RUC_AIM3 at TRECVID 2024: Ad-hoc Video Search

Xueyan Wang, Yang Du, Yuqi Liu, Qin Jin*
School of Information, Renmin University of China
qjin@ruc.edu.cn

Abstract

This report presents our solution for the Ad-hoc Video Search (AVS) task of TRECVID 2024. Based on our baseline AVS model in TRECVID 2023, we further improve the searching performance by integrating multiple visual-embedding models, performing video captioning to be used for topic-to-caption searches, and applying a re-ranking strategy for top candidate search selection. Our submissions from our improved AVS model rank the 3^{rd} in TRECVID AVS 2024 on mean average precision (mAP) in the main task, achieving the best run of 36.8.

1 Introduction

Ad-hoc Video Search (AVS) is a challenging vision-language task, which aims to model the end user video search use case, who is looking for segments of video containing people, objects, activities, locations, etc., and combinations of the former (Awad et al., 2023).

In this year's task, there are 20 text queries, each of which can return up to 1000 shot IDs. The retrieval dataset is V3C2 (Rossetto et al., 2019), which includes 9760 videos with a total duration of 1300 hours. The mainstream solutions for the AVS task usually rely on image-text embeding models (Zhang et al., 2021; Radford et al., 2021). For example, He et al. (2023) ensemble multiple embedding models, achieving promising results on the AVS task. This demonstrates the powerful visual and textual understanding capabilities of image-text models can be effectively transferred to video tasks.

Hence, we consider fusing multiple image-text embedding models to obtain preliminary result on Ad-hoc Video Search. Specifically, we fuse CLIP (Radford et al., 2021), BLIP(Li et al., 2022), and their variants, Align (Jia et al., 2021), Flava (Singh et al., 2022), and InternVideo2 (Wang et al., 2024) as basic video search models. In addition, we use video captioning models to generate video captions for the dataset to achieve text-to-text retrieval. Finally, we generate 1,500 candidate videos for each query and select the best 1,000 using a re-ranking method. With the above components, our system ranks the 3^{rd} place in TRECVID AVS 2024.

2 Related Work

Recent years, more and more people have used pre-trained models, allowing retrieval systems to utilize a large amount of multi-modal training data to achieve better results on zero-shot text-to-video retrieval (Liu et al., 2019; Miech et al., 2018; Gabeur et al., 2020; Croitoru et al., 2021). Particular attention has been paid to basic models (e.g., CLIP (Radford et al., 2021), BLIP (Li et al., 2022)) which learned from large numbers of weakly aligned image-text pairs. They can use the most scalable pre-training data, and these models perform well on a range of vision and language tasks.

Sun et al. (Sun et al., 2023) proposed EVAClip, which incorporates new techniques for representation learning, optimization, and augmentation, achieving superior performance than CLIP at significantly reduced training costs. Wang et al. (Wang et al., 2023) proposed a video-text representation learning model ViClip based on ViT-L, which learns on InternVid via contrastive learning. BLIP2 (Li et al., 2023) outperforms BLIP in retrieval performance on both COCO and Flickr30K by bootstrapping visual language pre-training strategies from off-the-shelf frozen pre-trained image encoders and frozen large-scale language models. Jia (Jia et al., 2021) et al. proposed Align to jointly learn visual and language representations on a private data set including 1.8B text-image pairs. Wang et al. (Wang et al., 2024) proposed InternVideo2, which achieves SOTA in video retrieval on the MSRVTT dataset via a progressive training method and a new form of dataset. Therefore, we integrate

* Corresponding author

the above pre-trained models as the backbone retrieval model of our system.

3 Method

As illustrated in Fig. 1, our AVS system mainly consists of three components: the Captioning Module, the Retrieval Module, and the Candidate Re-ranking Module. The Captioning Module generates descriptions for given video dataset. The Retrieval Module forms preliminary retrieval result through image-to-text retrieval and text-to-text retrieval. The Candidate Re-ranking Module assigns a quality score for each candidate video generated by our system. Our system generates multiple candidate videos by the Retrieval Module. This process integrates two branches: directly comparing the video and query embedding and first using the Captioning Module to generate caption for the video then comparing the video caption and query embedding. With our Re-ranking Module, we then evaluate the quality of each candidate video and select the best 1000 as the final result.

