Cross-Modality Video Segment Retrieval with Ensemble Learning

Xinyan Yu, Ya Zhang and Rui Zhang

Abstract Jointly modeling vision and language is a new research area which has many applications, such as video segment retrieval and video dense caption. Compared with video language retrieval, video segment retrieval is a novel task that uses natural language to retrieve a specific video segment from the whole video. One common method is to learn a similarity metric between video and language features. In this chapter, we utilize ensemble learning method to learn a video segment retrieval model. Our ensemble model aims to combine each single-stream model to learn a better similarity metric. We evaluate our method on the task of the video clip retrieval with the new proposed Distinct Describable Moments dataset. Extensive experiments have shown that our approach achieves improvement compared with the result of the state-of-art.

Keywords Video segment retrieval · Ensemble learning

1 Introduction

In the past few years, cross-modal retrieval has drawn more attention due to the rapid development of the Internet. Cross-modal retrieval is a kind of retrieval method which involves data from different modalities. It takes data from one modality as a query to retrieve data from another modality. Traditional retrieval methods only utilize single modal data. For example, if we use language query to search our interested videos on the Internet, the language query is only used to match the video caption. However, cross-modal retrieval can directly retrieve the elements in the video, such as actors,

X. Yu · Y. Zhang (✉) · R. Zhang (✉)
Cooperative Medianet Innovation Center, Shanghai Jiao Tong University,
Minhang, China
e-mail: ya_zhang@sjtu.edu.cn

X. Yu
e-mail: yuxinyan@sjtu.edu.cn

R. Zhang
e-mail: zhang_rui@sjtu.edu.cn

R. Singh et al. (eds.), *Domain Adaptation for Visual Understanding*,
https://doi.org/10.1007/978-3-030-30671-7_5

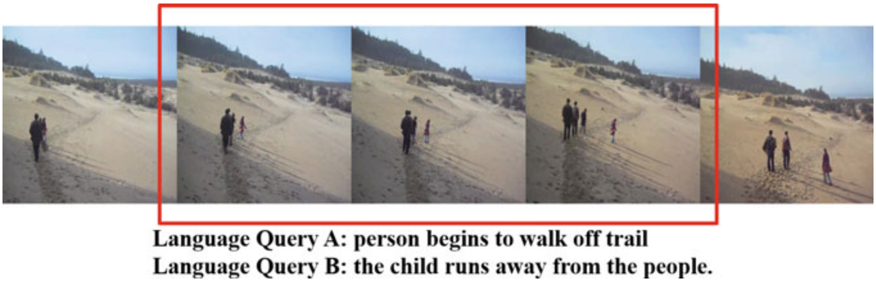
Language Query A: person begins to walk off trail
Language Query B: the child runs away from the people.

Fig. 1 Video segment retrieval is a task to retrieve a video segment from the entire video via language query. The video segment in red rectangle corresponds to the language query below. Though both language description A and B describe the same video segments at the same time, they are constructed with different words and depict the clip in different description perspectives. Description A describes the movement of the crowd in the video as a whole, and the description B depicts the movement of a specific person

actions, and objects. Therefore, cross-modal retrieval can help users to search for information in a more effective way.

In this chapter, we study a novel cross-modal retrieval task which connects video clips with natural language description. Different from traditional video language retrieval that focuses on finding the matched entire video with a given description, we want to retrieve a specific video segment from the entire video with a description. The difficulty to solve this problem is not only from the differences between each modality but also from the differences within each modality. Natural language is usually complicated and ambiguous. As shown in Fig. 1, one video segment can be described in totally different ways by two viewers. These two descriptions may be hardly considered to describe the same video scene if we only give these two sentences to another viewer. Language query A and B depict the video segment in different perspectives. Query A describes the movement of the crowd in the video while query B depicts the movement of a little child in the crowd. Although sometimes these two descriptions have the same meaning, they are not entirely made up of the same words, but of many synonyms. So it is hard to learn a suitable similarity metric to retrieve video segments with the corresponding language query.

