RecVis2024 Project Report
Topic A: Composed Image Retrieval

HO Quang Phuoc
ENS Paris-Saclay
11 Jan 2025

Abstract

This report presents my work on the papers "CoVR: Learning Composed Video Retrieval from Web Video Captions" [4] and "CoVR-2: Automatic Data Construction for Composed Video Retrieval" [5]. The first section provides a comprehensive study of the papers, offering a detailed explanation of the authors' methodology from my perspective. It highlights the main contributions and core approach of the proposed method.

In the subsequent sections, I describe my extended experimentation based on the paper's framework. The second section outlines the experimental setup, followed by an in-depth description of the three experiments I conducted to analyze their impact on performance.

1. Introduction to the Composed Video Retrieval

Composed Image Retrieval (CoIR) is a prominent task that uses both an image and a text query to retrieve the most relevant target images from a database. In this context, the input text describes the modification needed to transform the query image into the target images. Training a CoIR model requires manually constructing image-text-image triplets, a process that is inherently costly and labor-intensive.

In the original paper, the authors extend the scope of the task from Composed Image Retrieval to Composed Video Retrieval (CoVR), where the triplets are now in the form of image-text-video.

1.1. WebVid-CoVR dataset

The primary contribution of the paper is the creation of the WebVid-CoVR dataset, which consists of 1.6 million triplets. Figure 1 illustrates the methodology used to generate this dataset.

The authors automatically mine similar caption pairs from a large video-caption database sourced from the web. To describe the differences between these caption pairs, they use a modified text generation language model (MTG-LLM). MTG-LLM is trained on a dataset containing 715 triplet text annotations [1]. This approach allows the fully automated creation of triplets consisting of a query video q, a target video v, and a modification text t. Consequently, this method allows CoVR to generate scalable training data. After applying a post-processing pipeline, the final dataset comprises 1.6 million CoVR triplets.

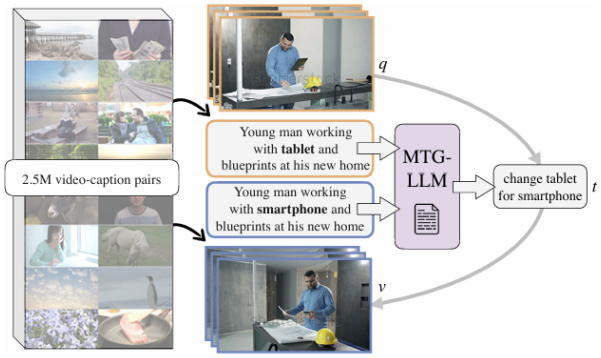


Figure 1. Method Overview

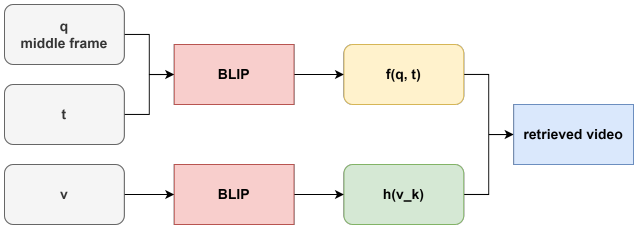


Figure 2. BLIP to CoIR/CoVR task

1.2. Training on WebVid-CoVR

The model architecture is built upon the pretrained image-text model, BLIP [2]. Figure 2 provides an overview of the workflow.

First, the query image q (corresponding to a key frame from the video) is encoded using the BLIP image encoder.

The resulting visual feature, along with the modification text t, is passed through the BLIP image-grounded text encoder, producing a multimodal embedding $f(q,t) \in \mathbf{R}^d$, where d represents the embedding dimension.

To retrieve a target video v_k, the embedding vectors for all gallery videos are computed as follows: N frames are sampled from each video, and the video embedding vector $h(v_k) \in \mathbf{R}^d$ is derived by calculating a weighted mean of the BLIP image embeddings. The weights are determined by evaluating the similarity between each frame and the modification text, using the pretrained BLIP image and text encoders.

Finally, given the multimodal embedding $f(q,t)$, the retrieved video is the one that maximizes the embedding similarity:

$$\arg\max_{v_k \in V} \left(h(v_k) \cdot f(q,t)^T \right)$$

The trained CoVR model is then transferred to image retrieval tasks on standard CoIR benchmarks, which is the main focus of my experiments.

