WHAT MATTERS: ATTENTIVE AND RELATIONAL FEATURE AGGREGATION NETWORK FOR VIDEO-TEXT RETRIEVAL

Xiaoshuai Hao[1,2] *Yucan Zhou*[2*] *Dayan Wu*[2] *Wanqian Zhang*[1,2] *Bo Li*[2] *Weiping Wang*[2] *Dan Meng*[2]

[1] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, 100049, China
[2]Institute of Information Engineering, Chinese Academy of Sciences, Beijing, 100093, China
{haoxiaoshuai, zhouyucan, wudayan, zhangwanqian,libo, wangweiping,mengdan}@iie.ac.cn;

ABSTRACT

Cross-modal video-text retrieval has been an emerging task due to the rapid growth of user-generated videos on the Internet. Most existing approaches focus on extracting visual feature for the video, while audio and caption on the screen containing rich information are ignored. Recently, the aggregations of multi-modal features in videos boost the benchmark of video-text retrieval. However, since these multi-modal features are high-dimensional and heterogeneous, their intrinsically structural relations have not been attached with enough importance and are often overlooked in previous methods. To address this issue, we propose a novel Attentive and Relational Feature Aggregation Network (ARFAN). Specifically, we introduce the self-attention mechanism to make videos adaptively assign higher weights to the representative modalities. Then, the graph convolutional layers are inserted to capture the relations among the multi-modal features to combine them. Our method achieves 15% and 12.9% relative improvements on R@1 when compared with the state-of-the-art method on MSR-VTT and MSVD datasets, respectively.

Index Terms— cross-modal, video-text retrieval, multi-modal aggregation

1. INTRODUCTION

With the rapid growth of user-generated videos, cross-modal retrieval between video data and natural language descriptions, known as video-text retrieval, has attracted much attention. The goal of video-text retrieval is to retrieve and rank the matching videos in the database according to the text query given by users. The current dominant paradigm for video-text retrieval [1, 2, 3] tries to map the queries and the videos into a joint embedding space, where the semantically-similar texts and videos are much closer to each other and vice versa.

Most existing methods [2, 4, 5, 6] adopt the visual feature to represent videos. However, other rich information in the videos which is effective for video-text retrieval is ignored. Given a query like 'A man is giving a speech about climate change', satisfactory results are difficult to be retrieved without the audio or the caption on the screen.

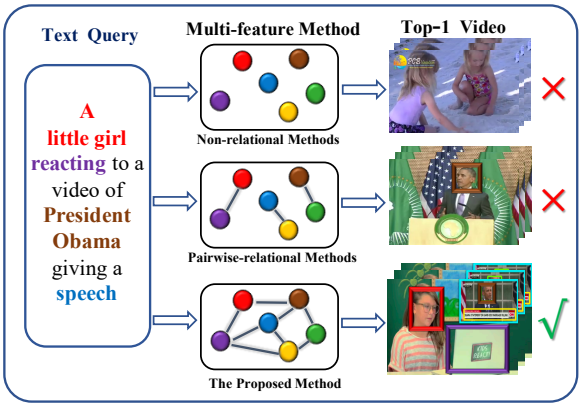

Fig. 1. Illustration of the differences between non-relational methods, pairwise-relational methods and the proposed method. The proposed method considers the structural relations among the multi-modal features and successfully retrieves the correct video given the complex query.

Recently, multi-modal feature aggregation methods greatly boost the benchmark of video-text retrieval, which make use of different features like object, motion, audio, and caption on the screen. However, an interesting observation is that the performance degrades rather than improves, when more and more features are adopted [3]. We address this to the very challenging issue of multi-modal feature aggregation. How to fully exploit knowledge from these high-dimensional features and combine the rich and heterogeneous information in videos is still an open problem, which is also the primary motivation of this paper.

Different from the traditional multi-modal tasks where a common fusion space is learnt for all the samples, multi-feature aggregation in video-text retrieval has its characteristics. Firstly, features act differently to encode different videos. For a meeting video, audios and captions are significant. While, for a game video, objects and motions should be emphasized. Secondly, features are structurally related and work together to meet the query. As illustrated in Figure 1,

∗ Corresponding author:zhouyucan@iie.ac.cn

978-1-6654-3864-3/21/$31.00

given a complex query like 'A little girl reacting to a video of President Obama giving a speech', the speech feature, the face feature 'Obama', and the visual feature 'girl' synergistically promote the target video. While, previous multi-modal feature aggregation methods either measure the similarity of the query and each feature independently or merely consider the pairwise relations, making the confused videos highly ranked.

