Renmin University of China at TRECVID 2022: Improving Video Search by Feature Fusion and Negation Understanding

Xirong Li, Aozhu Chen, Ziyue Wang, Fan Hu, Kaibin Tian, Xinru Chen, Chengbo Dong
MOE Key Lab of DEKE, Renmin University of China
AIMC Lab, School of Information, Renmin University of China
https://ruc-aimc-lab.github.io

Abstract

We summarize our TRECVID 2022 Ad-hoc Video Search (AVS) experiments. Our solution is built with two new techniques, namely Lightweight Attentional Feature Fusion (LAFF) for combining diverse visual / textual features and Bidirectional Negation Learning (BNL) for addressing queries that contain negation cues. In particular, LAFF performs feature fusion at both early and late stages and at both text and video ends to exploit diverse (off-the-shelf) features. Compared to multi-head self attention, LAFF is much more compact yet more effective. Its attentional weights can also be used for selecting fewer features, with the retrieval performance mostly preserved. BNL trains a negation-aware video retrieval model by minimizing a bidirectionally constrained loss per triplet, where a triplet consists of a given training video, its original description and a partially negated description. For video feature extraction, we use pre-trained CLIP, BLIP, BEiT, ResNeXt-101 and irCSN. As for text features, we adopt bag-of-words, word2vec, CLIP and BLIP. Our training data consists of MSR-VTT, TGIF and VATEX that were used in our previous participation. In addition, we automatically caption the V3C1 collection for pre-training. The 2022 edition of the TRECVID benchmark has again been a fruitful participation for the RUCMM team. Our best run, with an infAP of 0.262, is ranked at the second place teamwise.

1 Our Approach

Our solution for the TRECVID 2022 (TV22) AVS task is based on two newly developed techniques. One is the Lightweight Attentional Feature Fusion (LAFF) [6], an attention-based feature fusion method that performs feature fusion at both early and late stages and at both video and text ends. The other is Bidirectional Negation Learning (BNL) [20], a learning based method for training a negation-aware video retrieval model, which is used to handle queries that contain negative cues, *e.g.* "*A man is holding a knife in a non-kitchen location*".

1.1 LAFF-based Video Retrieval

Given a video x represented by a set of k_1 video-level features $\{f_{v,1},\ldots,f_{v,k_1}\}$, LAFF performs feature fusion by first transforming each of the k_1 features to a d-dimensional feature vector and then aggregating the transformed features into a combined feature $\bar{f}_v$ by a convex combination:

$$\begin{aligned}\bar{f}_v &= \text{LAFF}(\{f_{v,1},\ldots,f_{v,k_1}\}) \\ &= \textstyle\sum_{i=1}^{k_1} a_i \times Linear(f_{v,i}),\end{aligned} \tag{1}$$

where *Linear* denotes a fully connected layer followed by a *tanh* activation, while $\{a_i\}$ are feature-specific weights computed by a lightweight attentional mechanism, see Fig. 1. Similarly, given a textual query (or sentence) q represented by a set of k_2 sentence-level features $\{f_{t,1},\ldots,f_{t,k_2}\}$, we obtain its combined feature $\bar{f}_t$ as $\text{LAFF}(\{f_{t,1},\ldots,f_{t,k_2}\})$.

In order to compute a cross-modal similarity $s(x,q)$ between x and q, the above video-specific and text-specific LAFFs shall be paired and trained jointly. In this work, we use h pairs for computing their cross-modal sim $s(x,q)$. We use $j=1,\ldots,h$ to index each pair and the resultant features $\bar{f}_{v,j}$ and $\bar{f}_{t,j}$. Accordingly, we have

$$s(x,q) = \frac{1}{h}\sum_{j=1}^{h} \text{cosine-sim}(\bar{f}_{v,j},\bar{f}_{t,j}). \tag{2}$$

1.1.1 Choice of Visual Features

The following six deep visual features are used:

1. *wsl*: A 2,048-d frame-level feature, extracted by ResNeXt-101 which pre-trained on weakly labeled web images followed by fine-tuning on ImageNet[1] [16].

2. *beit*: A 1,024-d frame-level feature, extracted by BEiT which pre-trained on full ImageNet and fine-tune on 1k-class ImageNet [2] [3].