2. Experimental Study and Findings

2.1. Setup

Unless otherwise specified, I fine-tuned the standard pretrained BLIP model on the WebVid-CoVR dataset using the CIRR dataset [1] and the HN-NCE loss function [3]. The model was trained for 5 epochs with a batch size of 64, and the learning rate was set to 1×10^{-4}. The training process was performed using a CPU with 2 workers and a single T4 GPU.

2.2. Reproduce the results

I used the provided checkpoints [4], [5] and compared the performance with the results reported in the original paper. The outcomes matched exactly, with the addition of an extra decimal place for improved precision. Details are shown in Table 4.

For convenience, I will refer to the authors' performance as the state of the art (SOTA) for later comparison.

2.3. Study Information Loss during training

To evaluate whether important information is lost during multimodal embedding computation, I introduce a small neural network. The architecture is illustrated in Figure 6.

- **Input:** The embedding of the multimodal embedding, image embedding q, and text embedding t.
- **Hidden layers:** Three fully connected layers with ReLU activation.
- **Output:** Three weights (w_1, w_2, w_3).

The final embedding is then computed as:

$$\mathbf{emb}_{\text{final}} = w_1 \times \mathbf{emb}_{\text{multi}} + w_2 \times \mathbf{emb}_{\text{img}} + w_3 \times \mathbf{emb}_{\text{text}}$$

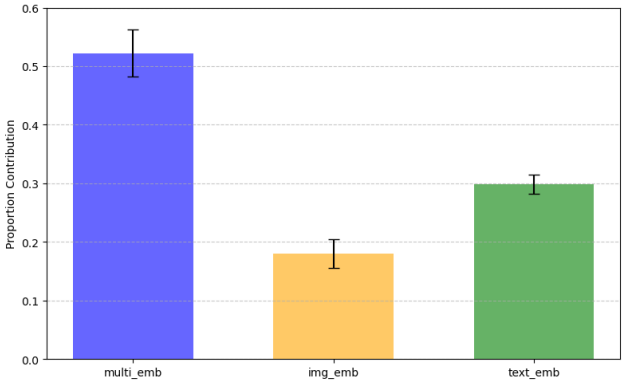


Figure 3. Proportional contribution of three embeddings

Figure 3 illustrates the proportional contribution of three embeddings to the final representation. The multimodal embedding contributes slightly more than 50%, while the text embedding accounts for 30%, and the reference image embedding contributes nearly 20%. Notably, the variance analysis shows that the text embedding exhibits the smallest variance, indicating that its contribution to the final embedding is the most stable.

R@1	R@5	R@10	R@50
50.15	80.60	89.35	**98.00**

Table 1. Testing performance on CIRR using the weighted embeddings method.

I observe that all scores are slightly lower than the SOTA, except for R@50, which shows a modest improvement from 97.64 to 98.00.

2.4. Introduce a Consistency Loss

The consistency loss is defined as a weighted sum of three similarity-based components: image-text consistency, multimodal consistency, and retrieval consistency. The first component encourages similarity between image and text embeddings. The second aligns the multimodal representation with the individual embeddings of image and text, promoting consistency between shared and unimodal spaces. The third component aligns the multimodal embedding with the target representation. The details are described in the Appendix D.

$$\mathcal{L}_{\text{consistency}} = \lambda_1 \mathcal{L}_{\text{image\_text}} + \lambda_2 \mathcal{L}_{\text{multi\_modal}} + \lambda_3 \mathcal{L}_{\text{image\_video}}$$

Here, λ_1, λ_2, and λ_3 are weighting coefficients for each component of the loss function, and they are set to be 1/3.

Compared to the authors' performance, this extension demonstrates significantly lower results, with a substantial drop in all R@k scores. This performance gap can be attributed to two key factors. First, the loss function is highly

R@1	R@5	R@10	R@50
13.35	36.80	52.39	80.19

Table 2. Testing performance on CIRR using the consistency loss.