In this paper, we propose a novel Attentive and Relational Feature Aggregation Network (ARFAN), which can adaptively discover valuable features for different videos and combine the multi-modal features considering their structural relations. Firstly, we employ the self-attention mechanism to learn different weights for the multi-modal features. Thus, videos can assign higher weights to the representative features. Then, to excavate and preserve the intrinsic relations among these features, we further insert the graph convolutional layers into the network, which can capture the global relations among all the features. To show the effectiveness of the proposed ARFAN, we conduct experiments on two benchmark datasets. The main contributions of this work can be summarized as follows:

- We emphasize that the structural relations in the multi-modal features of one video is important for video-text retrieval and thus propose an Attentive and Relational Feature Aggregation Network.
- We introduce the self-attention mechanism to make videos adaptively assign higher weights to the representative modalities. Then, the graph convolutional layers are inserted to excavate intrinsic relations among the multi-modal features to combine them.
- Our method achieves 15% and 12.9% relative improvements on R@1 compared with the state-of-the-art method on MSR-VTT and MSVD datasets, respectively.

2. RELATED WORK

Cross-modal video-text retrieval can be divided into two categories: concept-based approches and embedding-based approaches. Concept-based approaches [7, 8, 9] extract relevant concepts from queries and videos, and accordingly establish associations between these two modalities. The majority of the top-ranked solutions for TRECVID challenge depend on concept-based approaches [8, 9]. However, it is usually ineffective for complex long queries, since it is very difficult to describe the rich sequential information within both videos and queries using a few selected concepts.

Embedding-based approaches [2, 5, 10, 11] try to directly encode videos and texts into a common space. Many of these existing approaches [2, 5] only focus on the visual feature and encode the video inspired by the image-text embedding methods. However, such methods do not take advantage of the rich and various additional information present in videos, such as objects, actions, faces and their combination. Recently, it is widely studied that how to effectively aggregate the multi-modal features available in videos into a compact video representation. Mithun et al. [10] propose the JEMC framework using action, object and audio features by a simple concatenation fusion strategy. Miech et al. [11] propose a new model for learning a joint text-video embedding called Mixture-of-Embedding-Experts (MoEE) that simply adopts more features than JEMC method. Liu et al. [3] further adopt all video features extracted from all modalities and utilize the pairwise relations between two modalities to encode a video. However, we argue that the ignorations of both different impacts of video features and the intrinsic relations among them inevitably degrade the retrieval performance.

3. METHODOLOGY

Given a video V and a query text T, we try to create a pair of functions $\phi(V)$ and $\psi(T)$ mapping videos and texts into a joint embedding space, in which embeddings for matched texts and videos should lie close together, while embeddings for mismatched texts and videos should lie far apart. As illustrated in Figure 2, our proposed framework consist of three components: 1) the video embedding component extracts mulitiple features of videos and obtains fixed-length video feature vectors by temporal aggregation module, and then leverages Attentive and Relational Feature Aggregation module to adaptively discover valuable features for different videos and combine the multi-modal features considering their structural relations. 2) the text embedding component encodes the query sentence into a single vector representation, and then projects it to separated subspaces for each video feature. 3) the similarity estimation component learns the similarity between video and text.

3.1. Video Embedding

Feature Extraction. In order to make full use of the information in one video, we draw on a collection of pre-trained models to extract different video features. These operations on the video project the video to a collection of N video feature embeddings $\left\{I_{var}^{(1)}, ..., I_{var}^{(N)}\right\}$. $I_{var}^{(i)}$ represents the i^{th} video feature (subscript *var* denotes a variable-length output when applied to a sequence of frames). In this paper, we set N=6, and extract features for object, motion, audio, speech, OCR and face. We use the features publicly released by [3]. Each element of this collection is then aggregated along its temporal dimension, producing fixed-length video feature embeddings per video $\left\{I^{(1)}, .., I^{(N)}\right\}$. For temporal aggregation function, we adopt a simple approach to aggregate the features described above. For object, motion and face embeddings, we average frame-level features along the temporal dimension to produce a single feature vector per video.

