between the true image-sentence similarity and retrieved sentence for the given image similarity by a margin α. Similarly, calculate the loss for retrieved image for the query sentence. The triplet loss is minimized using the Adam optimizer with a learning rate of 0.0002. The image and text embeddings are passed to the shared attention model explained in 2.2 to calculate the similarity score in each case of image to text and text to image retrieval. Table 1 convey that the SCAN model improve the training performance recall at K, R@K, for $K = 1, 5,$ and 10 as the number of epochs E increases from 1 to 20. Similar observation holds for text to image retrieval as well as shown in 2. Comparing table 1 and table 2 reveals that the SCAN model is performing better in image to text retrieval compared to text to image retrieval. The best trained model is evaluated on the test dataset of Flickr30K and the performance is shown in table 3. Table 3 shows that as the number of retrievals increases from 1 to 10 the chance of ground truth presenting in the top retrievals increases. In addition, image to text retrieval has shown better performance than the text to image retrieval that justify test performance generalize the training performance.

4.1 Resources

The code and the datasets to reproduce the results of [1] is available on the github link https://github.com/kuanghuei/SCAN.

References

[1] Kuang-Huei Lee, Xi Chen, Gang Hua, Houdong Hu, and Xiaodong He, "Stacked cross attention for image-text matching," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 201–216.

[2] Andrej Karpathy, Armand Joulin, and Li F Fei-Fei, "Deep fragment embeddings for bidirectional image sentence mapping," *Advances in neural information processing systems*, vol. 27, 2014.

[3] Andrej Karpathy and Li Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3128–3137.

[4] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang, "Bottom-up and top-down attention for image captioning and visual question answering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6077–6086.

[5] Shaoqing Ren, "Faster r-cnn: Towards real-time object detection with region proposal networks," *arXiv preprint arXiv:1506.01497*, 2015.

[6] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders, "Selective search for object recognition," *International journal of computer vision*, vol. 104, pp. 154–171, 2013.

[7] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[8] RCNN Faster, "Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 9199, no. 10.5555, pp. 2969239–2969250, 2015.