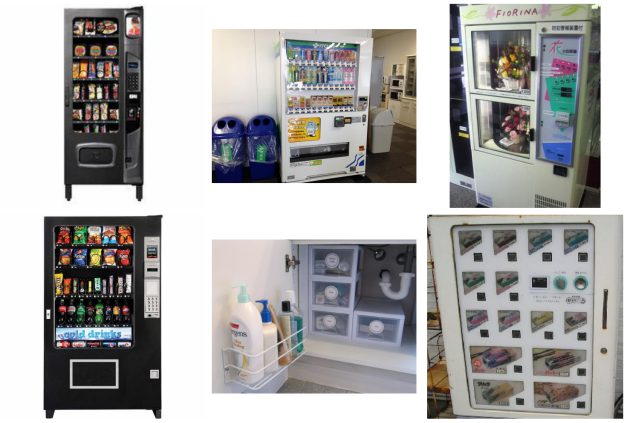


Figure 4. Example of images in the same member

sensitive to the values of λ_i, and using a constant value of $1/3$ for weighting is likely suboptimal, necessitating further fine-tuning. Second, there is a risk of conflicting objectives, as different consistency terms may pull the embeddings in opposite directions, potentially leading to slower convergence or reduced performance if not properly balanced. Additionally, this loss function increases computational complexity, resulting in longer training times for the model.

2.5. Sampling by Modifying the Data Loader

In this experiment, I modify the original data loader to adapt to two sampling methods. In the case of Hard Negative (HN) Sampling, the goal is to sample images from the same member within the same training batch, increasing the challenge for the model by forcing it to differentiate between similar inputs. In contrast, Excluding Sampling ensures that no images from the same member are included in the same training batch.

Figure 7 shows an example of an annotation in the CIRR test set, while Figure 4 displays the corresponding images from the same member.

Figure 5 compares the average training loss of the two sampling methods. It is evident that the training loss of Exclude Sampling is much lower than that of Hard Negative Sampling. This can be attributed to the fact that, with HN-NCE loss, sampling similar images in the same batch makes it difficult for the model to differentiate between positive and negative pairs, leading to poorer performance, as shown in Table 3. Notably, Exclude Sampling achieves a better score on R@50 compared to the SOTA.

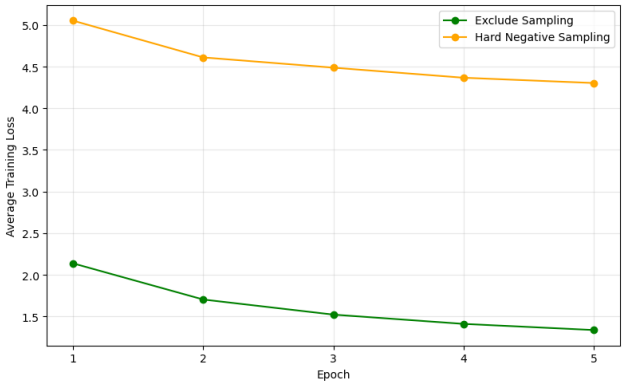


Figure 5. A comparison between training loss of two sampling methods

Method	R@1	R@5	R@10	R@50
HN sampling	43.30	79.49	80.10	94.10
Exclude sampling	48.75	79.78	88.29	**97.69**

Table 3. Testing performance on CIRR on two sampling methods.

3. Conclusion

In this report, I conducted a study on the papers [4] and [5] to explore the CoVR/CoIR problem and gain insights into its methodology through both theoretical and practical experiments. This was followed by three extended experiments aimed at improving performance. Although these experiments did not result in clear improvements, they provided valuable insights, revealing that several factors can influence performance. Therefore, further studies could be pursued, such as fine-tuning hyperparameters, increasing the batch size and number of epochs, or pre-training with the WebVid-CoVR dataset before fine-tuning on the CIRR dataset. I leave these avenues for future research.

References

[1] Tim Brooks, Aleksander Holynski, and Alexei A. Efros. Instructpix2pix: Learning to follow image editing instructions, 2023. 1, 2

[2] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Universal image-text representation learning, 2020. 1

[3] Filip Radenovic, Abhimanyu Dubey, Abhishek Kadian, Todor Mihaylov, Simon Vandenhende, Yash Patel, Yi Wen, Vignesh Ramanathan, and Dhruv Mahajan. Filtering, distillation, and hard negatives for vision-language pre-training, 2023. 2

[4] Lucas Ventura, Antoine Yang, Cordelia Schmid, and Gül Varol. CoVR: Learning composed video retrieval from web video captions. *AAAI*, 2024. 1, 2, 3, 4

[5] Lucas Ventura, Antoine Yang, Cordelia Schmid, and Gül Varol. CoVR-2: Automatic data construction for composed video retrieval. *IEEE TPAMI*, 2024. 1, 2, 3, 4

A. Acknowledgements

This report is primarily based on the original paper. For all experiments, I developed my extension using the base code provided by the authors of [4] and [5]. I would like to express my gratitude to the teaching team for their support, which included providing free Google Cloud credits and offering prompt assistance throughout the project.

