Topic A : CoVR: Learning Composed Video Retrieval from Web Video Captions
15/01/2024

Baptiste CALLARD
ENS Paris-Saclay
Paris
baptiste.callard@ens-paris-saclay.fr

Steven Zheng
ENS Paris-Saclay
Paris
steven.zheng@ens-paris-saclay.fr

Abstract

The paper "CoVR: Learning Composed Video Retrieval from Web Video Captions" [5] introduces for the first time the Composed Video Retrieval (CoVR) task, an advancement of Composed Image Retrieval (CoIR), integrating text and video queries to enhance video database retrieval. Overcoming the limitations of traditional CoIR methods, which rely on costly manual dataset annotations, the authors developed an automatic dataset creation process and also released states of the art models for CoIR and CoVR. Our aim is to provide a comprehensive analysis of the solutions proposed in the paper, in particular by reproducing their experiments. We also propose to go further by studying explainability using attention mechanisms to understand model predictions. We study the sampling process with three new approaches, and innovate by replacing the original BLIP architecture with the more advanced BLIP-2. As a result, we have obtained a slight improvement compared with the original methods. Our Code : GitHub

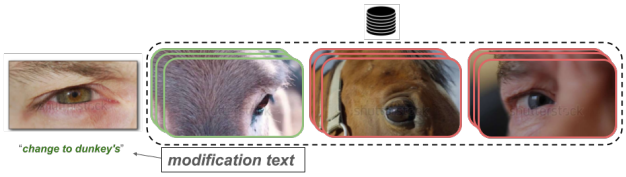

Figure 1. Composed Video Retrieval task [5]

1. Introduction

The research in Composed Image Retrieval (CoIR) [6] and the developments in Composed Video Retrieval (CoVR) [5] address crucial challenges in the field of digital media retrieval. Consider the scenario where you have access to a vast database of images or videos and need an efficient method to browse through them, seeking relevant content based on combined text and visual queries. This approach is particularly innovative in its use of a video accompanied by a text to guide the search for semantically altered versions of the actual video. The text describes the difference between the actual video and the target video. It could be a color changement (eye color), camera view (from the sky), subject (make it a woman) or action (make them dance). The implications of this technology are important in various professional fields. For instance, it could be a valuable tool for video editors, journalists, and even in everyday scenarios like tourism. The broad applicability and potential diverse use cases of this technology mark a significant advancement in digital media retrieval.

2. Problem Definition

The CoIR problem considers triplets. Let t be the text feature, q the query image feature. We denote the target database $B := \{v_i\}_i$ and $v_k \in B$ the target associated with the pair (t, q). Thus, given a triplet (q, t, v_k) our aim is to project $f(q, t)$ close to $h(v_k)$ into a latent space in $\mathbb{R}^d$ with f and h learnable functions. This joint projection is learned through contrastive learning.

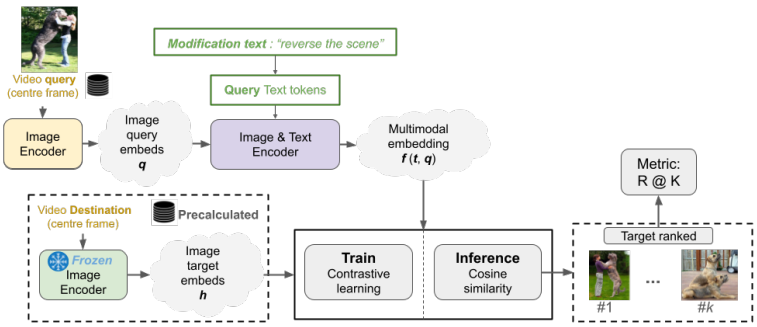

Figure 2. Overview of the **CoVR-BLIP** architecture

The CoVR problem can easily be transferred to the more classic CoIR problem. By definition, a video v is an ordered collection of images : $v := \{v_1, ..., v_n\}$. So all we need to do is identify a representative for each video in the form of an embedding. Several choices can be considered. We can

encode the middle frame or use random frames and average. To work correctly with textual data, images and videos, we need to embed them in a small latent space. The authors use the **BLIP** model [2]. **BLIP** is a model that is pre-trained for image-text contrastive learning, image-text matching and image-conditioned language modeling tasks. It can be used to encode text and image features: image-grounded text encoder, images only or text only. This model needs to be adapted to CoVR task therefore the author propose the architecture named **CoVR-BLIP** (see. figure 2). To be more precise, we display the structure of encoder block using the **BLIP** architecture in appendix 6.1.

3. Experiments

Experimental setup. In our study, we will restrict ourselves to the CoIR problem and study the CIRR dataset [4].