3.1 Captioning Module

According to the retrieval video dataset, we use BLIP2(Li et al., 2022) to generate short video captions for each video, waiting for similarity matching with the query embeddings, which will be described in detail in Section 3.2. The purpose of this operation is to generate text-to-text retrieval, effectively converting cross-modal tasks into comparisons within the text modality. Fig. 2 shows two AVS queries in 2023 and the caption examples of the corresponding top 3 offical retrieval results. It can be seen that the generated captioning matches the query well. And experiments show that the retrieval result of this branch perform better than text-to-video retrieval on some queries. Through the Captioning Module, we obtained 3 lists of similarity scores.

3.2 Retrieval Module

In the following section, we describe the details of the embedding models used to construct the Retrieval Module. This module contains two branches: text-to-video retrieval and text-to-text retrieval. In model selection, we follow the following principles: (1) Select as many models as possible; (2) Select as many models as possible whose training datasets do not overlap; (3) Increase the weight of models with good performance and reduce the weight of models with poor performance.

Text-to-video Retrieval uses CLIP(Radford et al., 2021) and its variations (ViT-B/16, ViT-B/32, ViT-L/14, ViT-L/14@336), BLIP(Li et al., 2022) and its variations (ViT-B-CoCo, ViT-L-CoCo, ViT-B-Flicker30k, ViT-L-Flicker30k), BLIP2 (Li et al., 2023) , Align (Jia et al., 2021), Flava (Singh et al., 2022), and InternVideo2 (Wang et al., 2024).

Text-to-text Retrieval uses Sentence-transformers (Reimers and Gurevych, 2021), OpenAI (text-embedding-ada-002), and OpenAI (text-embedding-3-large).

For all the above text-to-video retrieval models, we extract 4 images for each video in the dataset, encode image embeddings, and store the mean embeddings for retrieval. When the user enters a text query, we extract the corresponding text vector and calculate its similarity with the image features saved by the corresponding model. For all the above text-to-text retrieval, we encode text embeddings for each video captioning in the dataset in Section 3.1 and store them for retrieval. When the user enters a text query, we extract the corresponding text vector and calculate its similarity with the captioning features saved by the corresponding model. For different pre-trained models of the same type, we normalize them according to the weight sets as shown in Table 1. The calculation formula is as follows:

$$\mathrm{s}^{\mathrm{t}} = norm(\frac{\sum_{p \in M} (s_p \times w_p)^{\ t}}{\sum_{p \in M} w_p}) \tag{1}$$

The procession for different types of pre-trained models is the same. The calculation formula is as follows:

$$\mathrm{s} = norm(\frac{\sum_{t \in T} (s^t \times w^t)}{\sum_{t \in T} w^t}) \tag{2}$$

Here, "s" represents the feature similarity, "w" represents the corresponding weight we assign to, "t" represents the type of the t^{th} pre-trained model, and "p" represents different pre-trained models of the same type. After calculation, we take the top 1500 videos with the highest similarity score s as candidate video list to be re-ranked.

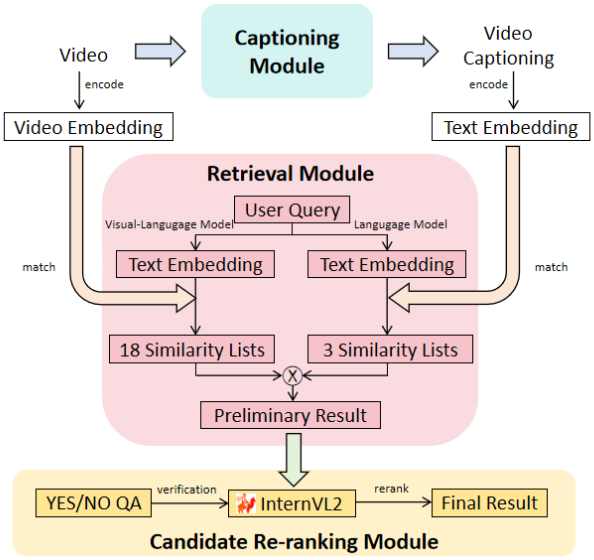

Figure 1: Our overall framework.

