A Multi-interaction Model with Cross-Branch Feature Fusion for Video-Text Retrieval

Junting Li, Dehao Wu, Yuesheng Zhu(✉), and Zhiqiang Bai

Shenzhen Graduate School, Peking University, Beijing, China
juntingli@stu.pku.edu.cn, {wudehao,zhuys,baizq}@pku.edu.cn

Abstract. With the explosive growth of videos on the internet, video-text retrieval is receiving increasing attention. Most of the existing approaches map videos and texts into a shared latent vector space and then measure their similarities. However, for video encoding, most methods ignore the interactions of frames in a video. In addition, many works obtain features of various aspects but lack a proper module to fuse them. They use simple concatenation, gate unit, or average pooling, which possibly can not fully exploit the interactions of different features. To solve these problems, we propose the Multi-Interaction Model (MIM). Concretely, we propose a well-designed multi-scale interaction module to exploit interactions among frames. Besides, a fusion module is designed to combine representations from different branches by encoding them into various subspaces and capturing interactions among them. Furthermore, to learn more discriminative representations, we propose an improved loss function. And we design a new mining strategy, which selectively reserves informative pairs. Extensive experiments conducted on MSR-VTT, TGIF, and VATEX datasets demonstrate the effectiveness of the proposed video-text retrieval model.

Keywords: Video-text retrieval · Feature interactions · Feature fusion · Loss function

1 Introduction

Since natural language texts contain richer content than keywords, video retrieval with natural language queries has received more attention. Usually, both texts and videos are projected into a latent space via different methods, which still have some limitations. First, most methods do not exploit sufficient inter-frame interactions. HGR [1] uses a weighted sum to get video embeddings, ignoring exploring more inter-frame interactions. Second, many works obtain

This work was supported in part by the National Innovation 2030 Major S&T Project of China under Grant 2020AAA0104203, and in part by the Nature Science Foundation of China under Grant 62006007.

T. Mantoro et al. (Eds.): ICONIP 2021, CCIS 1517, pp. 476–484, 2021.
https://doi.org/10.1007/978-3-030-92310-5_55

features of various aspects but fuse them with simple methods. CE [5] fuses the results of multiple experts by average pooling and gate unit. Third, most loss functions for video retrieval are not flexible enough. The hinge-based triplet ranking loss [7–9] treats all samples equally, ignoring the effect of different samples on optimization. And most loss functions either focus on the hardest negative pair or average all negative pairs. [10,12] The former may cause model affected by outliers, while the latter brings lots of redundancy.

To address the above limitations, we propose the Multi-Interaction Model (MIM). First, we propose a multi-scale inter-frame interactions module (MSIFI) to encode videos. It is implemented by a well-designed convolutional module. It regards each frame feature as a channel and performs 1-D convolution along the feature axis. Through MSIFI, each element of output embeddings comes from all the elements of inputs. Second, a fusion method is designed to merge features from MSIFI, bi-GRU, and global branches sufficiently. It maps the outputs of MSIFI and bi-GRU into different subspaces. Features from all subspaces will interact with each other. Then it is combined with global features via an adaptive gate unit. Third, we propose an improved loss function. It assigns weights to each pair with non-linear functions, whose value changes with the similarity score. Pairs whose similarity scores are far from the optimum will get larger weights and converge faster. Moreover, an adaptive mining strategy is designed to reserve informative samples with different weights. The main contributions of this work are as follows:

- To fully exploit interactions among frames in multi scales, we propose a novel MSIFI module. It utilizes a well-designed convolution operation to learn more accurate and significant information from multi-scale interactions.
- We design a novel fusion module to merge different features. Through sufficient interactions among features from multiple latent subspaces, we integrate features of various aspects and get an accurate video representation.
- Considering the influence of different samples on optimization, we propose an improved loss with a new mining strategy.
- Extensive experiments on several datasets validate the effectiveness of MIM.

2 Related Work

Frame Aggregations for Video-Text Retrieval. HGR [1] decomposes videos to match with texts in different levels and JSFusion [4] encodes all frames of videos with texts and directly predicts the video-text similarities. They both ignore exploring more inter-frame interactions.