Dataset	Type	# Triplets	#Unique words	Domain
CIRR [4]	Image	36,554	321,185	Natural

We evaluated our results using Recall at rank k (R@k). This metric look at the number of times the target is in the top k ranks. We placed ourselves in restricted conditions to do our training as we did not have the same computing power as in [5]. We used a batch size of 64 and we did our training over one epoch (except if we specify it). We then re-trained the base-line under these conditions and used this result as a new reference. We used a **T4 GPU** and **60G RAM CPU** for our experiments which is the smallest usable framework. We reuse the GitHub from [5], but we insist that all our experiments are our own contribution from scratch. All our experiments represent approximately **4.34 kg CO2 eq**.

3.1. Replication of paper experiments

We first used a checkpoint from the **CoVR-BLIP** model trained on *cirr_ft-covr+gt* available on their GitHub and we also trained from scratch. We tested our results on the hidden test dataset. In the table 1, we can see that we do not get as good results for our training without checkpoint, which could be explained by our restricted framework. We will use this result as reference for the rest of our study.

3.2. BLIP-2

Following on from their work on **BLIP**, the authors have proposed : **BLIP-2** [3]. Their new model is composed of a Vision-and-Language Representation Learning encoder and a Vision-to-Language Generative Learning decoder. Our aim was to use the new architecture of their Vision-and-Language Representation Learning encoder and integrate it into **CoVR-BLIP**. The Q-Former of **BLIP-2** [3] refers to a bi-directional self-attention mask where all queries and texts can interact with each other. We used a method similar to the Image-Text Matching in **BLIP-2**. Consequently, the resulting query embeddings encapsulate rich information. Instead of feeding each output query embedding into the Image-Text Matching head, we did a projection and then applied a mean pool to finally have a image-text representation that we aim to align with the image target's representation.

For the first experiment, we used the image target's embedding directly from the frozen image encoder as for CoVR-BLIP, we note **P-BLIP-2** the results for this experiment. For the second experiment we used the embedding from the Q-Former that we obtained using the Image-Text Contrastive Learning mode of BLIP-2, where the queries and text do not see each other. We note **T-BLIP-2** the poor results for this experiment (see. appendix 6.3 with some illustration of the Encoder architecture).

For the training, we kept the same parameter as for our **CoVR-BLIP** training except using a lower batch size of 32[1]. We did not manage to reach the score of **CoVR-BLIP** with our best model **P-BLIP-2**. We observed an improvement between batches of 16 and 32. So we could expect improvement by using a batch of 64 and with finetuning.

		Recall@K			
Method	Ckpt	K=1	K=5	K=10	K=50
BLIP [5]	-	48.84	78.05	86.10	94.19
BLIP [5]	Yes	49.69	78.60	86.77	94.13
BLIP ref. (ours)	-	46.67	75.95	84.53	94.14
BLIP (ours)	Yes	49.59	78.72	86.77	94.21
P-BLIP-2 (ours)	-	36.10	69.54	80.58	92.92

Table 1. Replication of paper and BLIP-2 experiments.

3.3. Adapting the sampling strategy

Our first strategy is Hard Negative Sampling (*HNS*). The idea is to have all the images belonging to the same member in the same batch. A member is made up of images that are semantically very similar (see. figure 3). On the other hand, we implement a Filtering Sampling (*FS*). We would like to see the influence when the images of the same member are not in the same batch. The pseudo code is in the appendix 6.2.

Figure 3. Instances within a member

We use constrastive learning as loss function. By nature, triplets with close semantic will be closer compared to other members. In practice, *HNS* leads to complex batches and makes the task more complex. We observed a drop in performance because the task could be too difficult (even

[1]Maximum batch size before reaching out of memory

	Recall@K			
Method	K=1	K=5	K=10	K=50
BLIP ref. (ours)	46.67	75.95	84.53	94.14
HNS (ours)	43.49	71.97	80.41	92.07
FS (ours)	46.53	76.12	84.467	94.24
0.8*HN-FS* (ours)	46.67	76.58	84.87	94.19
0.7*HN-FS* (ours)	46.84	76.17	84.64	94.145
0.6*HN-FS* (ours)	47.20	76.17	84.46	93.95
0.4*HN-FS* (ours)	46.22	75.63	84.24	93.71

Table 2. Scores depending on the Sampling strategy for BLIP based models and for one epoch (except 4 for BLIP + HNS)

though we trained for 5 epochs). On the other hand, for *FS* the model have a smoother training and this leads to slightly better results than our reference model (see. table 2).
We believe that *HNS* can improve performance when used with parsimony. Thus, we propose β-Hard/Filtering Sampling (βHN-FS) which allows to control the part β of the batches with which the *FS* strategy is used β.*FS* + $(1-\beta)$. *HNS*. When the parameter β is below 0.4, a drop in performance is observed. However, this parameter β enables the creation of models with better confidence in $R@1$ when $\beta = 0.6$. Additionally, as β increases, it improves predictions for $R@K$ with larger values of K (to the detriment of small K). This indicates that playing with β allows a balance to be struck between HFS which allows the model to discern more complex cases, and FS which allows the model to perform well on average.