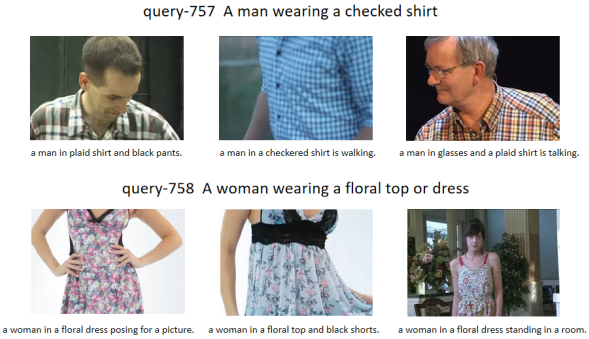

Figure 2: The examples of the top 3 offical retrieval results and their corresponding generated video captions.

3.3 Candidate Re-ranking Module

Because the basic video retrieval model may generate hallucination, we further use InternVL2 with a larger number of parameters to verify the preliminary retrieval result obtained above. First, we use GPT4(Achiam et al., 2023) to generate three "YES/NO" QAs for each AVS query. We require that all questions should revolve around people, objects, actions, locations, time, color, and quantity. The prompt of GPT4 is "You act as a question generator. Given a sentence, generate 3 simple and short QA pairs (only including YES/NO QA) for the given sentence, with extra emphasis on person/being, action, object, location, time, color, and quantity when specified. The number of YES and NO QAs should be balanced." We believe that questions that focus on a single factor and require a YES/NO answer are more helpful for verification. Fig. 3 shows an example of QA generation results:

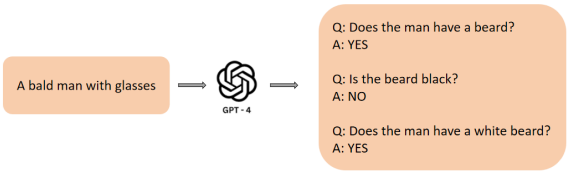

Figure 3: An example of QA generation for the query-751 Find shots of a bald man with glasses.

Query 751 looks for a bald man with glasses. We input the original AVS query to generate QAs. Secondly, we input the generated questions and candidate video lists into InternVL2-26B for visual-text question answering so that we can use it for post-processing re-ranking of retrieval results. We believe that InternVL2-26B reduces the generation of hallucinations due to its huge number of parameters and requires less data to be processed despite its high inference cost, thus ensuring the quality of re-ranking. Finally, we compared the verification results of InternVL2 on the candidate video list with the generated results of GPT4, and re-ranked the preliminary retrieval result from high to low according to the correctness of the answers. We take the top 1000 as the final retrieval results.

4 Experiment

4.1 the Impact of the Retrieval Module

The weight of each type of models is set to CLIP:openCLIP:BLIP:BLIP2:Align:Flava: InternVideo2:Captioning = 20:45:20:5:5:14:10:32. The

Table 1: Individual and fusion results for each model in 2023.

Model Type	Pre-trained Type	infAP	infAP after Fusion	
CLIP	ViT-L/14@336	0.0882	0.0945	**0.2936**
	ViT-L/14	0.0850		
	ViT-B/16	0.0220		
	ViT-B/32	0.0296		
openCLIP	ViT-L/14(Datacomp)	0.0526	0.1508	
	ViT-H/14(Laion)	0.1100		
	EVAClip	0.1346		
	ViClip	0.1217		
BLIP	ViT-B(CoCo)	0.1988	0.2343	
	ViT-L(CoCo)	0.2059		
	ViT-B(Flicker30k)	0.1749		
	ViT-L(Flicker30k)	0.1777		
BLIP2	BLIP2(CoCo)	0.1508	0.1508	
Align	Align-Base	0.1724	0.1724	
Flava	Flava-Base	0.0776	0.0776	
InternVideo2	s2-1B	0.1871	0.1871	
Captioning	Sentence-transformers	0.0689	0.1075	
	OpenAI(text-embedding-ada-002)	0.0898		
	OpenAI(text-embedding-3-large)	0.0914		

specific experimental results of each model and their Fusion in 2023 are shown in Table 1. It can be seen that the fusion of multiple embedding models will have better generalization ability than a single model, so the retrieval effect is better.