Fusion Methods for Video-Text Retrieval. Dual Encoding [9] concatenates the results of multiple encoders. Howto100m [6] aggregates different features by max pooling and concatenation. CE [5] aggregates various information with a gate unit and average pooling.

Loss Functions for Video-Text Retrieval. Most methods [7–9] adopt hinge-based triplet ranking loss or bi-directional max-margin ranking loss [5,6]. However, they treat all samples equally. Circle loss [12] assigns weights to different

pairs with a linear function and Polynomial Loss [10] just considers the hardest negative sample or averages all negative samples, which are not flexible enough.

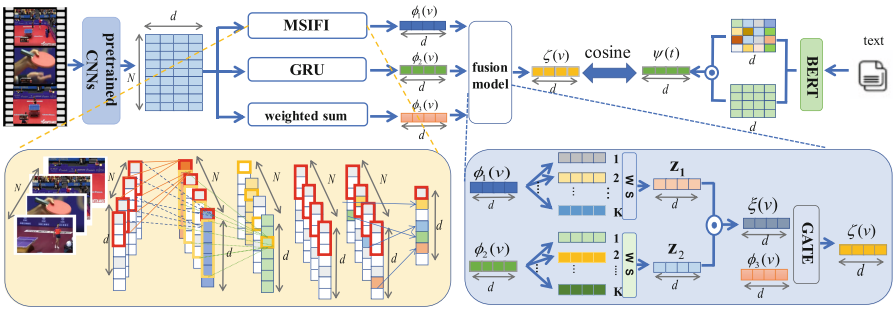

Fig. 1. The architecture of MIM. The video encoder has three branches. The Text encoder contains a multi-dimensional attention module. The MSIFI module captures multi-scaled interactions among frames. The fusion module merges three branches features. N is the number of video frames and the dimension of features is unchanged by proper padding. Details are in Sect. 3. *WS* denotes weighted sum and $\odot$ is Hadamard product.

3 Methodology

Given a video v and a text t, our model encodes them into fixed d-dimensional vectors in a common space. We use the features extracted by pre-trained CNNs [19–21] and BERT [11]. As illustrated in Fig. 1, the video encoder has three branches, whose outputs are denoted as $\phi_1(v)$, $\phi_2(v)$, $\phi_3(v)$. Then they are integrated into $\zeta(v)$ by the fusion module. Text encoder handles text features with a multi-dimensional attention mechanism to get the result $\psi(t)$.

3.1 Multi-scale Inter-frame Interactions (MSIFI) Branch

As shown in the upper-left part of Fig. 1, a video is projected into a matrix $I \in \mathbb{R}^{N \times d}$ by pre-trained CNNs. I is the input of MSIFI and N is the number of frames. Specifically, each element of the feature corresponds to a channel of the last layer in pre-trained CNNs. We treat each frame as one channel of MSIFI and perform the convolution along the feature axis. This actually combines different channels of pre-trained CNNs when sliding our convolutional kernels. As the number of layers increases, the receptive field of each layer is enlarged and it completely covers I in the last layer. In this way, we achieve multi-scale inter-frame interactions and merge significant information from all frames. They are aggregated into $\phi_1(v) \in \mathbb{R}^d$ by max pooling, reserving the most informative features. Each element of $\phi_1(v)$ is derived from the interactions among all frames.

3.2 Temporal Branch and Global Branch

Since temporal information plays an important role in video encoding, we employ the bi-GRU network to capture temporal information. The input is $I \in \mathbb{R}^{N \times d}$ and the output is aggregated into $\phi_2(v) \in \mathbb{R}^d$ by max pooling.

To obtain a more comprehensive video embedding, we also extract the global features. As the significance of frames in a video are different, we assign weights to them based on significance. Each frame $v_i \in \mathbb{R}^d$ is mapped into $\tau_i \in \mathbb{R}$ by a FC layer. The global embedding of the video is the weighted sum of all frames:

$$\phi_3(v) = \sum_{i=1}^{N} \gamma_i v_i, \quad \gamma_i = \frac{exp(\tau_i)}{\sum_{i=1}^{N} exp(\tau_i)}, \tag{1}$$

where $\gamma_i \in \mathbb{R}$ is the weight of the i-th frame and $\phi_3(v)$ represents relatively primitive video information.