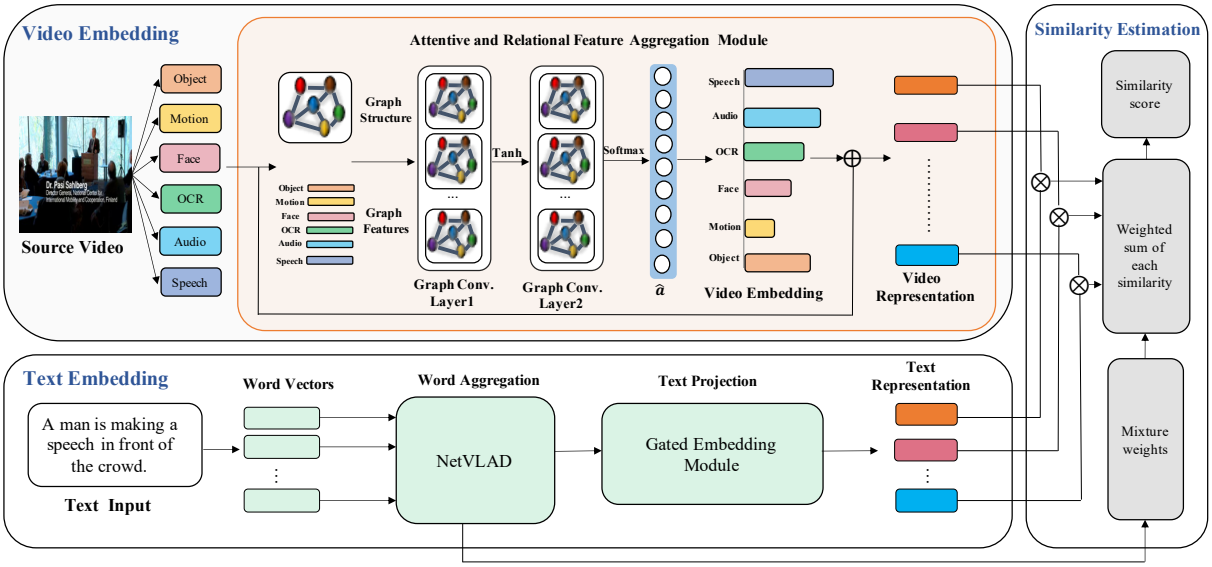

Fig. 2. The Framework of Attentive and Relational Feature Aggregation Network for cross-modal video-text retrieval.

For speech, audio and OCR features, we adopt the NetVLAD mechanism proposed by [12], which has proven effective for the retrieval [3]. Next, we apply linear projections to transform these time-aggregated embeddings to a common dimensionality. Thus, these projected video feature embeddings can be written as:

$$\boldsymbol{H} = \{h_1, h_2, ..., h_N\}. \tag{1}$$

Attentive and Relational Feature Aggregation Module. After extracting the multi-modal feature embeddings for each video, there comes the problem that how to combine them to represent the video content. Previous multi-modal feature aggregation methods either deal with each feature independently or simply model the pairwise relations, which inevitably leads to an unsatisfactory performance.

To combine multiple features into the same dimension, a straightforward way is to assign a weight to each feature, then the weighted sum is the combined feature embedding. Here we follow this paradigm. However, different from the traditional fusion methods where all the samples share the same weights to combine their multiple features, the combination weights for videos should be adaptive. Therefore, we adopt two successive fully-connected layers as the attention weighting function. Specifically, we apply a fully-connected layer $W_1 \in \mathcal{R}^{F\times k}$ with non-linear operation $tanh(\cdot)$ to $\boldsymbol{H}$, producing $tanh(\boldsymbol{H}W_1)$. Then, we apply another fully-connected layer $W_2 \in \mathcal{R}^{k\times 1}$ followed by a softmax layer to obtain the N-dim weight vector $\mathbf{a}$ for $\boldsymbol{H}$, that is:

$$\mathbf{a} = softmax\left(tanh\left(\boldsymbol{H}W_1\right)W_2\right). \tag{2}$$

However, existing attention methods only rely on $\boldsymbol{H}$ to guide the weight vector learning, which ignores the structural relations among multiple feature vectors in each video. To take such relations into consideration, we insert graph convolutional layers into the attention mechanism, which can excavate and preserve the relations among multiple feature embeddings in one video to make them more discriminative.