4.2 the Impact of Text-to-text Retrival Branch

We found that converting the video dataset into text-to-text retrieval by first generating captions for the retrieval can slightly improve the performance. We analyzed that this may be the reason for converting the cross-modal task into a task within the same modality. Combining reliable captioning models and embedding models, we can get the similarity between the query and the video.

In our experiments, although overall the performance of the text-to-text retrival branch is on par to different types of pre-trained models, we observe that the text-to-text retrival branch has advantages in certain queries such as Table 2. For example, in AVS 2023's queries 741, 742, 747 and 750, the text-to-text retrival branch performs significantly better than other pre-trained models, and even better than the fusion models in the table.

4.3 the Impact of the Candidate Re-ranking Module

To verify whether the Candidate Re-ranking Module really has an impact on the ranking list, we calculate the average position change of the video as follows:

$$c = \frac{1}{N}\sum_{i}^{N} |re\_rank\,(v_i) - ori\_rank\,(v_i)| \quad (3)$$

Table 2: Some queries text-to-text retrival branch performs better.

Model Type	infAP	Query
Fusion-CLIP	0.0044	741 A red or blue scarf around someone's neck.
Fusion-BLIP	0.0227	
Fusion-openCLIP	0.0169	
Captioning	**0.0312**	
Fusion-CLIP	0.0416	742 A child climbs an object outdoors.
Fusion-BLIP	0.0751	
Fusion-openCLIP	0.0795	
Captioning	**0.0976**	
Fusion-CLIP	0.0227	747 At least two persons are working on their laptops together in the same room indoors.
Fusion-BLIP	0.0977	
Fusion-openCLIP	0.0509	
Captioning	**0.2010**	
Fusion-CLIP	0.0054	750 A man with an earring in his left ear.
Fusion-BLIP	0.0192	
Fusion-openCLIP	0.0195	
Captioning	**0.0285**	

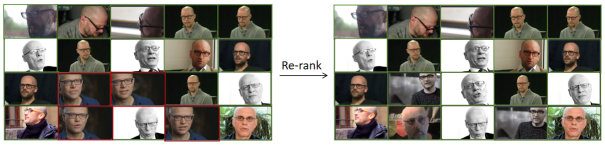

Figure 4: Re-ranking result of the query-751 Find shots of a bald man with glasses.

where "re-rank(v_i)" is the position of video v_i after re-ranking, "ori-rank(v_i)" is the position of video v_i before re-ranking, and "N" is the number of videos for re-ranking. Here, N is equal to 1500 in this paper. The result shows the average change c per query is about 100, which indicates the Candidate Re-ranking Module brings much change to the list.

Fig. 5 visualizes the original and updated rank lists of an improved query "A bald man with glasses". The related QA pairs is "Does the man have long hair?"-"No", "Is the man bald?"-"Yes", and "Is the man wearing glasses?"-"Yes". The performance of this query is elevated as videos of "people with thin hair" are pushed down.

4.4 Submission Results

We submitted a total of 4 runs in the main task of AVS 2024. Table 3 summarizes our solutions and evaluation results for each run of the main task of AVS 2024, among which run 4 has the best effect. By examining the performance of our runs on the main queries of this year, we conclude that adding a text-to-text retrival branch and the Candidate Re-ranking Module to the basic text-to-video retrival model are helpful to improve performance. il-lustrates the performance of all submitted runs in the AVS 2024 competition. Our runs achieved 3^{rd} place among all participating teams.

Table 3: Our solutions and evaluation results for each run of the main task of AVS 2023.

Run_ID	Text-to-video Retrival Branch	Text-to-text Retrival Branch	Candidate Re-ranking Module	infAP
0	✓			0.320
1	✓	✓		0.322
2	✓		✓	0.358
3	✓	✓	✓	0.368

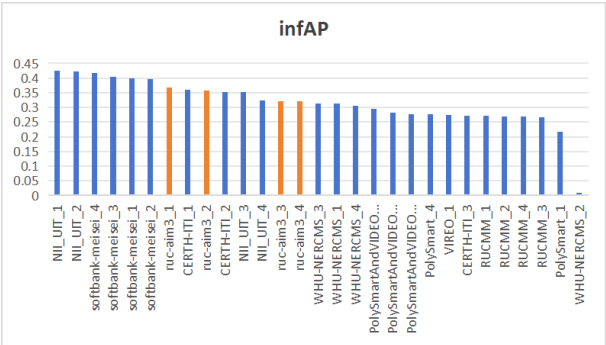

Figure 5: AVS 2024 ranking list of all submitted runs regarding the main task in infAP terms. Orange bars indicate our submitted runs.