B. Performance replication

Mode	Method	Pretraining	CIRR R@1	R@5	R@10	R@50
Train	CoVR-BLIP	-	49.16	79.76	88.65	97.49
	CoVR-BLIP (authors)	WebVid-CoVR	50.41	80.96	89.35	97.64
	CoVR-BLIP (mine)	WebVid-CoVR	50.386	80.964	89.349	97.639
Zero Shot	CoVR-BLIP	-	19.95	41.98	52.31	73.54
	CoVR-BLIP (authors)	WebVid-CoVR	39.28	68.22	78.94	94.65
	CoVR-BLIP (mine)	WebVid-CoVR	39.277	68.217	78.94	94.651

Table 4. Comparison of CoVR-BLIP performance between original papers and my reproduction on the CIRR dataset.

C. Embeddings incorporated by a neural network

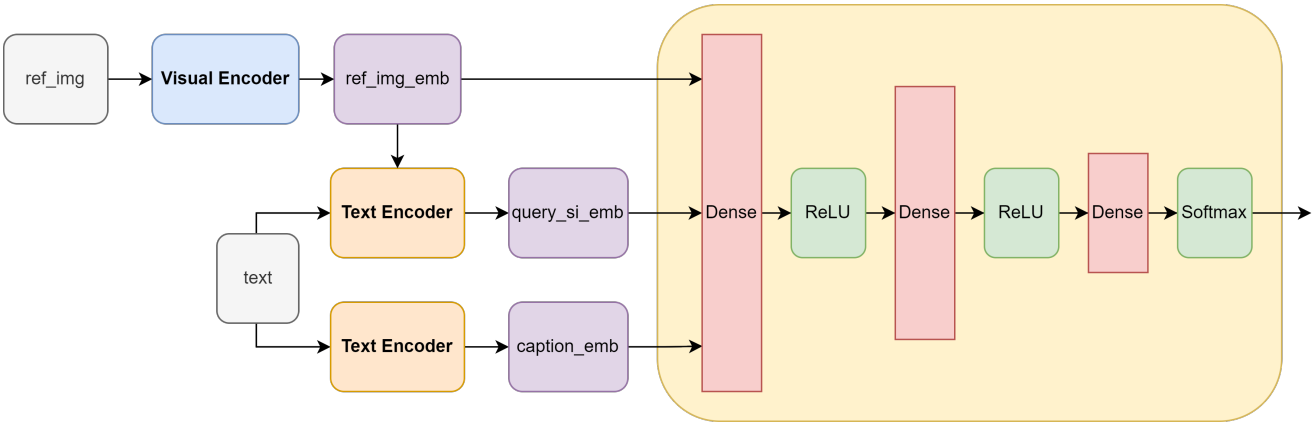


Figure 6. Embeddings incorporated by a neural network

D. Consistency Loss

Table 5. Consistency Loss Functions and Combined Loss

Consistency Type	**Loss Function**
Notation: Define $cs(a, b) :=$ cosine_similarity(a, b), $E_y(x)$ is the embedding of input x using encoder for data type y	
Image-Text Consistency	$\mathcal{L}_{\text{image\_text}} = 1 - cs(E_{\text{img}}(q), E_{\text{text}}(t))$
Multimodal Consistency	$\mathcal{L}_{\text{multi\_modal}} = 2 - cs(E_{\text{multi}}(q, t), E_{\text{img}}(q)) - cs(E_{\text{multi}}(q, t), E_{\text{text}}(t))$
Retrieval Consistency	$\mathcal{L}_{\text{image\_video}} = 1 - cs(E_{\text{multi}}(q, t), E_{\text{video}}(v))$
Combined Loss: $\mathcal{L}_{\text{consistency}} = \lambda_1 \mathcal{L}_{\text{image\_text}} + \lambda_2 \mathcal{L}_{\text{multi\_modal}} + \lambda_3 \mathcal{L}_{\text{image\_video}}$	