Graph convolutional layers are originally proposed for semi-supervised learning and now we employ it for the weight vector learning. Inspired by [13], we calculate the similarity graph $\mathbf{S}$ for each video $\boldsymbol{H} = \{h_1, h_2, ..., h_N\}$ during the preprocessing, in which S_{ij} is the cosine similarity between h_i and h_j. Besides, we define $S' = S + I_N$ and a diagonal matrix D with $D_{ii} = \sum_j S'_{ij}$. Then, the graph convolutional layer can be represented by a 1×1 convolution layer with parameters $\bar{S} = D^{-1/2}S'D^{-1/2}$. Finally, we insert two graph convolutional layers into Equation (2), and the ARFA can be written as:

$$\hat{\mathbf{a}} = softmax\left(\bar{S}tanh\left(\bar{S}\boldsymbol{H}W_1\right)W_2\right). \tag{3}$$

The generated $\hat{\mathbf{a}}$ is expected to be more discriminative than $\mathbf{a}$, different videos focus on different video characteristics, thus the video representation should been discriminative. After the weight vector $\hat{\mathbf{a}}$ has been computed, each feature embedding is calculated with:

$$h'_i = \hat{a}_i h_i + h_i. \tag{4}$$

By assigning different weights to different video features, we can obtain discriminative video representations. The final video embedding is then obtained by passing the modulated representations of each video feature embeddings through a Gated Embedding Module (GEM) (precise details in [11]) before concatenating the outputs together into a single fixed-length vector.

The advantage of our ARFA over the state-of-the-art CE [3] is obvious. CE only makes use of the collection of features

extracted from multiple modalities and the pairwise relations between each two features. The ignoration of the structural and global relations among all these features leads to a decreased performance when compared with our method. Extensive comparisions are reported in the Section 4.

3.2. Text Embedding

Given a query sentence, denoted T, we first propogate each words into the word2vec [14] model trained by Google News to achieve their word embeddings. Then, all word embeddings are passed through a pre-trained OpenAI-GPT model to extract the context-specific word embeddings. These word embeddings are then aggregated into a single vector $f(T)$ representing the entire sentence using a NetVLAD [12] aggregation module. Subsequently, this vector $f(T)$ is used to predict the mixture weights (described in the next subsection). After the aggregation, we project the aggregated feature $f(T)$ to separated subspaces for each video feature using GEM (precise details in [11]). The text representation then consists of N embeddings, represented by $\psi\left(T\right) = \left\{\psi^i\right\}_{i=1}^{N}$.

3.3. Similarity Estimation

We compute our final video-text similarity score $s(T, V)$ as the weighted sum of each video-text similarity $s_i(\psi^i, h_i')$, which is calculated as:

$$s(T,V) \quad = \quad \Sigma_{i=1}^{N} w_i(T) s_i(\psi^i, h_i'), \tag{5}$$

where $w_i(T)$ represents the weight for the ith video feature embedding. To obtain these mixture weights, we adopt the strategy in [11] to process the text representation $f(T)$ through a NetVLAD module and then perform a softmax operation as:

$$w_i(T) \quad = \quad \frac{e^{f(T)^\top b_i}}{\Sigma_{j=1}^{N} e^{f(T)^\top b_j}}, \tag{6}$$

where $(b_1, ..., b_N)$ are learnt parameters used to obtain the mixture weights. The intuition behind using a weighted sum is that a text provides a prior on which of the feature embedding should be more important to compute the final similarity score. Note that $w_i(T)$, ψ^i and h_i' can all be precomputed offiline for each text and each video, thus the retrieval only involves dot product operations.

3.4. Training

To train the model, we adopt the bi-directional hard-negatives ranking loss [2, 5]:

$$\begin{aligned} l \; = \; \frac{1}{B} \sum_{i=1}^{B} \sum_{j \neq i} [max\left(0, \; m + \widehat{s_{i,j}} - s_{i,i}\right) \\ + max\left(0, \; m + \widehat{s_{j,i}} - s_{i,i}\right)], \end{aligned} \tag{7}$$

where B is the batch size, $s_{i,j} = s(T_i, V_j)$ is the similarity score of query sentence T_i and video V_j, and m is the margin value for the pairwise ranking loss. $\widehat{s_{i,j}}$ and $\widehat{s_{j,i}}$ respectively indicate a negative text sample for V and a negative video sample for T.

