Similarity Reasoning and Filtration for Image-Text Matching

Harsh Maniar
hxm5250@psu.edu

1 Task

Cross-modal retrieval refers to the process of retrieving information from one modality (such as an image) using information from another modality (such as text) as shown in Figure 1. This is a powerful tool that can be used to improve the accuracy and efficiency of information retrieval systems. For example, a cross-modal retrieval system might allow users to search for images using text queries, or to search for text documents using images as the query. Cross-modal retrieval systems typically use machine learning algorithms to model the relationship between different modalities, such as the relationship between text and images. These algorithms are trained on large datasets containing a variety of different modalities, such as image-text pairs. The goal of the training process is to learn how to accurately map between different modalities so that the system can effectively match a query in one modality with relevant results in another.

One of the key challenges in cross-modal retrieval is the ability to effectively align or match information from different modalities. For example, when searching for an image using text, it is important to be able to accurately match the words in the query with the visual content of the image. This can be difficult because the way that information is represented in different modalities can vary greatly. Image-text retrieval is a fundamental cross-modal task whose main idea is to learn image-text matching.
Image-text matching refers to measuring the visual-semantic similarity between image and text. It uses computer vision algorithms to identify and extract text from images, and then match that text with corresponding text in a database or document. This can be useful for a variety of applications, such as automatically indexing the content of images for search, or providing alternative text for images in documents for accessibility. There are several different techniques that can be used for image text matching. One approach is to use optical character recognition (OCR) to convert the text in the image into machine-readable text. This text can then be compared to the text in the database or document using string-matching algorithms. Previous research in this field utilized deep neural networks to first encode image and text into compact representation and then learn to measure their similarity (Diao et al., 2021; Li et al., 2019). Another approach is to use feature extraction techniques to extract visual features from the image and the database or document, and then use machine learning algorithms to compare the visual features and determine whether the image and text match. This can be more robust than OCR, as it is not dependent on the quality of the image or the text recognition accuracy of the OCR algorithm. Although great progress has been made in this field, image-text matching remains a challenge due to complex matching patterns and large semantic discrepancies between image and text (Diao et al., 2021).

2 Related Work

The Visual Semantic Reasoning Network (VSRN) is a model proposed in Li et al. (2019) for generating visual representations that capture both objects and their semantic relationships. VSRN starts by identifying salient regions in images using bottom-up attention as shown in Figure 2, which is implemented using Faster R-CNN. The network then builds connections between these salient regions and performs reasoning using Graph Convolutional Networks (GCN) to generate features with semantic relationships. To select the most important information and generate a representation for the whole image, the network uses a gate and memory mechanism to perform global semantic reasoning on the relationship-enhanced features. This reasoning process is conducted on a graph topology and considers local and global semantic correlations. The resulting image representation captures more key semantic concepts than existing methods, improving image-text matching performance.

A potential flaw of Li et al. (2019) is that it doesn't fully concentrate on the similarity encoding mechanism that models global image text and local region-word alignments comprehensively and fully encodes fine-grained relations between image and text. Diao et al. (2021) was able to overcome this by employing self-attention on region/word features to get image/text representations.

Wang et al. (2020) introduced a visual scene graph (VSG) and a textual scene graph (TSG) to represent images and text, respectively, converting the conventional image-text retrieval problem into the matching of two

