Object Recognition and Computer Vision: Composed Image Retrieval TOPIC A - 15/01/2024

Alex Pierron
Université Paris-Saclay
3 Rue Joliot Curie, 91190 Gif-sur-Yvette
alex.pierron@universite-paris-saclay.fr

Maxence Gollier
Institut Polytechnique de Paris
19 Pl. Marguerite Perey, 91120 Palaiseau
maxence.gollier@ip-paris.fr

Abstract

In this work, we investigate Composed Image Retrieval (CoIR) which is a task that requires both an image and a text query to search for corresponding images in a (large) database. We rely on Ventura et al. [6] to create a baseline and improve their results from there. Our improvements are quite straightforward but reach promising performance. Our code is publicly available at https://github.com/alex-pierron/Object_Recognition_Computer_Vision

1. Introduction

For this work, we were given the paper from Ventura et al. [6] and were asked to improve their results for the task of Composed Image Retrieval (CoIR). They investigate the task of Composed Video Retrieval (CoVR) as well but we do not make use of this in our analysis. Our contributions can be summarized as: (i) improve the training with contrastive and similar training and (ii) improve the underlying network architecture, from BLIP [1] to BLIP-2 [2].

2. Problem Statement

The aim of Composed Image Retrieval (CoIR) is given an image and a text query respectively q and t to retrieve an image i from a large database. We consider triplets (q, t, i) from manually annotated datasets such as [3]. The underlying idea of this approach is that either the image or the text query should make the retrieving task easier and more reliable. Figure 1 shows an example of such task.

We use the BLIP architecture from Li et al. [1] and the CIRR dataset [3] for our work. The next section gives a small introduction to the dataset we used.

2.1. CIRR Dataset

Datasets for CoIR face a major difficulty: it is hard to label every candidate image i in the database for the query $\langle q, t\rangle$. This in turn has the effect of creating many false-negatives. This is the reason for using the Recall@K metric for this task. The CIRR [3] dataset adresses this issue by manually collecting text queries which distinguish a set of similar images. To make the task more challenging, the said set of similar images is constructed as a subset of 6 images. Figure 2 summarizes this approach.

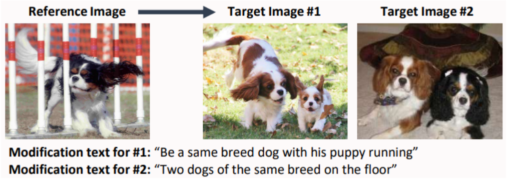

Reference Image
Target Image #1
Target Image #2
Modification text for #1: "Be a same breed dog with his puppy running"
Modification text for #2: "Two dogs of the same breed on the floor"

Figure 1. Example of CoIR (taken from: Liu et al. [3])

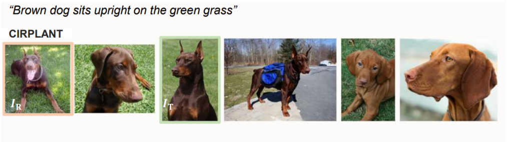


Figure 2. CIRR similar images (taken from: Liu et al. [3])

3. Upgrade I: Contrastive Training

Given the nature of the dataset. We can make use of the similarity subsets, called "members" described in Section 2.1. Namely, we use two opposite approaches:

1. For a given mini-batch, exclude all members from a same group. We call this approach *contrastive training*.
2. For a given mini-batch, include all members from a same group. We call this approach *similar training*.

We can make respectively two hypotheses on what should happen with these approaches.

1. *Hypothesis:* Contrastive training should make training *easier* because the elements in the mini-batches are easier to discriminate.

2. *Hypothesis:* Similar training should make training *harder* because the elements in the mini-batches are harder to discriminate.

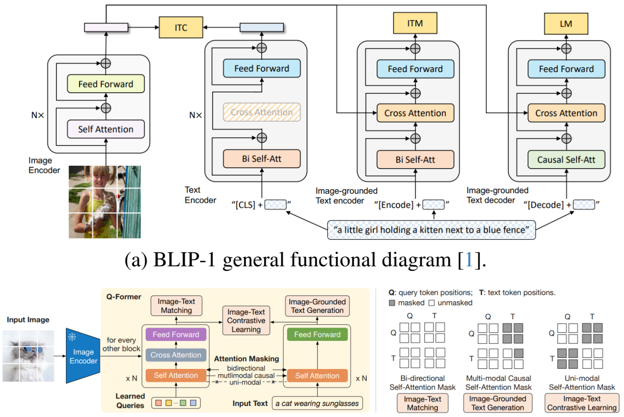
To implement this, we used pytorch Batch Sampler class and were able to guarantee that the batches keep the same given size. The above hypotheses are verified in our experiments available in Section 5.

4. Upgrade II: BLIP-2

In this section we detail the change of the base model used in [6] from BLIP-1 [1] to BLIP-2 [2].

4.1. Architectural change

We replace BLIP-1 used as described in [6] by the first-stage vision-language representation learning part of BLIP-2.
The major difference between BLIP-1 and BLIP-2 in our case is the introduction and the use of a new structure: the QFormer (see table 3).



(a) BLIP-1 general functional diagram [1].

(b) BLIP-2 first-stage vision-language representation learning part and QFormer [2]

Figure 3. general functional diagram for BLIP-1 and BLIP-2 first-stage vision-language representation learning part

This new architecture is supposed to be better than BLIP-1 for different tasks and benchmarks [2], especially on Image-Text-Matching which is the closest task to our goal.
In order to correctly use this architecture for our specific task, we had to perform several modifications.For the image-text input, after it passed through the QFormer we use a second projection to make sure it has the embedding dimension we want. Therefore we introduce a pooling on the channels of the query. This whole operation is also done for the target-images during computation of the embedding (for the embedding dimension) and during training for the pooling on channel axis.
These modifications allow us to use efficiently the same loss as described in [6].