3. *clip*: A 768-d frame-level feature, extracted by a pre-trained CLIP (ViT-L/14)@336 model. [3] [18].

[1] https://github.com/facebookresearch/WSL-Images
[2] https://github.com/microsoft/unilm/tree/master/beit
[3] https://github.com/openai/CLIP

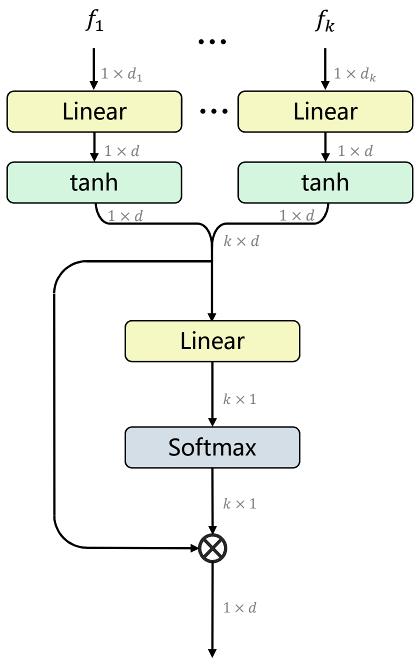

Figure 1: Lightweight Attentional Feature Fusion (LAFF) [6].

4. *clip-bnl*: A 512-d frame-level feature, extracted by a CLIP-*bnl*, which re-trains CLIP(ViT-B/32) by bidirectional negation learning on re-purposed MSR-VTT [20].

5. *blip*: A 256-d frame-level feature, extracted by a BLIP(ViT-B) [7] which pre-trained on 129M image-text pairs [4].

6. *ircsn*: A 2,048-d segment-level feature, extracted by irCSN-152 [5] which trained on IG-65M[5].

1.1.2 Choice of Textual Features

We experimented with the following five sentence encoders for textual features:

1. Bag-of-Words (bow) [4, 14], which has a dimensionality of 16k according to our training data.
2. Word2Vec (w2v) [17] pretrained on Flickr tags [10] .
3. Text encoder of CLIP(ViT-L/14)@336 [18], which is a GPT.
4. Text encoder of CLIP-*bnl* [20], *i.e.* a GPT.
5. Text encoder of BLIP(ViT-B) [7], which is a BERT.

1.1.3 Choice of (Pre-)Training Data

Different from our TV21 system [8] that uses image collections for pre-training, this year we use a self-built video-text dataset, namely V3C1-Pseudo-Caption (V3C1-PC), as the pre-training dataset, see Table 1. Specifically, for each video in V3C1, we use BLIP to generate a caption for each sampled frame. A n-frame video will have n captions. We remove duplicate captions and then use CLIP to rank the remaining captions in terms of their cross-modality similarity to the video. The top-3 ranked captions are preserved as the video's pseudo captions. The caption data is publicly available at https://github.com/ruc-aimc-lab/v3c1-pc.

As for training data, we use the joint set of MSR-VTT [21], TGIF [15] and VATEX [19] (M+T+V). Following our conventional setup [9, 11–13], the development set of the TRECVID 2016 Video-to-Text Matching task [2] is used as an external validation set[6] for base model selection.

Table 1: Statistics of V3C1-PC.

Dataset	Frames	Shots	Videos	Sentences
V3C1-PC	1,605,335	219,531	7,475	436,204

1.2 BNL for Negation-Aware Video Retrieval

BNL trains a negation-aware video retrieval model by letting a CLIP model learn from partially negated video descriptions. Such descriptions are automatically constructed as follows. Given a video x^+ and one of its associated original caption q, a negative cue (*e.g.* [not]) is randomly inserted into q, say right before an identified verb or after an auxiliary verb, to construct a partially negated video description q^-, as illustrated in Fig. 2.

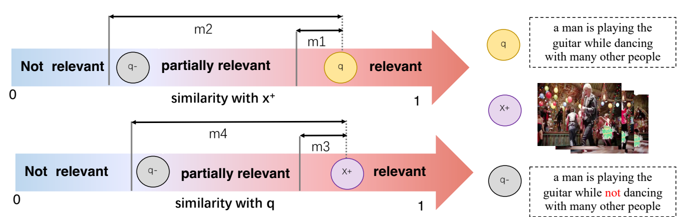

Figure 2: Bidirectional Negation Learning (BNL).