4. EXPERIMENTS

4.1. Experimental Settings

Datasets We present experiments on two benchmark datasets:MSR-VTT Dataset [18] and MSVD [19] to evaluate the performance of our proposed framework. The MSR-VTT contains 10,000 video clips. The dataset is split into 6,513 videos for training, 2,990 videos for testing and 497 videos for the validation set. Each video has 20-sentence descriptions. The MSR-VTT is the most widely used dataset for video-text retrieval. The MSVD dataset contains 1,970 Youtube clips, and each video is annotated with about 40 sentences. For a fair comparison, we use the same splits utilized in prior works [3], with 1,200 videos for training, 100 videos for validation, and 670 videos for testing. All the sentences associated with videos are used.

Evaluation Criteria We use the standard evaluation criteria in most prior work on the video-text retrieval task. We measure rank-based performance by R@K and Median Rank (Med R). R@K is the percentage of test queries for which at least one relevant item is found among the top-*K* retrieved results. In this paper, we report results for R@1, R@5 and R@10. Med R is the median rank of the first relevant item in the search results. Results with higher R@K and lower Med R are better performance. We also report the sum of R@1, R@5 and R@10 as Sum of Recalls in the tables.

4.2. Implementation Details

We adopt the Adam optimizer for all our experiments, set the margin of the bi-directional hard-negatives ranking loss to 0.3, and set k to 512. All aggregated video features are projected to the same size of 768 before fed into ARFA module (i.e., F=768). For MSR-VTT, we train the model with batch size of 64, learning rate of 0.01 and weight decay of 5E-5. For MSVD, we set batch size to 16, learning rate to 0.001 and weight decay to 5E-5. The model is evaluated on the validation set after every epoch, and of which the best sum of recalls on the validation set is chosen as the final model.

4.3. Compared with state-of-the-art

With the same setting and data partition, we compare our proposed ARFAN method with some existing state-of-the-art methods to verify the effectiveness. Video-text retrieval approaches can be divided into two categories: single-feature methods and multi-feature methods. For single-feature methods, we compare with VSE [15], VSE++ [15], W2VV [1],

Table 1. Video-to-Text and Text-to-Video search results on the MSR-VTT dataset.

Method	Text-to-Video Search				Video-to-Text Search				RSum
	R@1	R@5	R@10	Med R	R@1	R@5	R@10	Med R	
Single-feature method									
VSE [15]	5.0	16.4	24.6	47	7.7	20.3	31.2	28	105.2
VSE++ [15]	5.7	17.1	24.8	65	10.2	25.4	35.1	25	118.3
W2VV [1]	6.1	18.7	27.5	45	11.8	32.1	42.4	16	138.6
Dual Encoding [2]	7.7	22.0	31.8	32	13.0	30.8	43.3	15	148.6
CVTR [6]	7.8	23.2	33.5	28	13.1	29.6	41.8	17	149.0
HGR [5]	9.2	26.2	36.5	24	15.0	36.7	48.8	11	172.4
Multi-feature method									
JEMC [10]	7.0	20.9	29.7	38	12.5	28.9	39.1	21	138.1
Simple Concatenation	9.4	26.9	37.9	20	15.1	38.0	51.0	10	178.3
MoEE [11]	9.7	28.7	40.6	17	14.8	40.9	54.8	8	189.5
CE [3]	10.0	28.8	40.4	16	16.5	43.5	56.8	7.5	196.0
ARFAN	**11.5**	**31.3**	**42.8**	**15**	**19.9**	**49**	**62.4**	**6**	**216.9**

Table 2. Video-to-Text and Text-to-Video search results on the MSVD dataset.

