An Efficient Multimodal Aggregation Network for Video-Text Retrieval*

Zhi LIU††, Member, Fangyuan ZHAO†, and Mengmeng ZHANG†,††, Nonmembers

SUMMARY In video-text retrieval task, mainstream framework consists of three parts: video encoder, text encoder and similarity calculation. MMT (Multi-modal Transformer) achieves remarkable performance for this task, however, it faces the problem of insufficient training dataset. In this paper, an efficient multimodal aggregation network for video-text retrieval is proposed. Different from the prior work using MMT to fuse video features, the NetVLAD is introduced in the proposed network. It has fewer parameters and is feasible for training with small datasets. In addition, since the function of CLIP (Contrastive Language-Image Pre-training) can be considered as learning language models from visual supervision, it is introduced as text encoder in the proposed network to avoid overfitting. Meanwhile, in order to make full use of the pre-training model, a two-step training scheme is designed. Experiments show that the proposed model achieves competitive results compared with the latest work.

key words: video retrieval, CLIP, cross-modal, NetVLAD, multi-modal

1. Introduction

Retrieving video efficiently is becoming a key demand due to the rapid growth of video amounts. The mainstream approach [2]–[5] is to design text encoder at the query side and video encoder at the data side to obtain the representation of text and video, and return the results by comparing the similarity of the two representation. Among them, the design of video encoder is a critical challenge. For video contains not only spatial information, such as object and face, but also temporal information, such as action and audio. These kinds of information should be considered comprehensively to extract video representation. Prior work [2]–[5] considered using a variety of task-specific models to extract different expert from video, and the models trained on different specific task datasets can be integrated and updated without modifying the aggregation network. By introducing Transformer [7] structure to aggregate extracted features and BERT [14] to encode texts, MMT [4] achieved remarkable performance. However, it is found that the model is tend to get overfitting, since one single labeled video-text dataset is usually not sufficient for training the Transformer. To overcome this problem, one way is to increase the volume of training data. MDMMT [15] employed multiple video-text data sets to train MMT, and tested the performance with single data set, and obtained excellent performance. The other way is to reduce the parameters need to train.

In this study, to reduce the training parameters while improving the retrieval performance, an efficient multimodal aggregation network based on CLIP [1] (Contrastive Language-Image Pre-training) and NetVLAD [6] is proposed. CLIP is an outstanding model in learning visual representation from text supervision. If we regard an image as a video with one single frame, the pre training stage can be viewed as learning text representations from video supervision. By introducing CLIP, the training parameters can be reduced by pre training the text encoder on large datasets in other tasks, and fine-tune it in video retrieving task. Meanwhile, in order to improve the performance of video encoder for small training dataset, the NetVLAD, which had been proved to be efficient in video classification tasks, is introduced to substitute the Transformer structure in MMT to aggregate video features. In addition, to take full advantage of pre-trained CLIP text encoder, a two-step training scheme is designed in this study, which can fine tune the text encoder to adapt to video-text retrieval task.

2. Methodology

2.1 Mathematic Modeling

As shown in Fig. 1, suppose X representing all videos and Y representing all corresponding texts. Given a batch of B pairs data \{(vi, ci), \ldots, (vB, cB)\}, where vi ∈ X, ci ∈ Y, the problem of video retrieval task is to find a video encoder Fv and a text encoder Fc, and by comparison the similarity score between Fc and Fv, to find the matched video and text. The similarity can be expressed as:

\[ s = d(F_v(v_i), F_c(c_j)) \]  

(1)

where d(·, ·) can be any distance function, and in this paper, the cosine similarity is adopted. If i = j, the video and text are matched and the similarity score is high, otherwise the similarity score is low.