To solve this novel and challenging problem, we utilize ensemble learning to guide the aggregation of a multi-stream cross-modal retrieval model. Ensemble learning is a widely used algorithm which combines multiple models to improve the model performance. To learn a better similarity metric for retrieval task with ensemble learning, we propose a novel method which integrates ensemble learning to guide the aggregation of multi-stream retrieval model. We conduct our experiments over the Distinct Describable Moments (DiDeMo) dataset which consists of more than 10,000 untrimmed videos with an explicit video segment caption and corresponding time stamps.

We mainly contribute in the following aspects:

- We propose a multi-stream model to retrieve the specific video segments from the entire video via text query. Multi-stream model can learn multiple common spaces for vision and language features. It could improve the learned similarity metric with ensemble learning. The combination of the ensemble model is guided by a language-based aggregation module.
- We conduct experiments on Distinct Describable Moments dataset and assess the proposed method on top-1 recall (recall@1), top-5 recall (recall@5), and mean intersection over union (mIoU). The results demonstrate that our proposed method outperforms the state-of-art.

The remainder of this chapter is structured as follows: Sect. 2 introduces related work in recent years about vision and language understanding. Section 3 gives the detail of the proposed cross-modal retrieval model. Section 4 details the experimental index, experimental setup, and experiment results. Finally, Sect. 5 concludes our work.

2 Related Work

Localizing moments in a video with natural language is a new research task which jointly models visual and language information. This task is related to both vision and language understanding.

2.1 Vision Understanding

Convolution neuron network (ConvNets) has become the most effective and widely used visual features extractor since [10] won the ImageNet Large Scale Visual Recognition Challenge (ILSVRC). Their results significantly reduced top-5 error compared with the second place. Many of the following researches [6, 18, 19] focused on improving the image recognition accuracy through increasing the depth and width of the deep network. Inspired by the success of ConvNets in the image domain, various pretrained ConvNets are transferred to extract features from the videos for video recognition. However, compared with the still image which only has appearance information, the video consists of multi-frames and has motion information between frames. Therefore, it is not suitable to directly use ConvNets trained on still image to extract video features for the lack of motion information. To integrate the motion into ConvNets, [17] used two-stream networks to model appearance and motion simultaneously. Orthogonal to the two-stream method, [21] exploited the 3D convolution kernel to concrete the spatial and temporal information across the convolution layers.

2.2 Language Understanding

Natural language processing is one of the important technologies in artificial intelligence because language is the tool for people to communicate with each other. There are also many practical natural language applications in daily life, such as semantic analysis and language translation.

Learning high-quality distributed vector representations is the most fundamental and important work in NLP task systems. Reference [13] proposed Word2Vec model to learn embedding representation. They used the correlation of source context words and the target word to model the syntactic and semantic relationship between word sequences. Due to the simple model architecture, their Continuous Bag of Words (CBOW) and Skip-gram models were efficiently trained with one trillion words. Different from the predictive-based model, GloVe [15] learned geometrical embedding vectors of words based on co-occurrence counts. This method preserved the semantic analogies and also took the corpus word occurrence statistics into consideration. To keep the ordering and semantic meaning simultaneously, [11] proposed an unsupervised learning method to learn continuous distributed vector representations for sentence and document. In this chapter, we use GloVe trained on Wikipedia corpus as our word embedding method.

2.3 Cross-Modal Understanding

Despite deep learning having been widely used and achieving success in vision and language task individually, it is still a challenge to jointly understand vision and language. Previous work has focused on tasks, such as image/video caption, image/video retrieval, and video question answering.