Method	Text-to-Video Search				Video-to-Text Search				RSum
	R@1	R@5	R@10	Med R	R@1	R@5	R@10	Med R	
Single-feature method									
ST [16]	2.6	11.6	19.3	51	2.99	10.9	17.5	77	64.89
LJRV [17]	7.7	23.4	35.0	21	9.85	27.1	38.4	19	141.45
VSE [15]	12.3	30.1	42.3	14	15.8	30.2	41.4	12	172.1
VSE++ [15]	15.4	39.6	53.0	9	21.2	43.4	52.2	9	224.8
CVTR [6]	18.4	46.5	61.0	7	22.8	45.1	57.0	7	250.8
Multi-feature method									
JEMC [10]	20.3	47.8	61.1	6	**31.5**	51.0	61.5	5	273.2
Simple Concatenation	18.2	45.1	60.4	7	20.6	48.2	59.0	6	251.5
MoEE [11]	19.1	46.9	62.4	6	23.1	51.9	62.8	5	266.2
CE [3]	19.3	47.2	62.6	6	23.4	50.4	61.5	5.5	264.4
ARFAN	**21.8**	**51.6**	**66.3**	**5**	24.5	**53.4**	**64.9**	**4.5**	**282.5**

dual encoding [2], HGR [5], CVTR [6], LJRV [17], and ST [16]. Besides, we also compare with several multi-feature methods, including JEMC [10], Simple Concatenation, MoEE [11],and CE [3]. The simple concatenation method connects multi-modal features to a single high-dimensional embedding, followed by a GEM. For a fair comparison, all the multi-feature aggregation methods utilize the same video features. Moreover, we directly report the results of single-feature methods from corresponding papers, while re-implement multi-feature methods according to the authors' instructions. Note that, we also directly report the results of JEMC [10] since their method is based on an ensemble of several models, and it is very difficult to exactly re-implement the details.

Table 1 and Table 2 show the overall performance evaluation results of ARFAN and all the baselines on MSR-VTT and MSVD datasets, respectively. We can see that our proposed method performs best and consistently outperforms state-of-the-art methods in both text-to-video and video-to-text retrieval. The proposed ARFAN approach achieves 15%, 8.7% and 6% relative improvements compared with the current best method CE in R@1, R@5 and R@10 on MSR-VTT dataset, respectively. Similarly, ARFAN achieves 12.9%, 9.3% and 5.9% relative improvements compared with CE method in R@1, R@5 and R@10 on MSVD dataset respectively. In a nutshell, ARFAN verifies the effectiveness of considering the relations among the multi-modal features.

4.4. Ablation Study

In order to further explore the effectiveness of the proposed ARFAN method, we conduct an ablation study as follows.

ARFAN (w/o. ARFA) is the variant of ARFAN method which handles each feature independently. **ARFAN (uniform weight)** is the variant of ARFAN method which assigns uniform weights to features in each video instead of learning weights using ARFA module, i.e., equivalently sets $h'_i = \frac{1}{N} h_i + h_i$ in equation (4). **ARFAN (w/o. graph)** is the variant of ARFAN method which replaces the ARFA module with the self-attention mechanism.

By taking the MSR-VTT dataset as an example, we show the experimental results of ablation study in Table 3. By an-

Table 3. Video-to-Text and Text-to-Video retrieval results of the ablation study on the MSR-VTT dataset.

Method	Text-to-Video Search				Video-to-Text Search				RSum
	R@1	R@5	R@10	Med R	R@1	R@5	R@10	Med R	
ARFAN(w/o ARFA)	9.5	28.0	39.9	17	15.2	39.8	53.0	9	185.4
ARFAN(uniform weights)	10.6	29.3	41.0	17	18.2	44.6	56.8	7	200.5
ARFAN(w/o graph)	11.2	30.8	42.7	15	17.5	43.9	58.6	7	204.7
ARFAN	**11.5**	**31.3**	**42.8**	**15**	**19.9**	**49.0**	**62.4**	**6**	**216.9**

alyzing the results of ablation study, we can find the following observations: The results of ARFAN (uniform weights) are superiority over ARFAN (w/o. ARFA) method, indicating the benefit of considering the relation among multi-modal features. The results of ARFAN (w/o. uniform weights) are worse than ARFAN (w/o. graph), which proves the benefit of making videos adaptively assign higher weights to the representative modalities. Sums of recalls of ARFAN method is increased from 204.7 to 216.9 compared to ARFAN(w/o. graph), which shows the advantage of inserting graph convolutional layers to capture intrinsic relations among the multi-modal features. Moreover, the ARFAN (full) outperforms all three variants on MSR-VTT dataset, which indicates that the ARFA module plays an essential role and obtains a great performance in the video-text retrieval task.

