Combo-Attention Network for Baidu Video Advertising

Tan Yu[1], Yi Yang[2], Yi Li[2], Xiaodong Chen[2], Mingming Sun[1], Ping Li[1]
1. Cognitive Computing Lab, Baidu Research
2. Baidu Search Ads (Phoenix Nest), Baidu Inc.
10900 NE 8th St. Bellevue WA, 98004, USA
No. 10 Xibeiwang East Road, Beijing, 100085, China
{tanyu01,yangyi15,liyi01,chenxiaodong,sunmingming01,liping11}@baidu.com

ABSTRACT

With the progress of communication technology and the popularity of the smart phone, videos grow to be the largest medium. Since videos can grab a customer's attention quickly and leave a big impression, video ads can gain more trust than traditional ads. Thus advertisers start to pour more resources into making creative video ads to built the connections with potential customers. Baidu, as the leading Chinese search engine firm, receives billions of search queries per day. In this paper, we introduce a technique used in Baidu video advertising for feeding relevant video ads according to the user's query. Note that, retrieving relevant videos using the text query is a cross-modal problem. Due to the modal gap, the text-to-video search is more challenging than well exploited text-to-text search and image-to-image search. To tackle this challenge, we propose a Combo-Attention Network (CAN) and launch it in Baidu video advertising. In the proposed CAN model, we represent a video as a set of bounding boxes features and represent a sentence as a set of words features, and formulate the sentence-to-video search as a set-to-set matching problem. The proposed CAN is built upon the proposed combo-attention module, which exploits cross-modal attentions besides self attentions to effectively capture the relevance between words and bounding boxes. To testify the effectiveness of the proposed CAN offline, we built a Daily700K dataset collected from HaoKan APP. The systematic experiments on Daily700K as well as a public dataset, VATEX, demonstrate the effectiveness of our CAN. After launching the proposed CAN in Baidu's dynamic video advertising (DVA), we achieve a 5.47% increase in Conversion Rate (CVR) and a 11.69% increase in advertisement impression rate.

KEYWORDS

search, ranking, cross-modal, short video, neural networks

ACM Reference Format:
Tan Yu[1], Yi Yang[2], Yi Li[2], Xiaodong Chen[2], Mingming Sun[1], Ping Li[1]. 2020. Combo-Attention Network for Baidu Video Advertising. In *Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '20), August 23–27, 2020, Virtual Event, CA, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3394486.3403297

KDD '20, August 23–27, 2020, Virtual Event, CA, USA
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-7998-4/20/08…$15.00
https://doi.org/10.1145/3394486.3403297

1 INTRODUCTION

With the popularity of the smart phones, people can easily record and edit videos. Meanwhile, with the advancement of (wireless) communication technologies, users can receive and send short videos in seconds, which fosters the explosive growth of short-video APPs such as Snapchat, Vine, Tik Tok, and Kuaishou in United States and China. Short-videos are seconds-long or minutes-long videos, which are short but interesting and creative. Watching short videos in leisure time has become a fashionable way for relaxation. In 2019, DAU (daily active users) of several short-video APPs such as Tik Tok and Kuaishou has surpassed 200 million, and everyday, billions of new short videos are created and shared. The emergence of the short video market motivates the advertisers to put more efforts on making creative video ads for attracting attentions of potential customers. Baidu, as the leading search engine company in China, receives billions of search queries from the users. Feeding potential interested video ads according to a user's query is the core task of Baidu video advertising.

In fact, the text-to-video search is a cross-modal search, which is different from traditional text-to-text search used in current industry search engines. It also differs from well exploited content-based image/video search using image query. In the past a few years, video understanding [5, 26, 28, 31] have achieved break-through performance thanks to the convolutional neural network (CNN). Meanwhile, in recent years, text understanding also achieves significant progress thanks to the attention mechanism [7, 29]. Despite that significant improvement has been achieved in video understanding and text understanding by computer vision and natural language processing community, how to learn or design an effective text-to-video matching metric is still far from well addressed.