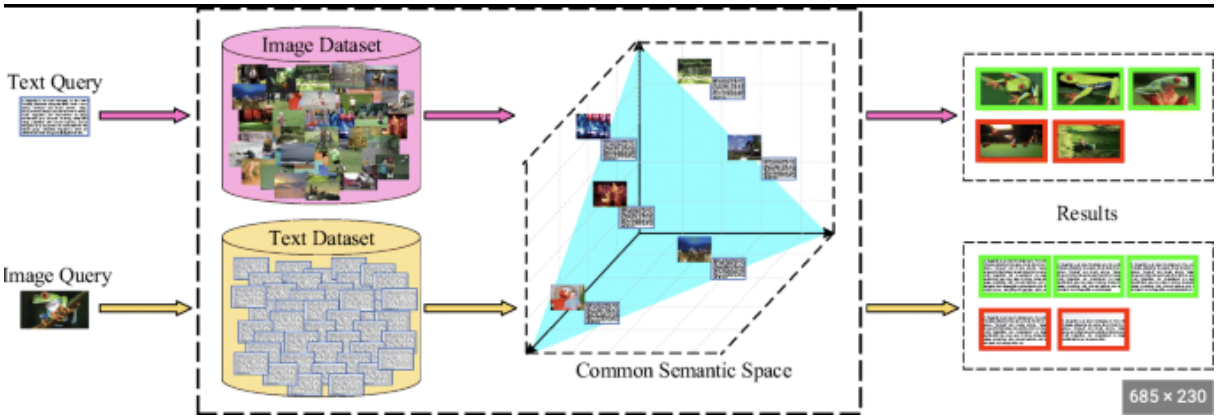

Figure 1: General illustration of Cross-modal Retrieval models

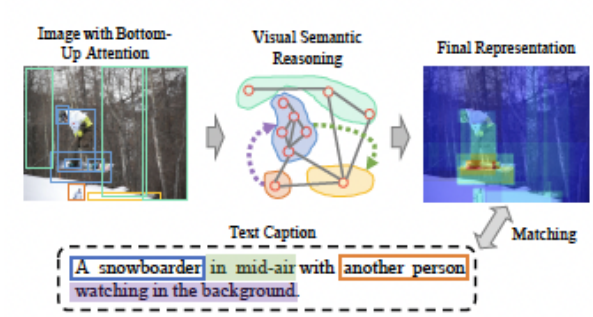

Figure 2: Illustration of Bottom-up Attention used in Li et al. (2019) and Diao et al. (2021)

scene graphs. To do this, they extract objects and relationships from the image and text to form the VSG and TSG, and design a Scene Graph Matching (SGM) model that uses tailored graph encoders to encode the VSG and TSG into visual and textual feature graphs. The VSG encoder is a Multi-modal Graph Convolution Network (MGCN) that enhances the representation of each node on the VSG by aggregating useful information from other nodes and updating the object and relationship features in different ways. The TSG encoder uses two different bi-GRUs to encode the object and relationship features. After this, object-level and relationship-level features are learned in each graph, and the two feature graphs corresponding to the two modalities can be matched at the object and relationship levels.
A flaw in Wang et al. (2020) is that The VSG and TSG separately refine visual and textual features including objects and relationships, both focusing on feature encoding by learning single-modality contextualized representations. However, it should have targeted similarity reasoning and explored more complex matching patterns with global and local cross-modal alignments.

Lee et al. (2018) approach strives to take a step towards attending differentially to important image regions and words with each other as context for inferring the image-text similarity. They introduced a method called Stacked Cross Attention that uses attention with context from both the image and sentence in two stages. In the Image-Text pipeline, given a image sentence pair, it first attends the words in the sentence to each image region, and proceeds to compare each region to the information from the sentence to decide the importance of the region. Similarly, in the Text-Image pipeline, it first attends the region in the image to each word in the sentence and then decides the importance. Stacked Cross Attention also discovers all possible alignments simultaneously, as the number of semantic alignments varies with different images and sentences, making image-text matching more interpretable.

Stacked Cross Attention uses a cosine similarity between the image vector and word feature to measure the relevance between the word and image. The final similarity score between the image and text is summarized by LSE (Lee et al., 2018). In contrast, Diao et al. (2021) network combines similarities by considering global and local relationships among vector-based alignments and reducing the influence of less-important alignments, which will do a better job of understanding the latent semantic alignments.

3 Approach

My motivation for choosing this paper was due to its excellent performance on Recall at 1 (R@1) which is defined as the proportion of queries whose ground truth is ranked within the top 1.

My approach to this task was fairly similar to the approach provided by the authors. The main difference between our approaches is during training. In the original paper (Diao et al., 2021), used a combination of the SAF and the SGR modules to create the SGRAF model. However, the authors failed to provide sufficient documentation to recreate the SGRAF model or even provide the pretrained weights for SGRAF model. No additional code was written by me. The main objective was to recreate the results of SOTA task and suggest some improvements. As shown in Figure 3 showcases there are two main tasks in this paper, Similarity Graph Reasoning (SGR) & Similarity Attention Filtration (SAF).