Early work on image caption usually used two-stage pipeline to generate sentences from still image. The semantic content is identified in the first stage and then used to generate a sentence using a language template. This two-stage pipeline simplified image caption task to only generate sentence related with some given objects and actions. Though the category of objects and actions should be elaborately selected, the limited number of categories is insufficient to model the complex sentence in the real world. Reference [24] changed this template-based model to a decoder–encoder structure. They first used deep convolution network to extract visual features from still image and then decode the fixed-length word embedding vector using Long Short-Term Memory Network (LSTM) to generate image description. Inspired by the success of this work, [23] introduced the end-to-end structure to the video caption. The difference between image caption and video caption is how to exploit temporal information of the video. To model temporal information of the video into description generation, LSTM could be used both as an encoder and decoder to generate the video caption.

Image/video-sentence retrieval is a cross-domain retrieval task. The core idea is to find the most related instance via the query from another domain. The query can be either image/video or semantic description. The common pipeline for cross-domain retrieval task is to first extract instance features from each domain and then do metric learning to narrow their similarity. Reference [3] leveraged the meaningful semantic label to improve the image classification model. They computed the similarity between joint representation of images and labels to help predict novel classes never before observed.

Reference [8] proposed the Deep Visual-Semantic Alignment (DVSA) model. They used R-CNN [5] object detector to extract image features and bidirectional LSTMs to encode sentence features. Instead of directly mapping the vision and semantic features into the common space, [9] proposed a finer-level bidirectional retrieval model that embeds the fragment of images and fragment sentences into the common space. Reference [27] integrated canonical correlation analysis (CCA) which is a traditional method for cross-modal retrieval into the deep network to match image and text. Reference [26] researched the domain structure in image–text embedding. They combined structure-preserving loss function with a bi-ranking loss to constrain the structure in each domain. Reference [12] proposed multimodal convolution network (M-CNN) to exploit the intermodal relations. They composed sentences to different-level semantic fragments to match the image. Reference [14] utilized visual and textual attention mechanisms to extract essential information from vision and language. Their dual-path attention model captured the fine-grained interplay between vision and language. Reference [22] advocated for learning a visual-semantic hierarchy over image and language.

Reference [16] collected a novel movie dataset with aligned text description—Large Scale Movie Description Challenge (LSMDC). Reference [20] studied order-embedding in joint language-visual neural network model architectures for the video text retrieval. Reference [28] proposed a high-level concept word detector and developed a semantic attention mechanism to selectively match the language description with video cue. Though many efforts have been made for video language retrieval, a few people work on localizing moments of the video via natural language query. The main obstacle for the video moment retrieval is lack of fine-grained video annotation that contains both language description and time stamps. Reference [7] collected over 10,000 unedited, personal videos and annotated video segments with referring expression. Reference [4] added sentence temporal annotations to Charades, a video dataset which consists of daily dynamic scenarios. They addressed the video segment retrieval task by using an object detection framework.

3 Methods

In this section, we introduce our multi-stream video language retrieval model and explain how to use the language information to ensemble each stream.

3.1 Model Overview

Generally, a cross-model retrieval includes two modal inputs, V and S. In our formulation, V represents the video clips and S represents the natural language. The goal of the retrieval model is to find a common embedding space for V and S. We could adopt metric learning method to learn each embedding functions $F(\cdot)$ and $G(\cdot)$. The entire cross-modal retrieval model M could be trained end-to-end with the following objective:

$$\hat{M} = \underset{M}{\arg\min}\, D_\theta\left(F(V), G(S)\right)$$

where $D_\theta(\cdot)$ is a distance function which is used to measure the similarity between projected features of different domains. Cosine distance and Euclidean distance are two common distance functions used in the retrieval task.

In our work, we still retain the idea of projecting two domain features into the same common space. To learn a better similarity metric, we utilize the language information to aggregate the multi-stream retrieval network.

The overview of our proposed model is shown in Fig. 2. Each stream in the whole model is a basic cross-modal retrieval model which tries to project features in different domains to the same common embedding space. Then we use a language-based aggregation module to obtain the final cross-modal distance. Details of the individual modules are shown below.