Traditionally, the text-to-video search is solved by the joint embedding [12, 13]. Basically, it represents a text query as a holistic feature vector and represents a reference video for retrieval in the database as a global visual feature vector. Then the visual feature vector of the video and the text query's holistic feature vector are further mapped into a joint semantic feature space. It seeks to minimize the distances between the text query and its relevant videos in the joint semantic feature space and meanwhile enlarges the distances between the text query and its irrelevant videos. In this scheme, both sentences and videos are represented by global features, which are efficient for training and retrieval. Nevertheless, the global representation is incapable of conducting local matching between a sentence and a video. For example, given a sentence $\mathcal{S} = [w_1, \cdots, w_n]$ containing n words and a video $\mathcal{V} = [f_1, \cdots, f_m]$ containing m frames. In some scenarios, only the i-th word w_i is closely related with a small area in j-th frame f_j

whereas the other words and other frames are unrelated. In this case, if we represent $\mathcal{S}$ as a global feature $\mathbf{S}$ and represent $\mathcal{V}$ as a holistic vector $\mathbf{V}$, the close relation between w_i and f_j will be distracted by the irrelevance between other words and other frames.

To overcome the drawback, in this paper, we formulate the sentence-to-short-video retrieval task as a set-to-set matching problem. To be specific, the sentence $\mathcal{S}$ is represented by a set of word-level features $\mathcal{W} = \{\mathbf{w}_i\}_{i=1}^{n}$ and the video $\mathcal{V}$ is represented by a set of bounding boxes features $\mathcal{B} = \{\mathbf{b}_j\}_{j=1}^{t}$. The bounding boxes represent some candidate locations of objects, obtained by a pre-trained object detector such as faster R-CNN [25]. The similarity between $\mathcal{S}$ and $\mathcal{V}$ is obtained through set-to-set matching between $\mathcal{W}$ and $\mathcal{B}$ based on the proposed combo-attention network (CAN):

$$\mathrm{sim}(\mathcal{S}, \mathcal{V}) = \mathrm{CAN}(\mathcal{W}, \mathcal{B}). \tag{1}$$

We visualize the overview of the proposed method in Figure 1.

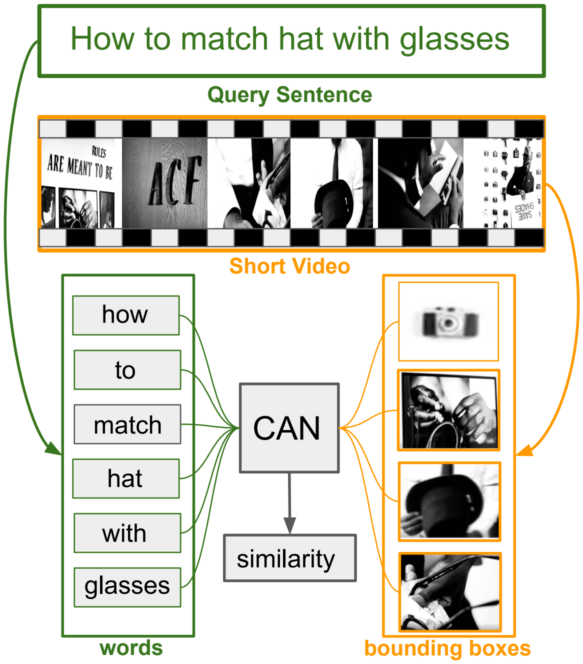


Figure 1: The overview of the proposed method. A query sentence is represented by a set of words features and a short video is represented by a set of bounding boxes features. The similarity between the query sentence and the short video is obtained by the proposed CAN, which conducts a matching between the set of bounding boxes and the set of words.

The proposed CAN is based on the proposed combo-attention module visualized in Figure 2(b). It is inspired by the recent success of transformer [29] in natural language processing. On one hand, the combo-attention module utilizes the video's bounding boxes features to generate attentions for the sentence's words features. On the other hand, it generates attentions for the bounding boxes features based on the words features. This cross-modal attention mechanism provides the context of the video for modeling the sentence and meanwhile gives the context of the sentence to model the video. In the training phase, the similarities are used to construct the training loss for updating the weights of CAN. In the testing phase, the query sentence and reference short videos are ranked based on their similarities with the query sentence.