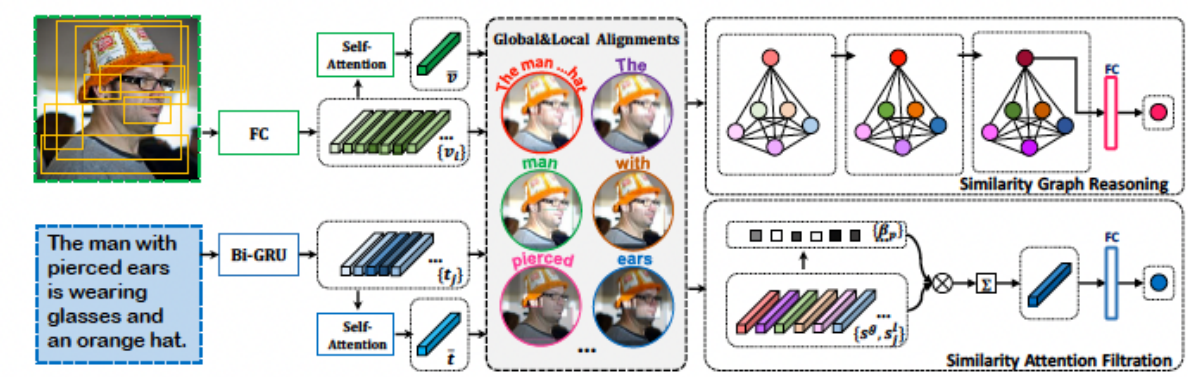

Figure 3: The SGR & SAF network as proposed in (Diao et al., 2021)for image-text matching. The image and sentence are firstly encoded into local and global feature representations, followed by a similarity representation computation module to capture the correspondence between all local and global cross-modal pairs. The Similarity Graph Reasoning (SGR) module reasons the similarity with giving consideration to the relationship between all the alignments, and the Similarity Attention Filtration (SAF) module attends to more informative alignments for more accurate similarity prediction

3.1 Generic Representation Extraction

For each of the input images, I used the bottom-up approach suggested in Anderson et al. (2018) to extract K region-level visual features using the Faster R-CNN proposed in Ren et al. (2015). This model was pretrained on the Visual Genomes dataset. We then performed self-attention over the local regions to obtain the global representation (Diao et al., 2021). For each input sentence, the authors used tokenization techniques to split it into L words and fed the word embeddings into a bidirectional GRU. The global text representation could to computed by using self-attention over all the word features (Diao et al., 2021).

3.2 Similarity Representation Learning

To represent the similarity between two feature vectors, we utilized the function shown below, to capture more detailed associations. Here W is a learnable parameter matrix used to obtain a m dimensional similarity vector.

$$s(x, y; W) = \frac{W|x - y|^2}{\left\|W|x - y|^2\right\|_2}$$

To exploit the local similarity representations between local features of images and text, we applied textual-to-visual attention used in Lee et al. (2018) to attend to each region with respect to each word.

3.3 Similarity Graph Reasoning

To achieve similarity reasoning, the authors incorporated a similarity graph to propagate similarity messages among the possible alignments at both the global and local levels (Diao et al., 2021). With this graph, we can perform similarity graph reasoning by updating the nodes and edges with the following functions shown below

$$\hat{s}_p^n = \sum_q e(s_p^n, s_q^n; W_{in}^n, W_{out}^n) \cdot s_q^n$$

$$s_p^{n+1} = ReLU(W_r^n \hat{s}_p^n)$$

This enables the SGR module to grasp the information propagation between local and global alignments. This will eventually capture more detailed and better interactions to facilitate the similarity prediction.

3.4 SAF

The Diao et al. (2021)'s approach emphasizes on enhancing the important alignments as well as supress ineffectual alignments. As we are given the local and global similarity representations, we can calculate the aggregate weight β_p for each similarity representation. This way it can learn the significant scores to increase the contribution of more informative similarity representation and reduce the disturbance of less-meaningful alignments (Diao et al., 2021).

$$\beta_p = \frac{\delta(BN(W_f s_p))}{\sum_{s_q \in \mathcal{N}} \delta(BN(W_f s_q))}$$

This enables the SGR module to grasp the information propagation between local and global alignments. This will eventually capture more detailed and better interactions to facilitate the similarity prediction.

4 Dataset

The SAF(Similarity Attention Filtration) and SGR(Similarity Graph Reasoning) models were trained on the MSCOCO (Lin et al., 2014) and Flickr30K datasets (Young et al., 2014). For this report, I recreated the results on the Flickr30K dataset. The Flickr30k

	Sentence Retrieval			Image Retrieval		
	$R@1$	$R@5$	$R@10$	$R@1$	$R@5$	$R@10$
(Full)SAF	73.4	92.3	95.8	56.1	81.4	87.8
(Pre)SAF	**73.6**	**92.7**	**96.9**	**56.5**	**82**	**88.4**
(Full)SGR	75.1	93.4	96.4	56.1	80.9	86.5
(Pre)SGR	**75.6**	**93.7**	**96.6**	**56.2**	**81.0**	**87**

Table 1: Comparison of bi-directional retrieval results (R@k) on Flickr30K test set. Here Pre prefix are the results of evaluating the pretrained model provided by Diao et al. (2021) and Full Prefix are the results from training the model from scratch

dataset was created to use the visual denotations of linguistic expressions, which here is the set of images they describe to define novel denotation similarity metrics. These metrics are beneficial for two tasks that require semantic inference. To compute these similarities, the dataset consists of a denotation graph, which is a subsumption hierarchy over constituents and their denotations (Young et al., 2014).