3.4. Attention

Close to the work of [1], we sought to observe the importance of the different modalities for predictions. To do this, we added a **MLP** whose role was to learn the importance of the different modalities. Formally, the aim of the **MLP** is to learn a mapping g defined by :

$$g : [q, t, f(t, q)] \mapsto W := [w_1, w_2, w_3] \in \mathbb{R}^{1\times 3}$$

Then, instead of returning $f(t, q)$, we return $w_1.q + w_2.t + w_3.f(t, q)$. We had easy access to q and $f(t, q)$. However, t was not directly accessible. We proposed two methods :
First, we extracted the text embedding within the **BLIP** image-grounded text encoder before it was mixed with the embedding image. Our second approach was to train a text encoder using BLIP text only encoder (see. figure 5 and 6). For these two approaches, it is evident that the model's performance (in App. 6.4) is enhanced by prioritizing multi-modal features and image over text features (see. figure 4). The uniform attention between the triplets, indicated by the low standard deviation, shows that attention is indifferent to the triplet. Overall, we experience significant performance drops, warranting cautious interpretation.
To address this, we implemented a finetuning strategy. We initially trained the entire model, followed by focusing on

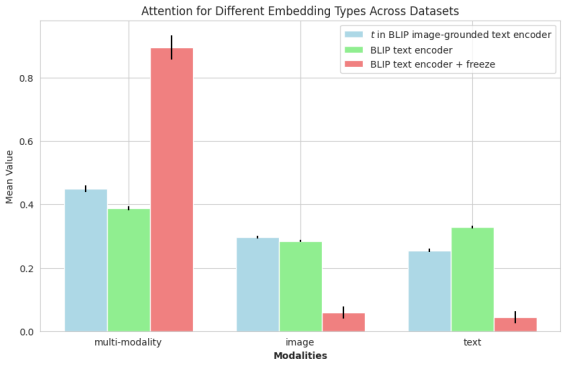

Figure 4. Modality attention

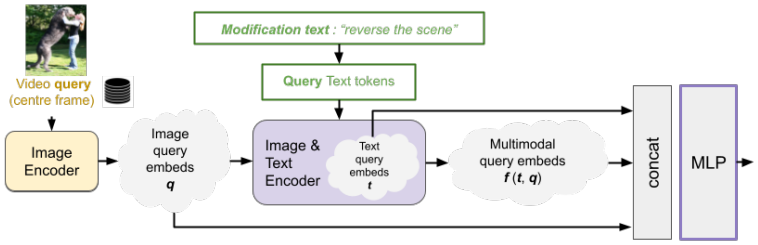

Figure 5. Time feature ***t*** extraction from the **BLIP** image-grounded text encoder

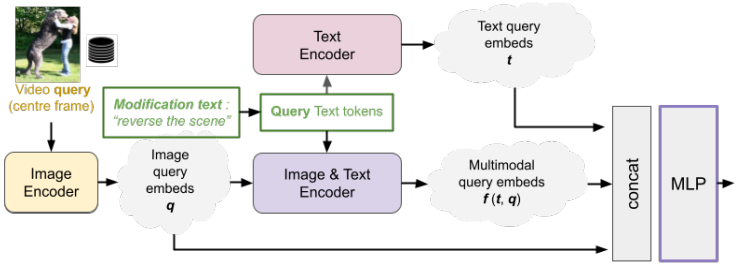

Figure 6. Time feature ***t*** extraction from a **BLIP** text only encoder

the text Encoder and MLP in the second epoch, and solely on the MLP thereafter. This approach not only improved performance and the focus on multi-modal features but also increased the standard deviation, indicating a more flexible use of modalities based on the triplet context. Something interesting that we did not look into due to lack of time.

4. Conclusion

The method studied is very general and therefore several extensions can be envisaged. We have studied the attention modalities of the model and could generalise it with attention maps on the image and also on the text. We observed that the sampling method had an impact on performance and we could imagine an active learning method where β_t could change over epochs. We could also include text reformulation, and classical image augmentation. We would also like to finetune BLIP-2-CoVR on larger machines and test alternative pooling methods.