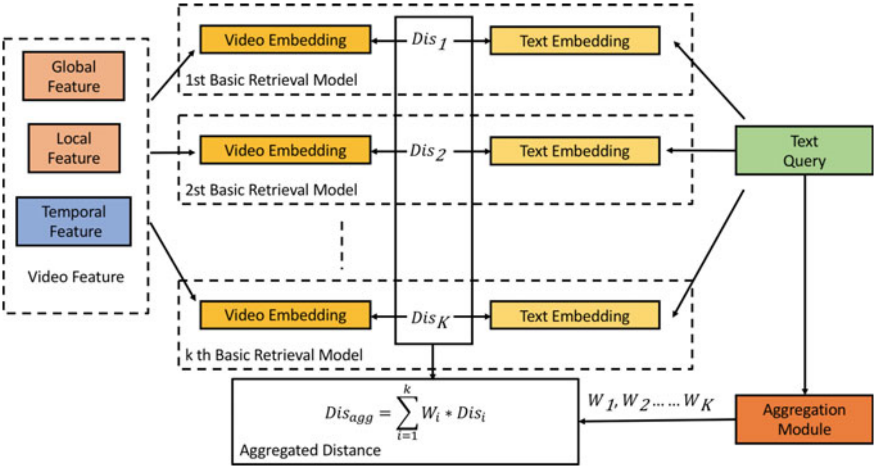

Fig. 2 The whole retrieval model contains k simple retrieval models. Video features and language query are sent into each stream to compute individual similarity distance Dis_i. The final distance is combined with k distance with aggregation module. The aggregation module exploits the semantic meaning of the query sentence to decide the importance of k basic retrieval model. Notice that our k video embedding networks share parameters of the first FC layer

3.2 Video Embedding

To localize the specific video segment from the entire video, we should take both the vision features and temporal features into consideration.

We construct our vision features using local video features V_{local} and context video features $V_{context}$. Local video features reflect what happened within a specific time span. Though the language query only depicts what occurs in the local video, context video features are important for it to provide the context information. Context information tells what happens before and after the specific time span in the video that could help localize the video segment. In our work, we first use a pretrained convolution neural network to extract features for each video frame. For a video V which consists of $[1 \ldots N]$ video frames, we construct video features as

$$V_{context} = Norm_2\left(\frac{1}{N}\sum_{i=1}^{N} Vi\right)$$

$$V_{local} = Norm_2\left(\frac{1}{N'}\sum_{i=start}^{end} Vi\right)$$

where N represents the total number of video frames, $start$ and end represent the start and end point of the local video segment; notice that $1 \leq start < end \leq N$. We use average pooling to aggregate the features in the time span. Then, L2 Normalization after pooling is applied to rescale the vision features.

Simultaneously, putting local video features and context video features into the model could weakly help the model learn temporal relation between the video segment and the entire video. To model more temporal information that indicates whether the video segment matches the language query, we add a temporal point $[T_s, T_e]$ which represents the time span into video features. The temporal features are also normalized(to [0, 1]) to be in the same numerical scale with video features. Finally, we concatenate video context features $V_{context}$, video local features V_{local}, and temporal features $[T_s, T_e]$ to construct input video representation V_{input}.

Since a video consists of several still images, we could use knowledge learned from the image dataset to learn the video information. We use the model pretrained on ImageNet [10] to extract appearance feature from the video dataset. Appearance information can represent the object and other attributes in still video frames. In video recognition, motion feature is also widely used to recognize video action in the form of optical flow [17]. To model the motion information of videos, we use a video recognition network [25] to extract motion feature. In our experiments, we construct our vision features individually with the appearance and motion feature. Two ensemble retrieval models are trained respectively with appearance and motion feature and aggregated with late fusion.

The video embedding network is constructed with two fully connected layers with ReLU. The first fully connected layer in each video embedding network is shared to reduce model parameters.

3.3 Language Embedding

The natural language input is a sequence of word embedding vector representing the text query. To capture the semantic meaning of the sentence, we use the LSTM to model the query text. We first convert each word in the text query with GloVe [15] into the word embedding vector. Although the corpus which GloVe is trained on and is not related to the DiDeMo dataset, we could use GloVe as a word embedding model for its generalization. Then, the sequence of embedding vectors is put into LSTM to aggregate the semantic meaning of the sentence. Finally, the last hidden state h_t of LSTM is linear transformed with a fully connected layer to achieve embedded text features.