Given the triplet $< x^+, q, q^- >$, the following two bidirectionally constrained losses are calculated:

$$l_{bcl}(x^+, q, q^-) = \max(0, m_1 + s(x^+, q^-) - s(x^+, q)) + \max(0, -m_2 - s(x^+, q^-) + s(x^+, q)), \tag{3}$$

where m_1 and m_2 are lower and upper boundaries to bound the gap between $s(x^+, q^-)$ and $s(x^+, q)$, with $0 < m_1 < m_2 < 2$, and

$$l_{bcl}(q, x^+, q^-) = \max(0, m_3 + s(q, q^-) - s(q, x^+)) + \max(0, -m_4 - s(q, q^-) + s(q, x^+)), \tag{4}$$

where m_3 and m_4 are lower and upper boundaries to bound the gap between $s(q, q^-)$ and $s(q, x^+)$, with $0 < m_3 < m_4 <$

[4] https://github.com/salesforce/BLIP
[5] https://github.com/facebookresearch/VMZ/tree/master/pt
[6] https://github.com/li-xirong/avs

2. The two losses are then weightedly added to a standard retrieval loss:

$$l_{bnl}(q, x^+) = \begin{array}{l} \max(0, m_0 + s(x^{\#}, q) - s(x^+, q)) + \\ \lambda_1(\ell_{bcl}(x^+, q, q^-) + \ell_{bcl}(q, x^+, q^-)), \end{array} \quad (5)$$

where $x^{\#}$ denotes the hardest negative video and λ_1 is a small positive weight for balancing the primary and the auxiliary losses.

We use the BNL loss to retrain CLIP(ViT-B/32) using a negation-enriched version of MSR-VTT [20]. The resultant model, denoted by CLIP-*bnl*, is used in the following two manners:

• As a cross-modal extractor for both video and query representation;

• As a re-ranking module specifically used for queries that have negative cues automatically detected.

1.3 Search Result Reranking

Given a top-ranked list of (5k) videos returned by a base model, we re-score each video in the list by considering a fine-grained cross-modal similairty between its n frames and the given query. The frame-query similarity is computed based on their embeddings obtained by CLIP(ViT-L/14)@336. Max pooling is used to aggregated the frame-level similarities to the video level. The new relevance score is obtained by a weighted linear fusion of the newly computed score (0.6) and the original score (0.4). To better handle queries that have negative cues detected, we use CLIP-*bnl* instead of CLIP(ViT-L/14)@336.

2 Internal Evaluation

According to our experiments on TV16-TV18, V3C1-PC is than MS-COCO for pre-training, see Table 2.

Table 2: Evaluating the influence of pre-training dataset on the TRECVID 16-18 AVS tasks.

Pre-Training Dataset	TV16	TV17	TV18	MEAN
MS-COCO	0.263	0.340	0.174	0.259
V3C1-PC	0.260	0.356	0.187	0.268

To assess the influence of using CLIP-*bnl* for video/text feature extraction and for search result reranking, we evaluate the following three settings on TV16-TV21:

1. LAFF$^{(1)}$: We pre-train LAFF on V3C1-PC and then fine-tune on M+T+V using a set of selected features[7].
2. LAFF$^{(2)}$: Substituting CLIP-*bnl* for CLIP (ViT-B/32) in LAFF$^{(1)}$.
3. LAFF$^{(2)}$ + *Re*: LAFF$^{(2)}$ + search result reranking.

[7] Video features: *wsl*, *beit*,*clip*, *clip(ViT-B/32)*, *blip* and *ircsn*. Text features: *bow*, *w2v*, *clip(ViT-B/32)*,*clip*, and *blip*.

As Table 3 shows, LAFF$^{(2)}$ is better than LAFF$^{(1)}$, suggesting that CLIP-*bnl* is beneficial. Comparing LAFF$^{(2)}$ and LAFF$^{(2)}$ + *Re*, reranking brings in consistent improvement. So we rerank the search results of each base model.