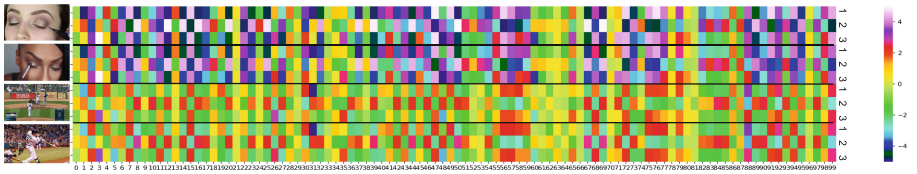

Fig. 2. Visualization of attentions of different videos to K subspaces. Each row denotes the attention of a subspace, and every K rows correspond to a video. We set K = 3. Semantic similar videos have similar attentions. The content of the first two and last two videos are different, so they have different attentions. This indicates different subspaces represent different aspects of video features.

3.3 Fusion Module

To fuse information from three branches, we conduct another kind of interaction between $\phi_1(v)$ and $\phi_2(v)$ and then merge the result with $\phi_3(v)$. As illustrated in the lower-right part of Fig. 1, we first map $\phi_1(v)$ and $\phi_2(v)$ into K subspaces respectively. They are denoted as $\{\boldsymbol{h}^{(k)}\}$ and $\{\boldsymbol{e}^{(k)}\}$, where k represents the k-th subspace. Different subspaces represent different aspects of video features. Figure 2 shows the representations of several videos in K subspaces. Semantic similar videos pay similar attention to certain subspaces, and unrelated videos have different dependencies on each subspace. After that, the representations from all subspaces are aggregated by weighted sum to obtain $\mathbf{z_1} \in \mathbb{R}^d$ and $\mathbf{z_2} \in \mathbb{R}^d$. They are fused into $\xi(v)$ by Hadamard product. The q-th element of $\xi(v)$ is as follow, where $\alpha^{(i)} \in \mathbb{R}$ and $\beta^{(i)} \in \mathbb{R}$ are trainable parameters.

$$\mathbf{z_1} = \sum_{i=1}^{K} \alpha^{(i)} \boldsymbol{h}^{(i)}, \quad \mathbf{z_2} = \sum_{j=1}^{K} \beta^{(j)} \boldsymbol{e}^{(j)}, \quad \xi(v)_q = \sum_{i=1}^{K} \alpha^{(i)} \boldsymbol{h}_q^{(i)} \sum_{j=1}^{K} \beta^{(j)} \boldsymbol{e}_q^{(j)}. \tag{2}$$

It can be seen that the representation from each subspace interacts with representations from all subspaces of another branch. As $\phi_3(v)$ contains global information, an adaptive fusion gate is uesd to mix $\xi(v)$ and $\phi_3(v)$ into $\zeta(v) \in \mathbb{R}^d$:

$$\zeta(v) = \lambda \cdot \xi(v) + (1 - \lambda) \cdot \phi_3(v), \quad \lambda = \sigma(\mathbf{FC_1}(\xi(v))), \tag{3}$$

where $\lambda \in R^d$ denotes the gating weight, $\mathbf{FC_1}$ represents a fully connected layer and σ is the sigmoid function.

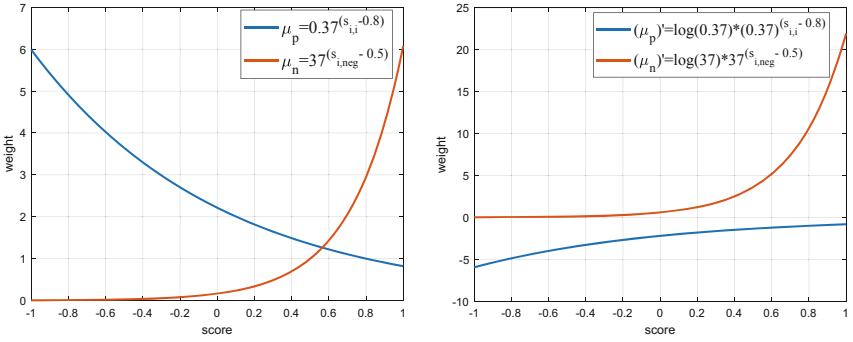

Fig. 3. The weight function curves (left) and their derivative curves (right) of pairs in loss function. Blue curves are for positive pairs and red curves are for negative pairs.