3.4 Language-Based Ensemble

The core problem for cross-modal retrieval is to learn a suitable similarity metric. To address this problem, we take ensemble learning into consideration. In our work, we propose a multi-stream model with a language-based ensemble. The multi-stream model contains k basic retrieval models which are shown in Fig. 1. Each basic retrieval model contains one video embedding network and one text embedding network. In our ensemble module, language query is used to aggregate the learned similarity metric in each stream. We compute the multi-stream weights with the input sentence as

$$W_i\left(s\right) = \frac{\mathrm{e}^{p_i^{\mathrm{T}} h(s)}}{\sum_{j=1}^{k} \mathrm{e}^{p_j^{\mathrm{T}} h(s)}} \quad i \subseteq [1 \ldots k] \tag{1}$$

where s represents the input text query, $h\left(\cdot\right)$ is the aggregate function to extract sentence meaning, and p_i denotes the linear transformer. We achieve the aggregated distance as

$$Dis_{agg} = \sum_{i=1}^{k} W_i * Dis_i \tag{2}$$

The distance Dis between the input text query and the video segment is computed in each retrieval stream first.

$$Dis = D_\theta (s, v, t) \tag{3}$$

where s is text query. t is the time stamp of the video segment v.

Our ensemble model is trained with triplet loss. Triplet loss aims to bring close the matched video clip–text pair and push away unmatched pairs. In traditional video-text retrieval task, a video–text pair is composed of video segments with its text query. Compared with that, we additionally take the time stamp of the video segment as a temporal feature. In our experiment, a training pair is denoted as $< s^i, v^i, t^i >$. s^i is the text description which describes the video segment v^i. t^i is the time interval of this video segment. During training time, we sample negative training pair within the same video or from another video. According to different sample ways, we define two triplet losses: inter-video loss and intra-video loss.

Intra-video loss Localizing a video segment from an entire video is a challenging task because a queried video segment may have little difference with its context video. To distinguish a queried video segment from its context, negative pair $< s^i, v^j, t^j >$ is sampled within the same video.

Different from traditional video retrieval task which only involves video features and text features, we integrate the temporal features in our model. The temporal features depict the position of the video clip throughout the entire video. With intra-loss, we also model the relationship between temporal features and vision features. We define intra-video loss as

$$Loss_{intra} = max\left(0, m - D_\theta\left(s^i, v^j, t^j\right) + D_\theta\left(s^i, v^i, t^i\right)\right) \tag{4}$$

where v^j is any other possible video segment in the same video. t^j denotes the time point of v^j. m is the margin variable for metric learning.

Inter-video loss Compared with intra-loss, inter-video loss is proposed to match the video segments with correct semantic concepts from other videos. For this purpose, we select a negative pair which has the same time span with the positive pair. The inter-video loss is defined as

$$Loss_{inter} = max\left(0, m - D_\theta\left(s^i, v^k, t^i\right) + D_\theta\left(s^i, v^i, t^i\right)\right) \tag{5}$$

where v^k is one possible video segment in another video. Negative pair has the same temporal features t^i with the anchor video segment v^i.

Total loss consists of weighted intra-video loss and inter-video loss.

$$Loss_{all} = \lambda Loss_{intra} + (1 - \lambda)\, Loss_{inter} \tag{6}$$

where λ is the parameter to adjust the importance of these two losses. In our experiment, λ is set to 0.8 for the intra-difference which is more subtle than inter-difference.