To evaluate the performance of the proposed CAN on short video retrieval, we built a Daily700K dataset, which consists of 600, 000 short videos collected from HaoKan APP. The queries and videos are paired through logs of user clicks. Our systematic experiments conducted on Daily700K as well as a public dataset, VATEX [32], demonstrate the effectiveness of the proposed CAN. Meanwhile, CAN has been launched in Baidu dynamic video advertising (DVA). After launch, it achieves a 5.47% increase in Conversion Rate (CVR) and a 11.69% increase in advertisement impression rate.

In a nutshell, the contributions of this paper are four-fold:

- We formulate the sentence-to-short-video retrieval problem as a set-to-set matching problem.
- We propose a combo-attention network (CAN) based on the proposed combo-attention module.
- A new dataset, Daily700K, is built for evaluating the proposed CAN on the text-to-video retrieval.
- The proposed CAN has been launched in Baidu dynamic video advertising (DVA), achieving excellent performance.

2 RELATED WORK

We review the related work in three fields: sentence representation, video representation and text-to-video retrieval.

2.1 Baidu Search Ads

Baidu Search Ads (a.k.a. "Phoenix Nest") is the major revenue source for the company. In the search industry, sponsored online advertising produces many billions of dollar revenues for online ad publishers. The task of CTR (Click-Through Rate) prediction [4, 10, 11, 38] plays a key role to determine the best ad spaces allocation. CTR prediction takes input (such as query-ad relevance, ad features, user portraits, etc.) to estimate the probability that a user clicks on a given ad. Since 2013 [11], Baidu Search Ads has been using ultra-high dimensional input data and ultra-large-scale deep neural networks for training CTR models, using MPI-based architectures.

Since around 2017, Baidu Search Ads has been undergoing several major upgrades by incorporating the rapid-growing technologies in near neighbor search, machine learning, and systems. For example, [37, 38] reported new architectures for distributed GPU-based parameter servers which have replaced the MPI-based system for training CTR models. [11] described the widespread use of approximate near neighbor search (ANNS) and maximum inner product search (MIPS) [36, 39] to substantially improve the quality of ads recalls in the early stage of the pipeline of Baidu's ads system.

In recent years, Baidu's short-form video recommendations [19] and video-based search ads have achieved great progress. In this paper, we introduce the technology for a representative project which has significantly boosted Baidu's video-based ads revenues.

2.2 Sentence Representation

Traditionally, a sentence's representation is obtained through word-level embedding followed by a recurrent neural network (RNN) [2]. Word-level embedding maintains the semantic consistency in the feature space whereas the RNN models the order of words in the sequence. Nevertheless, the sequential nature of the recurrent neural network makes it memory-costly and time-consuming when processing long sequences. To improve the efficiency, ByteNet [16] proposes to replace RNN by a one-dimensional CNN, which models the order of the sequence through convolution layers. It achieves a comparable sentence classification precision but well support parallelism and is efficient for training. QRNN [3] stacks a CNN module and an RNN module, which possesses high training efficiency thanks to the CNN module and meanwhile effectively models the temporal order through the RNN module. ConvS2S [14] is also built upon a CNN and encodes the positions of words in the sentence to explicitly model the order of words. It introduces an attention module to provide the context information for a more effective representation. Recently, Transformer [29] built on a stack of self-attention blocks has significantly improved the performance in many NLP tasks. BERT [7] further improves the Transformer using a bi-directional structure and achieves a better performance.

2.3 Video Representation

To model the dynamics in the video, early works gain the video representation by feeding a sequence of frame-level frames into an RNN [8]. Nevertheless, the dynamics normally exist in local patches and cannot be effectively modeled through the global frame features fed into RNN. To overcome the challenge, two-stream CNN [26] leverages optical flow as an addition stream to model the local dynamics. However, extracting the optical flows is considerably time consuming. In parallel, 3D-CNN [28] efficiently models the local dynamics through convolution along the temporal dimension. It has achieved excellent performance based on pre-training on a large scale video dataset [5]. Since 3D-CNN only models the local dynamics within neighboring a few frames, Non-local Neural Network [30] further improves 3D-CNN by additionally adding global context through the proposed non-local block. Interestingly, the non-local block is very similar to self-attention block used in Transformer [29]. Meanwhile, Wang *et al.* [31] extract the bounding boxes features of a video and conduct the graph-convolution on the bounding boxes features. In fact, the graph convolution layer used in [31] is also similar to the self-attention block used in Transformer.