3.4 Text Encoder with Multi-dimensional Attention

Inspired by MAGP [14], we believe that different dimensions attend to different properties and we adopt the text encoder of MAGP. The difference is that we add up the output of every 2 adjacent layers of BERT, and concatenate the results of 6 groups. Then the multi-dimensional attention module obtains attention weights for every word and aggregates them into a vector $\psi(t) \in \mathbb{R}^d$.

3.5 Video-Text Matching

The cosine similarity of $\zeta(v)$ and $\psi(t)$ is their similarity score: $s_{i,j} = \frac{\zeta(v)_i^T \psi(t)_j}{||\zeta(v)_i|| ||\psi(t)_j||}$. $s_{i,i}$ is a positive pair and $s_{i,j}$ is a negative pair, where $i \neq j$. An adaptive mining strategy is used to reserve informative pairs. We select and assign weights to informative pairs while discarding other pairs. All negative samples are sorted based on similarity scores. Harder samples rank higher. Then we save top $\frac{U}{r}$ samples, assign weights, and aggregate them to get the negative pairs representative $s_{i,neg}$ for the i-th query. r is a hyper-parameter, U is the size of one batch.

$$s_{i,neg} = \sum_{j=1, j \neq i}^{\frac{U}{r}} \eta_j s_{ij}, \quad \eta_j = \frac{exp(s_{ij})}{\sum_{j=1, j \neq i}^{\frac{U}{r}} exp(s_{ij})}, \tag{4}$$

Our loss function is as follow, where μ_n and μ_p are the weight functions of negative and positive pairs. Δ is the margin and $[\cdot]_+ = max(\cdot, 0)$, a, b_0 and b_1 are hyper-parameters.

$$L = log\left[1 + \sum_{i=1}^{U}\sum_{q=1}^{U} exp(\mu_n s_{i,neg} - \mu_p(s_{q,q} - \Delta)) \sum_{j=1}^{U}\sum_{k=1}^{U} exp(\mu_n s_{neg,j} - \mu_p(s_{k,k} - \Delta))\right], \tag{5}$$

$$\mu_p = \left[a^{s_{i,i}-\Delta}\right]_+, \quad \mu_n = \left[b_0^{s_{i,neg}-b_1}\right]_+, \tag{6}$$

Table 1. Comparison with state-of-the-arts on MSR-VTT, TGIF and VATEX dataset.

Dataset	Methods	Text-to-Video				Video-to-Text				rsum
		R@1	R@5	R@10	MedR	R@1	R@5	R@10	MedR	
MSR-VTT	VSE++ [8]	8.7	24.3	34.1	28	15.6	36.6	48.6	11	167.9
	W2VV++ [7]	11.1	29.6	40.5	18	17.5	40.2	52.5	9	191.4
	HGR [1]	11.1	30.5	42.1	16	18.7	44.3	57.6	7	204.4
	Dual Encoding [9]	11.6	30.3	41.3	17	22.5	47.1	58.9	7	211.7
	MAGP [14]	13.0	34.7	47.0	12	22.2	48.6	59.8	**6**	225.3
	Ours	**13.6**	**36.0**	**48.3**	**11**	**23.8**	**49.2**	**62.1**	**6**	**233.0**
TGIF	VSE++ [8]	1.6	5.9	9.8	220	1.4	5.6	9.6	192	33.9
	Corr-AE [13]	2.1	7.4	11.9	148	2.2	7.3	11.5	158	42.4
	PVSE [2]	3.0	9.7	14.9	109	3.3	9.9	15.6	115	56.4
	HGR [1]	5.0	13.6	19.4	110	7.2	18.0	24.8	66	88
	MAGP [14]	6.0	15.6	22.1	85	9.1	21.0	28.6	49	102.4
	Ours	**6.8**	**17.3**	**24.2**	**68**	**9.3**	**21.1**	**29.1**	**46**	**107.8**
VATEX	VSE++ [8]	31.3	65.8	76.4	-	42.9	73.9	83.6	-	373.9
	CE [5]	31.1	68.7	80.2	-	41.3	71.0	82.3	-	374.6
	HGR [1]	35.2	73.5	83.4	**2**	45.8	**76.9**	**85.4**	**2**	400.2
	MAGP [14]	34.1	74.6	85.1	**2**	-	-	-	-	-
	Dual Encoding [9]	**36.8**	73.6	83.7	-	46.8	75.7	85.1	-	401.7
	Ours	36.0	**75.4**	**85.2**	**2**	**48.5**	74.7	82.7	**2**	**402.5**