5 Conclusion

This report presents our solution for the AVS challenge in TRECVID 2024. We integrate a series of powerful visual-text pre-trained models as the backbone to generate preliminary results for video retrieval. In order to transform the task to the same modality, we introduce video captioning models to achieve text-to-text retrieval. Finally, we re-rank the preliminary retrieval list to generate the final result. Experiments demonstrate the effectiveness of our designs, and our submissions rank 3^{rd} in the main task.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

George Awad, Keith Curtis, Asad A. Butt, Jonathan Fiscus, Afzal Godil, Yooyoung Lee, Andrew Delgado, Eliot Godard, Lukas Diduch, Deepak Gupta, Dina Demner Fushman, Yvette Graham, and Georges Quénot. 2023. Trecvid 2023 - a series of evaluation tracks in video understanding. In *Proceedings of TRECVID 2023*. NIST, USA.

Ioana Croitoru, Simion-Vlad Bogolin, Marius Leordeanu, Hailin Jin, Andrew Zisserman, Samuel Albanie, and Yang Liu. 2021. Teachtext: Crossmodal generalized distillation for text-video retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11583–11593.

Valentin Gabeur, Chen Sun, Karteek Alahari, and Cordelia Schmid. 2020. Multi-modal transformer for video retrieval. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16*, pages 214–229. Springer.

Jiangshan He, Ruizhe Li, Jiahao Guo, Hong Zhang, Mingxi Li, Zhengqian Wu, Zhongyuan Wang, Bo Du, and Chao Liang. 2023. Whu-nercms at trecvid 2023: Ad-hoc vedio search (avs) and deep video understanding (dvu) tasks. *Proceedings of TRECVID 2023*.

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. 2021. Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning*, pages 4904–4916. PMLR.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. 2022. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pages 12888–12900. PMLR.

Yang Liu, Samuel Albanie, Arsha Nagrani, and Andrew Zisserman. 2019. Use what you have: Video retrieval using representations from collaborative experts. *arXiv preprint arXiv:1907.13487*.

Antoine Miech, Ivan Laptev, and Josef Sivic. 2018. Learning a text-video embedding from incomplete and heterogeneous data. *arXiv preprint arXiv:1804.02516*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.

Nils Reimers and Iryna Gurevych. 2021. Sentence transformers: Multilingual sentence, paragraph, and image embeddings using bert & co. *URL: https://github.com/UKPLab/sentence-transformers*.

Luca Rossetto, Heiko Schuldt, George Awad, and Asad A Butt. 2019. V3c–a research video collection. In *MultiMedia Modeling: 25th International Conference, MMM 2019, Thessaloniki, Greece, January 8–11, 2019, Proceedings, Part I 25*, pages 349–360. Springer.

Amanpreet Singh, Ronghang Hu, Vedanuj Goswami, Guillaume Couairon, Wojciech Galuba, Marcus Rohrbach, and Douwe Kiela. 2022. Flava: A foundational language and vision alignment model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15638–15650.

Quan Sun, Yuxin Fang, Ledell Wu, Xinlong Wang, and Yue Cao. 2023. Eva-clip: Improved training techniques for clip at scale. *arXiv preprint arXiv:2303.15389*.

Yi Wang, Yinan He, Yizhuo Li, Kunchang Li, Jiashuo Yu, Xin Ma, Xinhao Li, Guo Chen, Xinyuan Chen, Yaohui Wang, et al. 2023. Internvid: A large-scale video-text dataset for multimodal understanding and generation. *arXiv preprint arXiv:2307.06942*.

Yi Wang, Kunchang Li, Xinhao Li, Jiashuo Yu, Yinan He, Guo Chen, Baoqi Pei, Rongkun Zheng, Jilan Xu, Zun Wang, et al. 2024. Internvideo2: Scaling video foundation models for multimodal video understanding. *arXiv preprint arXiv:2403.15377*.

Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. 2021. Vinvl: Revisiting visual representations in vision-language models. In *Proc. of CVPR*, pages 5579–5588.