2.4 Text-to-vision Retrieval

Joint Embedding. A range of prior work [12, 13] have exploited image-text joint embedding for text-to-image retrieval. To be specific, they map the images and natural languages into the same semantic space. They keep a close distance between relevant images and sentences, and maintain a large distance between the irrelevant images and sentences. Recently, with the emergence of videos, the research community gradually pays more and more attention to video-text joint embedding. Traditional video-text embedding methods [23, 24, 33] normally rely on the frame-level features. Nevertheless, frame-level features fail to encode the global visual information, which might be important in some scenarios. To obtain a global video-level representation, [22] simply conducts average pooling over frame-level features, achieving better performance than methods [23, 24, 33] based on frame-level features. Nevertheless, average pooling cannot model the complex relations among the video frames, and average pooling also leads to significantly information loss since is straightforwardly sums up the activations of multiple frames. Note that, joint embedding optimizes the cross-modal metric in the late stage where we have already obtained the feature of a video/image and the feature of a sentence.

Early Fusion. To enhance the effectiveness of matching the video feature and the text feature, some work [6, 21, 34] fuse the video feature and the text feature in the early stage. To compute the representation of a sentence, m-CNN [21] takes input the image feature besides words' features when conducting 1D convolution. On the other side, to compute the representation of an image, BCN [6] uses the sentence's feature to modulate feature maps of the image. CT-SAN [34] generates an attention map by fusing the LSTM feature of the sentence and the image feature. They further use the attention map and the image's feature map to generate a vector, which is fed to an LSTM to generate the sentence's final feature. Recently, inspired by the great progress achieved by BERT [7], some methods [20, 27] extend the original BERT language model to a cross-modal model to tackle the language-vision tasks. VideoBERT [27] utilizes the clustering to convert to a frame's visual feature into a visual word, and thus converts a video into a visual sentence. It further concatenates the video sentence and the original language sentence as the input of the original BERT model. Nevertheless, VideoBERT suffers from distortion error from clustering, and it treats the vision and text equally, ignoring the differences between these two modals. ViLBERT [20] further improves VideoBERT by designed a two-stream architecture consists of a text stream and an image stream. In each stream, they design a co-attention transformer layer which takes both two modals as input to generate the attention. Similarly, MCN [35] also uses the text feature to guide the attention when generating the feature of the image. Nevertheless, both ViLBERT and MCN are designed for the image-text tasks, which can not be directly used for video-text tasks.

3 METHOD

Given a short video $\mathcal{V}$, we extract T frames through uniformly sampling. For each frame f_t, we detect K bounding boxes through faster R-CNN [25], which serve as the potential locations of objects in the frame. The visual feature of a bounding box is obtained by sum-pooling over the bounding box's region in a convolutional feature map. On the text side, given a sentence $\mathcal{S}$ consisting of M words, we obtain a sequence of features $[\mathbf{w}_1, \cdots, \mathbf{w}_j, \cdots, \mathbf{w}_M]$, where $\mathbf{w}_j$ represents the feature of j-th word in the sentence $\mathcal{S}$.