2.2 Video Encoder

In order to learn effective representations from video, we begin with video feature extractors called “expert” [3]. For
a video v, a series of pre-trained experts \( \{I^n\}_{n=1}^N \) are obtained. Each expert is a model trained for a particular task that is used to extract features from video. Each expert extracts \( M_n \) features \( \{I^n_k\}_{k=1}^{M_n} \). To improve the performance of the video encoder for small training dataset, NetVLAD model is adopted for aggregation. \( N \) NetVLADs are used to aggregate all embedding of \( N \) experts along temporal dimension and to get a unique embedding for each expert \( \{I_{agg}^1, I_{agg}^2, \ldots, I_{agg}^N\} \):

\[
I_{agg}^n = \text{NetVLAD}^n(I^n_1, I^n_2, \ldots, I^n_{M_n})
\]

where the dimension of features of each expert is \( d_n \). These aggregated experts are concatenated into a vector \( I_c \) and the dimension of \( I_c \) is:

\[
d = N \sum_{n=1}^{N} d_n
\]

We take two fully connected networks to obtain the final representation of videos. The input of the connected network is \( I_c \) with dimension \( d \). The dimension of output is consistent with the dimension of the output of the text encoder. The output of the fully connected network is passed to a Gated embedding module [2]. The output \( Z \) of the last layer is regarded as the video representation:

\[
Z = f(FC(I_c))
\]

where \( FC \) is two-layers fully connected network, and \( f \) is Gated embedding module.

2.3 Text Encoder

The CLIP is applied as the text encoder for the proposed model. It is a Transformer with 12-layer 512-wide model and 8 attention heads, and is used to generate representation for captions. For a caption \( c_i \in Y \), we denote its representation as \( T_i \):

\[
T_i = F_v(c_i)
\]

2.4 Similarity Estimation and Loss Function

Given a batch of \( B \) pairs data \( \{(v_1, c_1), \ldots, (v_i, c_i), \ldots, (v_B, c_B)\} \), where \( v_i \in X \), \( c_i \in Y \), the representation \( \{(Z_1, T_1), \ldots, (Z_i, T_i), \ldots, (Z_B, T_B)\} \) can be obtained using video encoder \( F_v \) and text encoder \( F_c \). A similarity matrix with dimension \( B \times B \) can be expressed as:

\[
s_{i,j} = d(Z_i, T_j)
\]

where \( \tau \) is a temperature hyper-parameter.

The retrieval model is optimized using InfoNCE loss:

\[
L_{c2v} = -\frac{1}{B} \sum_{i=1}^{B} \log \frac{\exp(s_{i,i}/\tau)}{\sum_{j=1}^{B} \exp(s_{i,j}/\tau)}
\]

\[
L_{v2t} = -\frac{1}{B} \sum_{i=1}^{B} \log \frac{\exp(s_{j,i}/\tau)}{\sum_{i=1}^{B} \exp(s_{j,j}/\tau)}
\]

\[
L = L_{c2v} + L_{v2t}
\]

2.5 Training

In the proposed network, the video encoder is randomly initialized, while the text encoder has been pre-trained. There is a problem if we train the randomly initialized video encoder and the pre-trained text encoder simultaneously. During training, the randomly initialized parameters of the video encoder will greatly affect the pre-trained parameters of the text encoder, and will impair the performance of the CLIP.

To avoid this problem, training process is divided into two stages, as shown in Fig. 1. In the first stage, the parameters of the text encoder are fixed, and only the parameters of video encoder are updated. In this way, after training, the semantic embeddings of the video encoder are similar to that of the text encoder. However, these semantic embeddings are not well fitted for the text video pairs due to the difference between video and image. Therefore, in the second stage, all parameters of the network are fine-tuned. In this way, based on the semantic embeddings trained from the text-image pairs, we can obtain the semantic embeddings.
Table 1. Retrieval performance on the MSRVTT. We report results in 1K-A split and 1K-B split. Compared with previous work, our methods achieve competitive results. The latest work Hit is better than ours in some indicators, but on the whole, our results is better.