4.2. Zero-Shot Learning

As a baseline, we use this model in a Zero-Shot Learning set-up with an embedding dimension of 768 and without doing any pooling as described before.
Implementation relied heavily on salesforce-lavis [5] and Lucas Ventura's code [4]. We used only the "pretrain" model type among those available for this zero-shot baseline and an embedding dimension of 768.

Recall@	**Score (Accuracy Perc.)**
1	6.988
2	11.542
5	19.614
10	26.819
50	43.904

Table 1. Recall Score for Zero-Shot Learning with BLIP-2 on CIRR [3] dataset.

All further experiments conducted and results obtained are detailed in section 5

5. Experiments

For all experiments, we used google cloud virtual machines with 1 NVIDIA T4, 60 Go RAM, 100 Go Disk and CUDA 11.8. In most cases, training time was about one hour per epoch. All implementations are available at `https://github.com/alex-pierron/Object_Recognition_Computer_Vision` which contains a *.txt* with all needed commands to reproduce our baseline. The GitHub repo is mainly a clone of [4] with some modifications for the relevant upgrades.

5.1. Contrastive Training

For the upgrade from Section 3, we trained the model for one epoch with a batch size of 64 and one GPU device. The other configurations are unchanged from the code of Ventura et al. [6]. The results are given at Table 2.
This corresponds to the hypotheses stated in Section 3, similar training is harder to train which explains why we get worse results after one epoch. Our computational power

Type	**Recall@1**	**Recall@5**	**Recall@10**
Baseline	**46.64**	75.95	84.53
Contrastive	46.41	**76.17**	**84.68**
Similar	42.24	69.59	79.42

Table 2. Contrastive training results on one epoch

did not allow us to go further one epoch but the similar training would probably however get better generalization performance after more epochs. We see on the other hand that contrastive training beats the baseline on two thirds of metrics.

Finally, results from 6, Table 3, we see a performance of 48.84. To verify that we had the correct approach, we let the model train for three epochs with a batch size reduced to 64 compared to the 512 from Ventura et al. [6]. We see on Table 3 that after 3 epochs, our baseline almost reaches the same performance which shows that our baseline is correct.

NumEpochs	Recall@1	Recall@5	Recall@10
1	46.68	75.95	84.53
3	**48.1**	**77.25**	**85.59**

Table 3. Baseline improvement for more epochs

5.2. BLIP-2

For our upgrade with BLIP-2, we conducted the following experiments on CIRR [3]: we trained all of our model for one epoch and for botch case where we use a maxpooling or a meanpooling, we tried two types of ViT model (*eva_clip_g* and *clip_L*. The Q-Former model was initialized with the *pretrain* model type available in lavis-force code[5].We used a batch size of 64 due to memory issues and tried two learning rate of $1e-4$ and $5e-5$.
Image embeddings were precomputed with *coco* pretrained weights with a final embedding dimension of 256.

ViT model	Pooling	Learning Rate	Recall@1	Recall@2	Recall@5	Recall@10	Recall@50
eva_clip_g	max pooling	1e-4	12.627	21.181	36.843	50.747	78.265
eva_clip_g	mean pooling	1e-4	20.578	31.783	50.554	65.398	86.964
eva_clip_g	mean pooling	1e-5	19.566	31.036	49.759	64.988	87.687
clip_L	max pooling	1e-5	9.229	16.337	30.434	45.759	77.012
clip_L	mean pooling	1e-5	19.59	30.265	48.337	63.205	87.349

Table 4. Recall results for BLIP-2 on CIRR test set. All models are trained for 1 epoch.

Table 4 shows that mean pooling is more efficient for every configuration tested. Since we only train for 1 epoch due to computing limitation, we can not be assertive that this trend continues if we extend the training. We still observe noticeable improvement by simply changing the type of pooling for our different model.

6. Conclusion

In summary, our work sought to replicate the results detailed in the original paper [6] for CIRR dataset. We can assert that our efforts were successful, as we were able to reproduce the reported results.

We implemented Contrastive Learning and this upgrade has yield some minor improvement to our baseline results. However due to computational limitations, it would be interesting to reconduct these experiments for a larger number of epoch to see how it generalizes.

Finally we implemented BLIP-2 as a base model. Given our technical limitations and compare to our baseline, BLIP-2 underperformed but It can be linked to the pretrained weights used and the higher complexity of the network. Nevertheless we have shown that mean-pooling yields higher accuracy than max-pooling.
For a future work, we suggest to conduct additionnal training with the different pretrained weights available (notably *coco* pretrained weights) and with more epochs to identify the real behavior of the proposed models.

References

[1] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified visual-language understanding and generation. *arXiv preprint arXiv:2201.07285*, 2022. 1, 2

[2] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models, 2023. 1, 2

[3] Zheyuan Liu, Cristian Rodriguez-Opazo, Damien Teney, and Stephen Gould. Image retrieval on real-life images with pre-trained vision-and-language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2125–2134, 2021. 1, 2, 3

[4] Lucas Ventura. Covr github. https://github.com/lucas-ventura/CoVR, 2023. Accessed: 2023-12. 2

[5] Salesforce Research, LAVIS Team. Blip-2 documentation. https://github.com/salesforce/LAVIS/tree/main/projects/blip2, 2023. Accessed: 2023-12. 2, 3

[6] Lucas Ventura, Antoine Yang, Cordelia Schmid, and Gül Varol. Covr: Learning composed video retrieval from web video captions. *arXiv preprint arXiv:2301.07762*, 2023. 1, 2, 3