3.1 Basic Block

Our model is built on two basic blocks, self-attention (SA) block and combo-attention (CA) block. We visualize architectures of two blocks in Figure 2. As shown in the figure, the SA block is a standard module in BERT. As for the proposed CA block, we give more details here. Our CA block takes two types of feature $\mathbf{X}$ and $\mathbf{Y}$ as input. To be specific, $\mathbf{X}$ is a matrix of $D_x \times N_x$ size where N_x is the number of features in $\mathbf{X}$ and D_x is the feature dimension. Similarly, $\mathbf{Y}$ is a matrix of $D_y \times N_y$ size where N_y is the number of features in $\mathbf{Y}$ and D_y is the feature dimension. The proposed CA block first computes the value matrix $\mathbf{V}_1$ and the key matrix $\mathbf{K}_1$ based on $\mathbf{Y}$, and then compute the query matrix $\mathbf{Q}_1$ based on $\mathbf{X}$ by

$$\mathbf{V}_1 = f_1(\mathrm{Y}),\ \mathbf{K}_1 = g_1(\mathrm{Y}),\ \mathbf{Q}_1 = h_1(\mathrm{X}), \tag{2}$$

where $\mathbf{V}_1 \in \mathbb{R}^{N_y \times D_v}$, $\mathbf{K} \in \mathbb{R}^{N_y \times D_k}$, $\mathbf{Q} \in \mathbb{R}^{N_x \times D_q}$, $D_q = D_k$ and

$$f_1(\mathrm{Y}) = \mathbf{W}_{f_1}\mathrm{Y},\ g_1(\mathrm{Y}) = \mathbf{W}_{g_1}\mathrm{Y},\ h_1(\mathrm{X}) = \mathbf{W}_{h_1}\mathrm{X}. \tag{3}$$

We define the j-th column of the query matrix $\mathbf{Q}_1$ as the query vector $\mathbf{q}_j$. By computing the matrix-vector product between $\mathbf{q}_j$ and the key matrix $\mathbf{K}_1$ followed by a softmax operation, the soft-attention vector $\mathbf{a}_j$ is obtained by

$$\mathbf{a}_j = \mathrm{softmax}(\mathbf{K}_1\mathbf{q}_j^\top / \sqrt{D_q}). \tag{4}$$

Then the attended feature vector $\mathbf{f}_j$ is obtained by a weighted summation over all columns of $\mathbf{V}_1$ and the weights are items in $\mathbf{a}_j$:

$$\mathbf{f}_j = \mathbf{V}_1\mathbf{a}_j^\top. \tag{5}$$

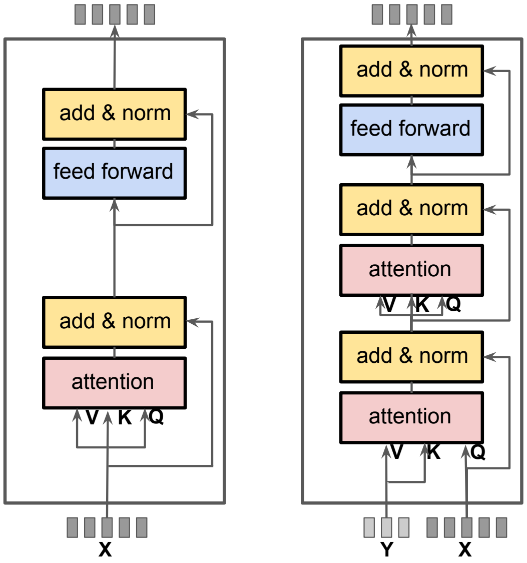
The attended feature matrix $\mathbf{F}_1$ consists of all attended features:

$$\mathbf{F}_1 = [\mathbf{f}_1, \cdots, \mathbf{f}_j, \cdots, \mathbf{f}_N]. \tag{6}$$

$\mathbf{F}_1$ goes through an add&norm layer and generates:

$$\hat{\mathbf{F}}_1 = \mathrm{norm}(\mathbf{F}_1) + \mathrm{X}, \tag{7}$$

where norm$(\cdot)$ denotes the layer normalization [1]. $\hat{\mathbf{F}}_1$ is further used to compute the $\mathbf{V}_2$, $\mathbf{K}_2$ and $\mathbf{Q}_2$, which further go through a self-attention layer and another add&norm layer, and generate $\hat{\mathbf{F}}_2$. Finally, $\hat{\mathbf{F}}_2$ goes through a feed-forward layer and another add&norm layer to generate the output of the block. Note that, for easiness of illustration, the above formulation is based on the single head. In implementation, we adopt a 8-head settings for all attention blocks.