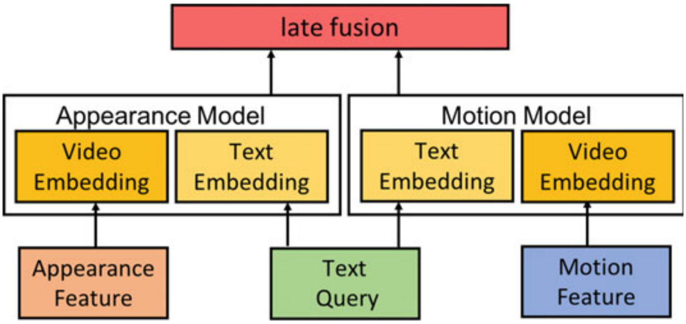

Fig. 3 The final result is obtained with aggregating the results of appearance and motion model in a late fusion way. Notice that the embedding networks in each model are trained individually

3.5 *Late Fusion*

For different visual input, we train two different multi-stream retrieval models individually: appearance model and motion model. The language-based aggregation module is only used to aggregate the distance computed in each single-stream model. To fuse the results of models trained with appearance and motion feature, we use the late fusion as shown in Fig. 3. Late fusion formula is defined as

$$Dis_{final} = (1 - \eta)\, Dis^{a}_{agg} + \eta Dis^{m}_{agg} \tag{7}$$

where Dis^{a}_{agg} and Dis^{m}_{agg} are the distance computed with appearance and motion model, η denotes the late fusion parameter. We set η to 0.5 via experiments on the validation set.

4 Experiments

In this section, we describe details of our training method and experiment results on the DiDeMo dataset.

4.1 *Experiment Setup*

We conduct experiments on the Distinct Describable Moments (DiDeMo) dataset [7]. DiDeMo consists of over 10,000 videos lasting about 25–30 s. They select about 14,000 videos from YFCC100M and eliminate those trimmed videos. The rest of the videos are then annotated by several annotators. The total number of language annotations with referring time point is over 40,000. Each description is verified to only refer to a single video moment.

The reason we choose DiDeMo as our experiment dataset is that DiDeMo contains more camera and temporal words than other video description datasets. This means the video segment in DiDeMo is depicted in multi-views. The complexity of the language description makes it more challenging to model the semantic information. It also increases retrieval difficulty that each video only has 2.57 distinct moments in average.

We report the results of our model on Rank@1 (R@1), Rank@5 (R@5), and mean intersection over union (mIoU). Each video in DeDiMo is separated into several 5 s video clips. For example, a 30 s video is broken into six five-second video segments. These video segments build up 21 possible video segments according to different time points. Our model is trained to find the most relative video segment from the 21 possible video proposals via the text query. For there are four time annotations for each video segment, four-choose-three combination is used to find the highest score.

4.2 *Implement Details*

Details of our training procedure are given below:

Data preprocessing We use GloVe [15] pretrained on the corpus from Wikipedia as our word embedding method. The dimension of the embedding vector is 300. As for visual features, the appearance feature is extracted from fc_7 using VGG [18] pretrained on ImageNet [2]. We also use a video recognition network [25] to extract motion feature. The two kinds of vision features could capture the video features in different views. To speed up the model training, all these features have been extracted before. The fine-tuning of the features extraction model is not implemented in our model. Two ensemble retrieval models are trained respectively with appearance and motion feature and aggregated with late fusion. These two models are denoted as appearance model and motion model corresponding to their video feature composition.

Training details We train the entire retrieval model which contains k basic model with TensorFlow [1]. k is set to 4 in our experiments. For each single-stream retrieval model, we set their hyperparameters to the same. The LSTM hidden dimension is 1000. Common embedding space is a 100-dimension vector space. The margin m in the ranking loss function is 0.1. To optimize the whole retrieval model, we apply stochastic gradient descent (SGD) to minimize the loss function.

It is insufficient to only use the aggregated loss computed by language-based aggregation module to optimize all k retrieval models. We also train all k retrieval models with ranking loss computed in each stream. The final loss function we use is

$$Loss_{final} = \alpha \sum_{i=1}^{k} Loss^{i}_{stream} + \beta Loss_{ens} \tag{8}$$

where $Loss^{i}_{stream}$ represents the loss in every single stream and is only backpropagated to each stream. $Loss_{ens}$ represents the loss computed with aggregated distance. α and β are two scalar parameters to balance the loss. In our experiment, α and β are set to 0.5 and 1.0.