The curves of weight functions and their derivative functions are shown in Fig. 3. Our loss functions satisfy the following characteristics. When the similarity score is far from its optimum, this pair is more informative. The value and derivative value of its weight function will be greater. It means that this pair gets a bigger weight in the loss function and updates at a faster pace, and vice versa.

4 Experiments

4.1 Experimental Settings

Datasets and Metrics. We conduct experiments on MSR-VTT [15], VATEX [16], and TGIF [17]. We use the official partition of MSR-VTT. For

VATEX and TGIF, we follow the experimental setup of HGR [1].The performance is evaluated with common retrieval metrics, namely R@K (Recall at rank K), MedR (Median Rank), MnR (Mean Rank), and rsum (the sum of all recall scores).

Implementation Details. For MSR-VTT, the visual features are extracted by ResNet-152 and ResNeXt-101 pre-trained on ImageNet [9]. For TGIF and VATEX, we use the pre-trained ResNet-152 visual feature and the officially provided I3D [19] visual feature respectively. The MSIFI module has 5 convolutional layers with kernel size = 3,5,5,7,9. The number of subspaces K is 3. The dimension d is 4096. For loss function, we choose hyper-parameters by grid search. We set $r = 20$, $a = 0.37$, $\Delta = 0.8$, $b_0 = 37$, and $b_1 = 0.5$. The model is trained for 20 epochs using Adam optmizer [18] with batch size of 64 and learning rate is 1e−4.

Table 2. Ablation studies on MSR-VTT dataset.

Methods	Text-to-Video Retrieval					Video-to-Text Retrieval					rsum
	R@1	R@5	R@10	MedR	MnR	R@1	R@5	R@10	MedR	MnR	
Ours w/o MSIFI	10.7	30.5	42.6	15	91.6	14.7	37.2	49.9	11	63.4	185.6
Ours-transformer	13.5	35.4	48.0	12	84.8	22.8	47.0	59.7	**6**	44.6	226.4
Ours-gate	13.3	35.7	48.2	**11**	90.8	21.4	46.2	58.8	7	51.7	223.6
Ours-concat	11.9	32.6	44.9	14	94.9	19.7	43.6	56.8	7	53.3	209.5
Ours-CircleLoss	12.7	33.9	46.2	13	94.2	20.2	45.1	58.0	7	51.3	216.1
Ours-MaxPolyLoss	13.0	34.3	46.6	13	97.3	21.9	47.8	60.7	**6**	47.9	224.3
Ours-TripleLoss [8]	11.8	32.5	44.7	14	**84.0**	16.0	39.8	52.6	9	73.9	197.4
Ours-hard	13.4	34.5	46.8	13	92.3	21.7	48.0	61.3	**6**	45.9	225.7
Ours-avg	9.1	26.2	37.3	20	92.1	12.8	33.5	46.7	12	95.8	165.6
Full model	**13.6**	**36.0**	**48.3**	**11**	87.4	**23.8**	**49.2**	**62.1**	**6**	**43.8**	**233.0**

4.2 Comparisons with State-of-the-Arts (SOTAs)

As shown in Table 1. On all datasets, MIM has the highest rsum, demonstrating the advantages of MIM. Specifically, MIM outperforms MAGP. As they have the same text encoder, it proves that our video encoder is more effective. As the features of VATEX are not frame-level features, it is hard to implement inter-frame interactions as sufficiently as on MSR-VTT or TGIF. Our performance on VATEX degrades slightly. Nevertheless, our rsum is still the highest, proving the superiority of our fusion module and loss function.