The Flickr30K dataset contains 31,783 images with 5 corresponding captions each. Following the split in Karpathy and Fei-Fei (2015), I use 1,000 images for validation, 1,000 images for testing, and the rest for training. Diao et al. (2021) have extracted the precomputed image features of Flickr30K from the raw Flickr30K images using the bottom-up attention model. This preprocessing was optional and I decided to not use this as the image features of Flickr30K are readily available in numpy array format, which can be used for training directly. However, if I planned on testing these models on another dataset, I would need to incorporate the bottom-up attention model for preprocessing.

5 Results

For each image, They take the Faster-RCNN detector used in Ren et al. (2015) with ResNet-101 provided by Anderson et al. (2018) to extract the topK = 36 region proposals. Using this information, the paper obtained a 2048-dimensional feature for each region. Moreover, for each sentence, the word embedding size is set to 300 with 1024 hidden states. The dimension of similar representation is set to 256 with a smooth temperature of 9 and a reasoning step of 3. Both the models (SGR & SAF) employ the Adam Optimizer with a mini batchsize of 128. The learning rate is initially set to 0.0002 for the first 30 epochs (20 for SAF) and then decay this by 0.1 for the last 10 epochs. The repository provided by the authors provide us the instructions to recreate their results as well as the pretrained weights which can be used to evaluate the model.

However, the authors didn't provide any instructions to recreate their 3rd model,SGRAF.(https://github.com/Paranioar/SGRAF/tree/python3.6). Additionally, the pretrained SGRAF model was also not provided to us. So for our evaluation, we will only compare the results of the SGR and SAF model. The authors decided to measure the performance by Recall at K (R@K). This is the metric widely used to better understand the performance for image-text retrieval tasks. It is defined as the proportion of queries whose ground-truth is ranked within the top K. We adopt R@1, R@5 and R@10 as our evaluation metrics.

Table 1 showcases the results I obtained by evaluating the SAF & SGR models by using the pretrained model and training from scratch. As you can see, we obtained marginally better results on the pretrained model on sentence retrieval as well as image retrieval. This might be because the authors might have trained and evaluated the model multiple times and chose to report the best results.

6 Possible Improvements & Results

In our search for improvements, we considered some hyperparameter tuning. For the SGR model, we decided to increase the learning rate by 33% (0.0003), increase the decay of the learning rate update by 50% (0.2) and start the decay earlier. The other parameters (number of epochs, hidden states, etc) were left unchanged. After evaluating this model, we noticed that the performance significantly decreased for sentence and image retrieval.

However, by changing the same parameters for the SAF model and reducing the number of epochs by 33% (20), we were able to recreate or better the results while reducing the training time by 33% percent. Table 2 showcases the results I obtained by evaluating the SAF models by using the pretrained model and the model we created by changing the hyper-parameters. As you can see, we obtained marginally better results on the pretrained model on image retrieval. However, when it comes to sentence retrieval we were able to outperform the pretrained model on the R@5 and R@10 metrics.

	Sentence Retrieval			Image Retrieval		
	$R@1$	$R@5$	$R@10$	$R@1$	$R@5$	$R@10$
(Full)SAF	**73.4**	92.3	95.8	**56.1**	**81.4**	**87.8**
(Our)SAF	72.7	**92.4**	**96.3**	54.4	80.1	86.8

Table 2: Comparison of bi-directional retrieval results (R@k) on Flickr30K test set. Here (Full)SAF are the results of evaluating the pretrained model provided by Diao et al. (2021) and (Our)SAF are the results from training the model with our modifications

7 Code Repository

My implementation of this paper is publically available at `google.com`. I have provided detailed instructions to recreate the results provided in Diao et al. (2021) and recreate the results provided in this paper. The repository also has links to download the dataset and the pretrained model provided by the original authors.

References

Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. 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*, pages 6077–6086.

Haiwen Diao, Ying Zhang, Lin Ma, and Huchuan Lu. 2021. Similarity reasoning and filtration for image-text matching. In *AAAI*.

Andrej Karpathy and Li Fei-Fei. 2015. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3128–3137.

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

Kunpeng Li, Yulun Zhang, Kai Li, Yuanyuan Li, and Yun Fu. 2019. Visual semantic reasoning for image-text matching. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4654–4662.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28.

Sijin Wang, Ruiping Wang, Ziwei Yao, Shiguang Shan, and Xilin Chen. 2020. Cross-modal scene graph matching for relationship-aware image-text retrieval. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 1508–1517.

Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. 2014. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78.