4.3 Result

In this part, we evaluate our proposed multi-stream language aggregation retrieval model on the Distinct Describe Moments dataset and report the results on Rank@1, Rank@5, and mIoU. The results of our model and baseline model are shown in Table 1. We compare our model with the traditional method CCA and MCN [7].

We notice that CCA performs not as well as other methods. It is a traditional method to bridge the gap between different domains. The reason for its poor result is mainly for it cannot distinguish the subtle difference between video segment and its context. Appearance model in Table 1 represents our multi-steam retrieval model which only uses appearance feature as input. It outperforms CCA in Rank@1 and Rank@5 with 4.54 and 23.61%, but gets a lower result in mIoU. Compared with the appearance model, motion model achieves a better result on all the metric: Rank@1 = 27.78%, Rank@5 = 76.82%, and mIoU = 40.67%. This suggests that the motion feature is important in video tasks. Its better performance also attributes to the motion feature is extracted with video recognition network.

Our late fusion model achieves the best results: Rank@ = 1:29.39%, Rank@5 = 79.28%, and mIoU = 42.82%. Compared with MCN [7] which only uses single-stream retrieval model, our model leverages the language query information to aggregate the learned similarity metrics of multi-stream network. The late fusion model outperforms their results on all three evaluation metrics, respectively. The results show that our multi-stream retrieval network aggregated with language information learns a better similarity metric compared with single-stream network.

Table 1 Comparison of different methods of DiDeMo

Method	Rank@1	Rank@5	mIoU
CCA	18.11	52.11	37.82
MCN [7]	28.10	78.21	41.08
Appearance model	22.65	75.70	33.69
Motion model	27.78	76.82	40.67
Fusion model	**29.39**	**79.28**	**42.82**

Table 2 Comparison of different ensemble methods

Ensemble method	Rank@1	Rank@5	mIoU
Linear ensemble	27.01	76.73	39.62
Ours	27.78	76.82	40.67

Our language-based aggregation module unites each stream model in the spirit of ensemble learning. In our experiments, we train our aggregation module with a text query in an end-to-end way. To better analyze the effect of our text embedding module, we train a new motion model with another ensemble method. In this ensemble method, we obtain the final distance by directly inputting distance of each stream to a fully connected layer. Weights for each stream are trained as parameters of this FC layer. This ensemble method is denoted as a linear ensemble in our experiment. All the hyperparameters and optimization methods in this model are set to be the same with our standard motion model. Difference between these two models is only in the ensemble module. We compare the results of these two motion models with different ensemble methods in Table 2. Compared with the linear ensemble method, our ensemble method achieves better results on all three evaluation metrics. It demonstrates that it is better to use text information to the aggregate distance in each stream network.

5 Conclusion

In this chapter, we address the problem of localizing video segments via language query. Different from retrieving video from a video library, retrieving video segments should distinguish the subtle difference between corresponding video segments and other possible video segments within the same video. With a single-stream retrieval model, it is insufficient to learn a suitable similarity metric for this novel retrieval task. We propose multi-steam language aggregation retrieval model, in which semantic information is used to guide the aggregation of every single stream. With the language-based aggregation module, each single-stream network can be trained to obtain a better similarity metric. The whole retrieval model is optimized with in-stream loss and aggregated loss.

Our method outperforms other results on the DiDeMo dataset. Extensive experiments show that under our proposed aggregation module, multi-stream retrieval model can be effectively combined to accurately measure the distance between video and text domain. Future work will focus on excavating more video information and combining appearance and motion feature in a more efficient way.