Table 3: Evaluating the influence of CLIP-bnl and Reranking on the TRECVID 16-21 AVS tasks.

Model	TV16	TV17	TV18	TV19	TV20	TV21	MEAN
LAFF$^{(1)}$	0.259	0.35	0.189	0.238	0.359	0.353	0.291
LAFF$^{(2)}$	0.262	0.357	0.193	0.243	0.357	0.363	0.296
LAFF$^{(2)}$+Re	0.276	0.361	0.192	0.255	0.361	0.365	0.302

3 Submissions

We submit the best performance setting[8] on the TV16-TV21 as Run 4. As for Run 3, we remove two heavy text features (*i.e.* bow and w2v) to see whether those heavy text encoders could be removed. As for Run 2, we fuse the results of narrative of queries by LAFF (w/o bow and w2v) and Run3. As for Run 1, we tried a naive query augmentation strategy by automatically appending noun / adjective based keywords at the end of each query. Our four runs are as follows:

- *Run 4*: LAFF
- *Run 3*: LAFF (w/o bow and w2v)
- *Run 2*: Late average fusion of Run3 on test queries and narrative of queries.
- *Run 1*: Late average fusion of multiple augmented query retrieval results.

The performance of our four runs on the TV22 AVS task is shown in Fig. 3. Our best run is Run 2, which with a mean infAP of 0.262, is ranked second teamwise.

Acknowledgments

The authors are grateful to the TRECVID coordinators for the benchmark organization effort [1]. This research was supported by the National Natural Science Foundation of China (No. 62172420, No. 61672523), Beijing Natural Science Foundation (No. 4202033), and the Fundamental Research Funds for the Central Universities and the Research Funds of Renmin University of China (No. 18XNLG19).

References

[1] G. Awad. TRECVID 2022. https://www-nlpir.nist.gov/projects/tv2022/, 2022.

[8] Video features: *wsl*, *beit*,*clip*, *clip-bnl*, *blip* and *ircsn*. Text features: *bow*, *w2v*, ,*clip*,*clip-bnl* and *blip*. Pre-training data: V3C1-PC. Training data: M+T+V

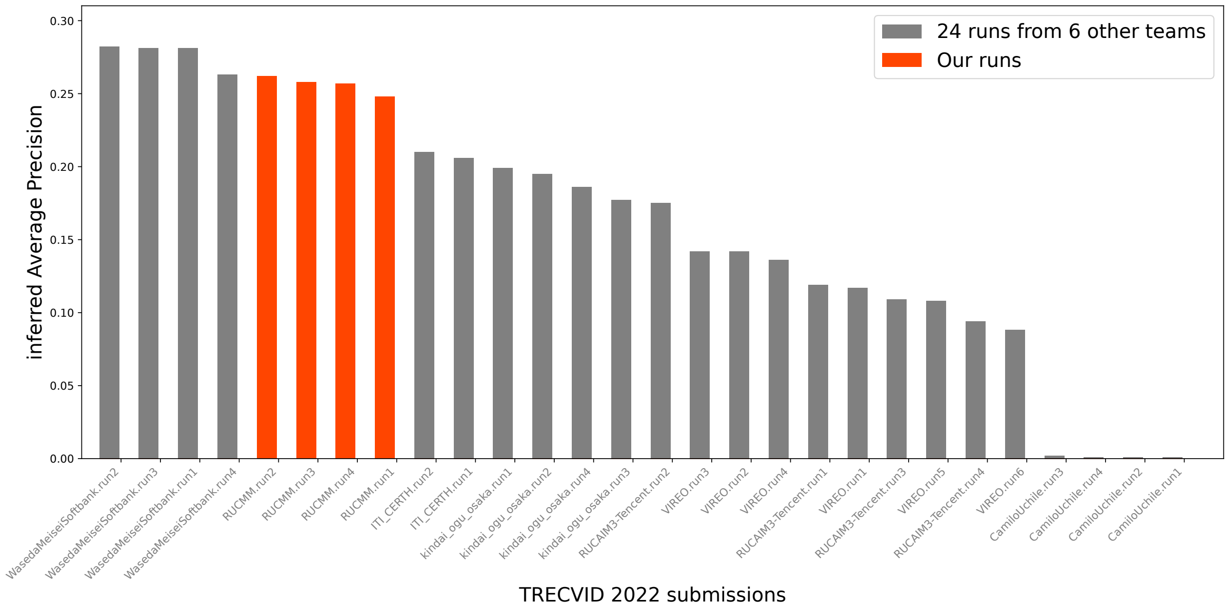

Figure 3: Overview of the TRECVID 2022 AVS benchmark evaluation.