5. CONCLUSION

In this paper, we have proposed a novel attentive and relational feature aggregation network (ARFAN) to deal with the high-dimensional and heterogeneous multi-modal features in the videos to promote the video-text retrieval. The graph convolutional layers work together with the self-attention mechanism to capture intrinsic relations among the multi-modal features and make the combination adaptively. Experiments on MSR-VTT and MSVD datasets have demonstrated that our method achieves significant improvements compared to the state-of-the-art researches.

6. ACKNOWLEDGEMENT

This work was supported by the National Natural Science Foundation of China under Grants 62006221, the National Key Research and Development Program of China (No. 2018YFC0825102, No.2019YFC0850202).

7. REFERENCES

[1] Jianfeng Dong, Xirong Li, and Cees G. M. Snoek, "Predicting visual features from text for image and video caption retrieval," *TMM*, pp. 3377–3388, 2018.

[2] Jianfeng Dong, Xirong Li, Chaoxi Xu, Shouling Ji, Yuan He, Gang Yang, and Xun Wang, "Dual encoding for zero-example video retrieval," in *CVPR*, 2019, pp. 9346–9355.

[3] Y. Liu, S. Albanie, A. Nagrani, and A. Zisserman, "Use what you have: Video retrieval using representations from collaborative experts," in *BMVC*, 2019.

[4] Jianfeng Dong, Xirong Li, and Cees G M Snoek, "Word2visualvec: Image and video to sentence matching by visual feature prediction," *arXiv:1604.06838*, 2016.

[5] Shizhe Chen, Yida Zhao, Qin Jin, and Qi Wu, "Fine-grained video-text retrieval with hierarchical graph reasoning," in *CVPR*, 2020.

[6] Zheng Li, Caili Guo, Bo Yang, Zerun Feng, and Hao Zhang, "A novel convolutional architecture for video-text retrieval," in *ICME*, 2020.

[7] Shuang Wu, Sravanthi Bondugula, Florian Luisier, Xiaodan Zhuang, and Pradeep Natarajan, "Zero-shot event detection using multi-modal fusion of weakly supervised concepts," in *CVPR*, 2014, pp. 2665–2672.

[8] Foteini Markatopoulou, Damianos Galanopoulos, Vasileios Mezaris, and Ioannis Patras, "Query and keyframe representations for ad-hoc video search," in *ICMR*, 2017, pp. 407–411.

[9] Tetsunori Kobayashi, "Improving semantic video indexing: Efforts in waseda trecvid 2015 sin system," in *ICASSP*, 2016.

[10] Niluthpol Chowdhury Mithun, Juncheng Li, Florian Metze, and Amit K Roychowdhury, "Learning joint embedding with multimodal cues for cross-modal video-text retrieval," in *ICMR*, 2018, pp. 19–27.

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

[12] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic, "Netvlad: Cnn architecture for weakly supervised place recognition," in *CVPR*, 2016, pp. 5297–5307.

[13] Max Welling Thomas N Kipf, "Semi-supervised classification with graph convolutional networks," in *ICLR*, 2017.

[14] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean, "Efficient estimation of word representations in vector space," in *ICLR*, 2013.

[15] Fartash Faghri, David J Fleet, Jamie Ryan Kiros, and Sanja Fidler, "Vse++: Improving visual-semantic embeddings with hard negatives," in *BMVC*, 2018.

[16] Ryan Kiros, Ruslan Salakhutdinov, and Zemel, "Unifying visual-semantic embeddings with multimodal neural language models," in *NeurIPS*, 2014.

[17] Naokazu Yokoya, "Learning joint representations of videos and sentences with web image search," in *ECCV*, 2016.

[18] Jun Xu, Tao Mei, Ting Yao, and Yong Rui, "Msr-vtt: A large video description dataset for bridging video and language," in *CVPR*, 2016, pp. 5288–5296.

[19] David L Chen and William B Dolan, "Collecting highly parallel data for paraphrase evaluation," in *ACL*, 2011, pp. 190–200.