<table>
<thead>
<tr>
<th>Method</th>
<th>T2V R@1</th>
<th>T2V R@5</th>
<th>T2V R@10</th>
<th>V2T R@1</th>
<th>V2T R@5</th>
<th>V2T R@10</th>
<th>MdR</th>
<th>MnR</th>
<th>MdR</th>
<th>MnR</th>
</tr>
</thead>
<tbody>
<tr>
<td>JSFusion[12]</td>
<td>1K-A</td>
<td>10.2</td>
<td>31.2</td>
<td>43.2</td>
<td>13</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>CE[5]</td>
<td>1K-A</td>
<td>20.9</td>
<td>48.8</td>
<td>62.4</td>
<td>6</td>
<td>28.2</td>
<td>20.6</td>
<td>50.3</td>
<td>64</td>
<td>5.3</td>
</tr>
<tr>
<td>MMT[4]</td>
<td>1K-A</td>
<td>24.6</td>
<td>54.0</td>
<td>67.1</td>
<td>4.0</td>
<td>26.7</td>
<td>24.4</td>
<td>56</td>
<td>67.8</td>
<td>4.0</td>
</tr>
<tr>
<td>Hit[5]</td>
<td>1K-A</td>
<td>27.7</td>
<td>59.2</td>
<td>72.0</td>
<td>2.9</td>
<td>-</td>
<td>28.8</td>
<td>60.3</td>
<td>72.3</td>
<td>3.0</td>
</tr>
<tr>
<td>ours</td>
<td>1K-A</td>
<td>29.4</td>
<td>59.7</td>
<td>72.6</td>
<td>3.0</td>
<td>19.4</td>
<td>29.3</td>
<td>59</td>
<td>72.4</td>
<td>4.0</td>
</tr>
<tr>
<td>MEE[2]</td>
<td>1K-B</td>
<td>13.6</td>
<td>37.9</td>
<td>51.0</td>
<td>10.0</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>CE[5]</td>
<td>1K-B</td>
<td>18.2</td>
<td>46.0</td>
<td>69.7</td>
<td>7.0</td>
<td>35.3</td>
<td>18.0</td>
<td>46.0</td>
<td>60.3</td>
<td>6.5</td>
</tr>
<tr>
<td>MMT[4]</td>
<td>1K -B</td>
<td>20.3</td>
<td>49.1</td>
<td>63.9</td>
<td>6.0</td>
<td>29.5</td>
<td>21.1</td>
<td>49.4</td>
<td>63.2</td>
<td>6.0</td>
</tr>
<tr>
<td>ours</td>
<td>1K-B</td>
<td>25.8</td>
<td>58.3</td>
<td>72.3</td>
<td>4</td>
<td>21.6</td>
<td>26.5</td>
<td>58.6</td>
<td>71.8</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 2. Retrieval performance on ActivityNet dataset. Our results are competitive.

<table>
<thead>
<tr>
<th>Method</th>
<th>T2V R@1</th>
<th>T2V R@5</th>
<th>T2V R@10</th>
<th>V2T R@1</th>
<th>V2T R@5</th>
<th>V2T R@10</th>
<th>MdR</th>
<th>MnR</th>
<th>MdR</th>
<th>MnR</th>
</tr>
</thead>
<tbody>
<tr>
<td>FSE[13]</td>
<td>18.2</td>
<td>44.8</td>
<td>89.1</td>
<td>7.0</td>
<td>-</td>
<td>-</td>
<td>16.7</td>
<td>43.1</td>
<td>88.4</td>
<td>7.4</td>
</tr>
<tr>
<td>CE[3]</td>
<td>18.2</td>
<td>47.7</td>
<td>91.4</td>
<td>6.0</td>
<td>23.1</td>
<td>-</td>
<td>17.7</td>
<td>46.6</td>
<td>90.9</td>
<td>6.0</td>
</tr>
<tr>
<td>HSE[13]</td>
<td>20.5</td>
<td>49.3</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>18.7</td>
<td>48.1</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>MMT[4]</td>
<td>22.7</td>
<td>54.2</td>
<td>93.2</td>
<td>5.0</td>
<td>20.8</td>
<td>-</td>
<td>22.9</td>
<td>54.8</td>
<td>93.1</td>
<td>4.3</td>
</tr>
<tr>
<td>Hit[5]</td>
<td>27.7</td>
<td>58.6</td>
<td>94.7</td>
<td>4.0</td>
<td>-</td>
<td>-</td>
<td>26.5</td>
<td>59.4</td>
<td>95.7</td>
<td>4</td>
</tr>
<tr>
<td>ours</td>
<td>25</td>
<td>57.5</td>
<td>95.8</td>
<td>4.0</td>
<td>16.2</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