[2] G. Awad, J. Fiscus, D. Joy, M. Michel, A. Smeaton, W. Kraaij, G. Quénot, M. Eskevich, R. Aly, R. Ordelman, G. Jones, B. Huet, and M. Larson. Trecvid 2016: Evaluating video search, video event detection, localization, and hyperlinking. In *TRECVID Workshop*, 2016.

[3] H. Bao, L. Dong, and F. Wei. BEiT: BERT pre-training of image transformers. 2021.

[4] J. Dong, X. Li, and C. G. M. Snoek. Predicting visual features from text for image and video caption retrieval. *IEEE Transactions on Multimedia*, 2018.

[5] D. Ghadiyaram, D. Tran, and M. Feiszli. Large-scale weakly-supervised pre-training for video action recognition. In *CVPR*, 2019.

[6] F. Hu, A. Chen, Z. Wang, F. Zhou, J. Dong, and X. Li. Lightweight attentional feature fusion: A new baseline for text-to-video retrieval. In *ECCV*, 2022.

[7] J. Li, D. Li, C. Xiong, and S. C. H. Hoi. BLIP: bootstrapping language-image pre-training for unified vision-language understanding and generation. In *ICML*, 2022.

[8] X. Li, A. Chen, F. Hu, X. Chen, C. Dong, and G. Yang. Renmin University of china at trecvid 2021: Searching and describing video. In *TRECVID Workshop*, 2021.

[9] X. Li, J. Dong, C. Xu, J. Cao, X. Wang, and G. Yang. Renmin University of China and Zhejiang Gongshang University at TRECVID 2018: Deep cross-modal embeddings for video-text retrieval. In *TRECVID Workshop*, 2018.

[10] X. Li, S. Liao, W. Lan, X. Du, and G. Yang. Zero-shot image tagging by hierarchical semantic embedding. In *SIGIR*, 2015.

[11] X. Li, C. Xu, G. Yang, Z. Chen, and J. Dong. W2VV++: Fully deep learning for ad-hoc video search. In *ACMMM*, 2019.

[12] X. Li, J. Ye, C. Xu, S. Yun, L. Zhang, X. Wang, R. Qian, and J. Dong. Renmin University of china and Zhejiang Gongshang University at TRECVID 2019: Learn to search and describe videos. In *TRECVID Workshop*, 2019.

[13] X. Li, F. Zhou, and A. Chen. Renmin University of China at TRECVID 2020: Sentence encoder assembly for ad-hoc video search. In *TRECVID Workshop*, 2020.

[14] X. Li, F. Zhou, C. Xu, J. Ji, and G. Yang. SEA: Sentence encoder assembly for video retrieval by textual queries. *IEEE Transactions on Multimedia*, 23:4351–4362, 2021.

[15] Y. Li, Y. Song, L. Cao, J. Tetreault, L. Goldberg, A. Jaimes, and J. Luo. TGIF: A new dataset and benchmark on animated GIF description. In *CVPR*, 2016.

[16] D. Mahajan, R. Girshick, V. Ramanathan, K. He, M. Paluri, Y. Li, Y. Bharambe, and L. Maaten. Exploring the limits of weakly supervised pretraining. In *ECCV*, 2018.

[17] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. In *ICLR*, 2013.

[18] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

[19] X. Wang, J. Wu, J. Chen, L. Li, Y. F. Wang, and W. Y. Wang. Vatex: A large-scale, high-quality multilingual dataset for video-and-language research. In *ICCV*, 2019.

[20] Z. Wang, A. Chen, F. Hu, and X. Li. Learn to understand negation in video retrieval. In *ACMMM*, 2022.

[21] J. Xu, T. Mei, T. Yao, and Y. Rui. MSR-VTT: A large video description dataset for bridging video and language. In *CVPR*, 2016.