4.3 Ablation Studies

We conduct ablation studies on MRS-VTT and results are displayed in Table 2.

Effectiveness of MSIFI. We remove MSIFI and compare it with Transformer [3]. To maintain similar number of parameters, we use 1 layer Transformer with 4096 hidden dimensions and 8 attention heads. Results show that MSIFI is effective.

Effectiveness of Fusion Module. We replace the fusion module with gate unit and concatenation respectively. And rsum decreases by 9.4 and 23.5, which proves that our fusion strategy can integrate different features more effectively.

Effectiveness of Loss Function. We compare our loss function with other loss functions and replace the mining strategy with hard mining and average operation. Results confirm the superiority of our loss function and mining strategy.

5 Conclusions

This paper introduces a multi-interaction model for video-text retrieval, with an MSIFI branch to capture multi-scale interactions among videos frames and a fusion method to exploit multiple complementary information between different video features. Moreover, a loss function and a mining strategy are proposed. Extensive experiments show the effectiveness of this approach.

References

1. Chen, S., Zhao, Y., Jin, Q., et al.: Fine-grained video-text retrieval with hierarchical graph reasoning. In: CVPR, pp. 10638–10647 (2020)
2. Song, Y., Soleymani, M.: Polysemous Visual-semantic embedding for cross-modal retrieval. arXiv preprint arXiv:1906.04402 (2019)
3. Vaswani, A., et al.: Attention is all you need. In: NIPS, pp. 5998–6008 (2017)
4. Yu, Y., Kim, J., Kim, G.: A joint sequence fusion model for video question answering and retrieval. In: ECCV, pp. 471–487 (2018)
5. Liu, Y., et al.: Use what you have: video retrieval using representations from collaborative experts. In: BMVC (2019)
6. Miech, A., Zhukov, D., et al.: Howto100m: learning a text-video embedding by watching hundred million narrated video clips. In: ICCV, pp. 2630–2640 (2019)
7. Loko, J., et al.: A W2VV++ case study with automated and interactive text-to-video retrieval. In: MM, pp. 2553–2561 (2020)
8. Faghri, F., et al.: VSE++: improving visual-semantic embeddings with hard negatives. In: BMVC (2018)
9. Dong, J., et al.: Dual encoding for video retrieval by text. In: TPAMI (2021)
10. Wei, J., et al.: Universal weighting metric learning for cross-modal matching. In: CVPR, pp. 13005–13014 (2020)
11. Devlin, J., et al.: Bert: pre-training of deep bidirectional transformers for language understanding. In: NAACL-HLT, pp. 4171–4186 (2019)
12. Sun, Y., et al.: Circle loss: a unified perspective of pair similarity optimization. In: CVPR, pp. 6398–6407 (2020)
13. Feng, F., et al.: Cross-modal retrieval with correspondence autoencoder. In: MM, pp. 7–16 (2014)
14. Wu, D., et al.: Multi-dimensional attentive hierarchical graph pooling network for video-text retrieval. In: ICME (2021)
15. Xu, J., et al.: MSR-VTT: a large video description dataset for bridging video and language. In: CVPR, pp. 5288–5296 (2016)
16. Wang, X., et al.: Vatex: a large-scale, high-quality multilingual dataset for video-and-language research. In: CVPR, pp. 4581–4591 (2019)

17. Li, Y., et al.: TGIF: a new dataset and benchmark on animated GIF description. In: CVPR, pp. 4641–4650 (2016)
18. Kingma, DP., Ba, J.: Adam: a method for stochastic optimization. In: ICLR (2015)
19. Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. In: CVPR, pp. 6299–6308 (2017)
20. Xie, S., et al.: Aggregated residual transformations for deep neural networks. In: CVPR, pp. 1492–1500. (2017)
21. He, K., et al.: Deep residual learning for image recognition. In: CVPR, pp: 770–778 (2016)