3.3 Results

Table 1 and Table 2 present the results of the proposed clipVLAD network on MSR-VTT and ActivityNet Captions. We also compare clipVLAD with other state-of-the-art methods which is based on feature experts. As can be found from the tables, on both datasets, the proposed method outperform MMT. The proposed work is also competitive compared to state-of-the-art work Hit, while has less network parameters than the latter.

3.4 Ablation Studies

NetVLAD vs Transformer. In this experiment, we evaluate two video encoder models, NetVLAD and Transformer, as shown in Table 3. The text encoder is initialized using CLIP. We use Multi-modal Transformer in MMT without any modifications and take the output of the highest layer corresponding to [CLS] token as the representation of the video. The effects of different video encoder and training methods is evaluated. From the table, we can find that the two-step training scheme can greatly improve the performance compared with the direct fine-tuning method, no matter what video encoder is used, which indicates that the scheme can make better use of the pre-trained CLIP. In addition, for the same training method, the performance of using NetVLAD to encode video features is better than that of using Transformer. The reason is that there are a large number of parameters in Transformer, which requires big volume of data to train. However, most of the datasets cannot meet this requirement. In contrast, the NetVLAD can be well fitted for small datasets.

3.4 Hyper-parameter \( \tau \). According to the paper [10], [11], the value of \( \tau \) in the loss function can affects the alignment and uniformity of the feature representation in the embedding.
Table 3 Comparison of different modal of video encoder. Fine-tune means that the model is fine-tuned incautiously, while two-step represents that the model is trained using the proposed two step training method.

<table>
<thead>
<tr>
<th>Video encoder</th>
<th>training</th>
<th>R@1</th>
<th>R@5</th>
<th>R@10</th>
<th>MnR</th>
<th>MnR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multi-modal transformer</td>
<td>fine-tune</td>
<td>24.8</td>
<td>56.7</td>
<td>70.4</td>
<td>4</td>
<td>22.852</td>
</tr>
<tr>
<td>Clip2VLAD</td>
<td>two-step</td>
<td>23.3</td>
<td>58.5</td>
<td>71.6</td>
<td>4</td>
<td>20.357</td>
</tr>
</tbody>
</table>

Fig. 2 Effect of the hyper-parameter $\tau$. $\tau_1$ represent the value of $\tau$ in the first stage, while $\tau_2$ represent the value of the second stage. We report the results at R@1 and R@5.

Fig. 3 Comparison of different experts.

Comparison of the different experts. In this experiment, the number of pre-trained experts $N$ is discussed. We study the impact of different numbers of features on retrieval performance. As shown in Fig. 3, by adding experts in turn we report results at R@5 and MnR on MSR-VTT. It is found that all experts except speech helped to reduced MnR. In fact, although speech will increase MnR, it improved R@K ($K = 1, 5, 10$). Therefore, it is believed that speech also can help to improve performance, and in the remaining experiments, the number of experts is set to 7.

4. Conclusion

In this paper, to improve the video retrieving efficiency, a model named clipVLAD is proposed. The NetVLAD model is applied to aggregate video features and CLIP model is applied to extract query text information. Different from the previous work that training video encoder and text encoder simultaneously, the video encoder and text encoder are trained using two-stage scheme. Experiment results show that the proposed methods outperform the reference methods for video retrieval.

References