E. An example of CIRR annotation


```
"pairid": 12075,
"reference": "test1-105-1-img0",
"caption": "Change to black vending machine, Fill
machine with colorful drink and snack packages",
"img_set": {
    "id": 71,
    "members": [
        "test1-780-1-img0",
        "test1-546-2-img1",
        "test1-853-0-img0",
        "test1-76-2-img0",
        "test1-494-3-img0",
        "test1-105-1-img0"
    ],
    "reference_rank": 0
}
```


Figure 7. An example of CIRR annotation

F. Sample Retrieval Results

The following are two examples of retrieval tasks on the CIRR dataset. In each example, the left column contains the query image along with the modification text describing the desired change. The right side displays the top six results retrieved based on relevance to the query, arranged in descending order of relevance from left to right, top to bottom.

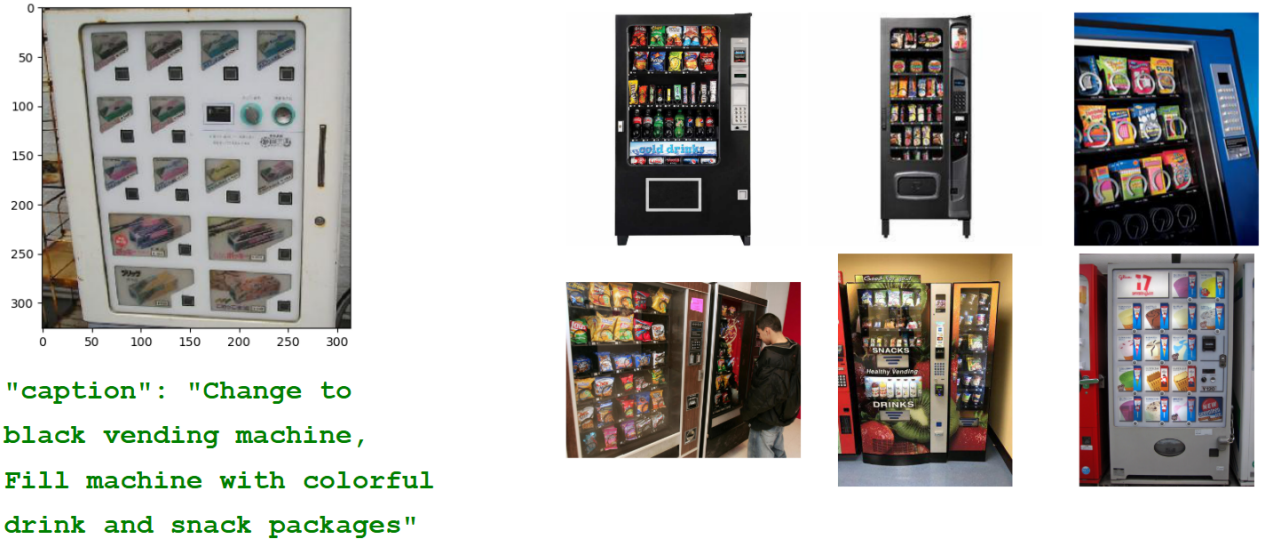
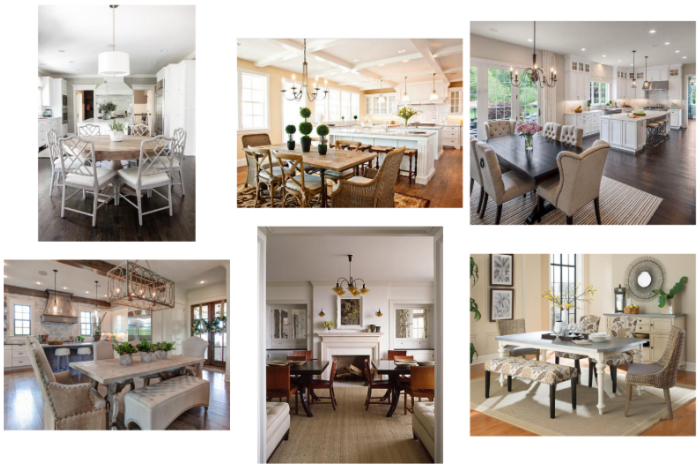


Figure 8. An example of retrieval results when target matches query image and text



"caption": "Shows another kitchen table that is round with a white base and dark brown top with four chairs to match."

Figure 9. An example of retrieval results when the target seems not to match query image and text