5. Acknowledgement

We would like to thank Lucas Ventura for his mentoring during this project and his advice on the problems we encountered and the avenues we could pursue. We would also like to thank the teachers of the Recvis'23 course for giving us access to GCP machines.

References

[1] Matan Levy, Rami Ben-Ari, Nir Darshan, and Dani Lischinski. Data roaming and early fusion for composed image retrieval. *arXiv preprint arXiv:2303.09429*, 2023. 3

[2] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning*, pages 12888–12900. PMLR, 2022. 2, 4

[3] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023. 2

[4] 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*, pages 2125–2134, 2021. 2

[5] Lucas Ventura, Antoine Yang, Cordelia Schmid, and Gül Varol. Covr: Learning composed video retrieval from web video captions. *arXiv preprint arXiv:2308.14746*, 2023. 1, 2

[6] Nam Vo, Lu Jiang, Chen Sun, Kevin Murphy, Li-Jia Li, Li Fei-Fei, and James Hays. Composing text and image for image retrieval-an empirical odyssey. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6439–6448, 2019. 1

6. Appendix

6.1. BLIP encoders

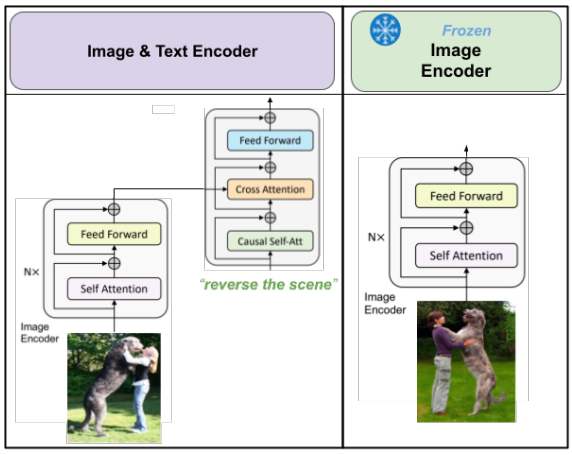

Figure 7. Encoder Structures in **CoVr-BLIP** [2]

6.2. Pseudo code sampling strategies

We propose a pseudo code for our Sampling methods.

Algorithm 1 Pseudocode for Creating *HNS*

```
Initialize batch to an empty set
Initialize batches to an empty set
Shuffle members
while there are enough members for a new batch do
    Clear batch
    loop
        Add a member to the batch
        if len(batch) ≥ batch_size then
            Break the loop
        end if
    end loop
    Shuffle batch
    Add batch to batches
end while
return Shuffle batches
```

Algorithm 2 Pseudocode for Creating *FS*

```
Initialize batches to an empty list
while there are triplets available do
    Shuffle members
    Initialize batch to an empty set
    i = 0
    while i ≤ batch_size and there are triplets available do
        Add a random triplet to batch
        i = i + 1
    end while
    Add batch to batches
end while
return shuffle batches
```

6.3. BLIP

6.3.1 BLIP-2 Encoder

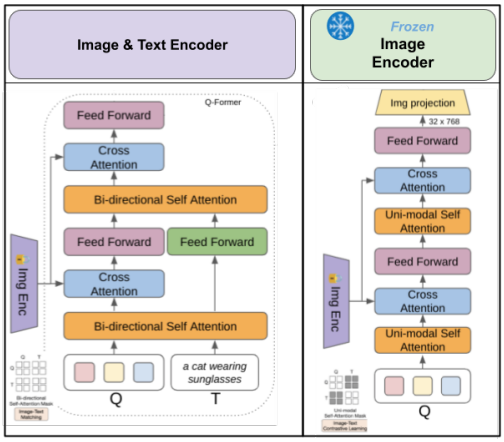

Figure 8. Encoder Structures in **CoVR-BLIP-2** (credit *L. Ventura*)

6.3.2 Results on T-BLIP-2

Poor performance due to a model with no pretrain and few computation resources.

	Recall@K			
Method	K=1	K=5	K=10	K=50
BLIP ref. (ours)	46.67	75.95	84.53	94.14
T-BLIP-2 (ours)	0.45	1.79	3.56	13.10

6.4. Performance for the study of attention

Here we present the performance of the different approaches. Adding the attention mechanism decreases performance. The results for **BLIP** + att_2 + *freeze* (ours) are better and are correlated to greater use of the mutlimodal feature.

	Recall@K			
Method	K=1	K=5	K=10	K=50
BLIP ref. (ours)	46.67	75.95	84.53	94.14
BLIP + att_1 (ours)	40.55	71.133	81.18	93.37
BLIP + att_2 (ours)	36.55	67.56	78.74	92.50
BLIP + att_2 + *freeze* (ours)	44.655	74.55	83.42	94.361