References

1. Abadi M, Barham P, Chen J, Chen Z, Davis A, Dean J, Devin M, Ghemawat S, Irving G, Isard M et al (2016) Tensorflow: A system for large-scale machine learning
2. Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L (2009) ImageNet: a large-scale hierarchical image database. In: CVPR09
3. Frome A, Corrado GS, Shlens J, Bengio S, Dean J, Mikolov T et al (2013) Devise: a deep visual-semantic embedding model. In: Advances in neural information processing systems, pp 2121–2129
4. Gao J, Sun C, Yang Z, Nevatia R (2017) Tall: temporal activity localization via language query
5. Girshick R (2015) Fast r-cnn. arXiv:1504.08083
6. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778
7. Hendricks LA, Wang O, Shechtman E, Sivic J, Darrell T, Russell B (2017) Localizing moments in video with natural language. arXiv:1708.01641
8. Karpathy A, Fei-Fei L (2015) Deep visual-semantic alignments for generating image descriptions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3128–3137
9. Karpathy A, Joulin A, Fei-Fei LF (2014) Deep fragment embeddings for bidirectional image sentence mapping. In: Ghahramani Z, Welling M, Cortes C, Lawrence ND, Weinberger KQ (eds) Advances in neural information processing systems 27. Curran Associates, Inc, pp 1889–1897. http://papers.nips.cc/paper/5281-deep-fragment-embeddings-for-bidirectional-image-sentence-mapping.pdf
10. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems, pp 1097–1105
11. Le Q, Mikolov T (2014) Distributed representations of sentences and documents. In: International conference on machine learning, pp 1188–1196
12. Ma L, Lu Z, Shang L, Li H (2015) Multimodal convolutional neural networks for matching image and sentence. In: Proceedings of the IEEE international conference on computer vision, pp 2623–2631
13. Mikolov T, Chen K, Corrado G, Dean J (2013) Efficient estimation of word representations in vector space. arXiv:1301.3781
14. Nam H, Ha JW, Kim J (2016) Dual attention networks for multimodal reasoning and matching. arXiv:1611.00471
15. Pennington J, Socher R, Manning C (2014) Glove: global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp 1532–1543
16. Rohrbach A, Torabi A, Rohrbach M, Tandon N, Pal C, Larochelle H, Courville A, Schiele B (2017) Movie description. Int J Comput Vis
17. Simonyan K, Zisserman A (2014) Two-stream convolutional networks for action recognition in videos. In: Advances in neural information processing systems, pp 568–576
18. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556
19. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A et al (2015) Going deeper with convolutions. In: CVPR
20. Torabi A, Tandon N, Sigal L (2016) Learning language-visual embedding for movie understanding with natural-language. arXiv:1609.08124
21. Tran D, Bourdev L, Fergus R, Torresani L, Paluri M (2015) Learning spatiotemporal features with 3d convolutional networks. In: 2015 IEEE international conference on computer vision (ICCV). IEEE, pp 4489–4497
22. Vendrov I, Kiros R, Fidler S, Urtasun R (2015) Order-embeddings of images and language. arXiv:1511.06361
23. Venugopalan S, Rohrbach M, Donahue J, Mooney R, Darrell T, Saenko K (2015) Sequence to sequence - video to text. In: The IEEE international conference on computer vision (ICCV)

24. Vinyals O, Toshev A, Bengio S, Erhan D (2015) Show and tell: a neural image caption generator. In: The IEEE conference on computer vision and pattern recognition (CVPR)
25. Wang L, Xiong Y, Wang Z, Qiao Y, Lin D, Tang X, Van Gool L (2016) Temporal segment networks: towards good practices for deep action recognition. In: European conference on computer vision. Springer, pp 20–36
26. Wang L, Li Y, Lazebnik S (2016) Learning deep structure-preserving image-text embeddings. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 5005–5013
27. Yan F, Mikolajczyk K (2015) Deep correlation for matching images and text. In: 2015 IEEE conference on computer vision and pattern recognition (CVPR). IEEE, pp 3441–3450
28. Yu Y, Ko H, Choi J, Kim G (2016) End-to-end concept word detection for video captioning, retrieval, and question answering. arXiv:1610.02947