(a) Self-Attention (b) Combo-Attention

Figure 2: The architecture of the self-attention (SA) module and the combo-attention (CA) module.

3.2 Architecture

Figure 3 visualizes the architecture of the proposed model. It can be divided into two streams: the sentence stream and the video stream. The input of the short video stream is a set of bounding boxes features $\mathcal{B} = \{\mathbf{b}_1, \cdots, \mathbf{b}_N\}$. The feature of a bounding box $\mathbf{b}$ is a sum of its visual feature $\mathbf{v}$ and its spatio-temporal location vector $\mathbf{l}$:

$$\mathbf{b} = \mathbf{W}_v\mathbf{v} + \mathbf{W}_l\mathbf{l}, \tag{8}$$

where $\mathbf{W}_v$ and $\mathbf{W}_l$ are learnable projection matrices to make the dimension of the bounding box visual feature identical to that of the spatio-temporal location vector. The visual feature $\mathbf{v}$ is obtained by sum-pooling the convolutional features within the detected region. The spatio-temporal location vector is defined as $\mathbf{l} = [\hat{x}_0, \hat{x}_1, \hat{y}_0, \hat{y}_1, \hat{t}]$, where

$$\hat{x}_0 = \frac{x_0}{W},\ \hat{x}_1 = \frac{x_1}{W},\ \hat{y}_0 = \frac{y_0}{H},\ \hat{y}_1 = \frac{y_1}{H},\ \hat{t} = \frac{t}{L}, \tag{9}$$

and x_0 is the x-axis coordinate of the upper-left corner, x_1 is the x-axis coordinate of the lower-right corner, y_0 is the y-axis coordinate of the upper-left corner, x_1 is the y-axis coordinate of the lower-right corner, and t is the frame index, W is the frame width, H is the frame height and L is the number of sampled frames from the short video. The input of the sentence stream is a set of words features $\mathcal{W} = \{\mathbf{w}_1, \cdots, \mathbf{w}_M\}$. The word feature $\mathbf{w}$ is a summation of the word-embedding feature and the positional feature. To be specific,

$$\mathbf{w} = \mathbf{w}_e + \mathbf{w}_p, \tag{10}$$

where $\mathbf{w}_e$ is extracted through a word2vector model and $\mathbf{w}_p$ is the positional embedding of the word in the same manner as Transformer [29]. Horizontally, the architecture can be partitioned into three parts: 1) the self-attention part, 2) the combo-attention part and 3) the similarity-computation part.

The self-attention part. In the video stream, the set of boxes features go through a cascade of two SA layers and generate a set of self-attended bounding boxes features $\hat{\mathbf{B}} = [\hat{\mathbf{b}}_1, \cdots, \hat{\mathbf{b}}_N]$. In parallel, in the sentence stream, the words features also go through a cascade of two SA layers and generate a set of self-attended words features $\hat{\mathbf{W}} = [\hat{\mathbf{w}}_1, \cdots, \hat{\mathbf{w}}_M]$.

The combo-attention part. In the video stream, self-attended bounding boxes features $\hat{\mathbf{B}} = [\hat{\mathbf{b}}_1, \cdots, \hat{\mathbf{b}}_N]$ are used for generating the query matrix $\mathbf{Q}$ for the first CA block of the video side. The key matrix $\mathbf{K}$ and the value matrix $\mathbf{V}$ of the input of the first CA of the video stream are obtained from the self-attended words features $\hat{\mathbf{W}} = [\hat{\mathbf{w}}_1, \cdots, \hat{\mathbf{w}}_M]$. The output of the first CA in the video stream is the cross-attended bounding boxes features $\bar{\mathbf{B}}^{(1)} = [\bar{\mathbf{b}}_1^{(1)}, \cdots, \bar{\mathbf{b}}_N^{(1)}]$. In parallel, in the sentence stream, the self-attended words features $\hat{\mathbf{W}} = [\hat{\mathbf{w}}_1, \cdots, \hat{\mathbf{w}}_M]$ are used for generating the query matrix $\mathbf{Q}$ in the input of the first CA block in the sentence stream. Meanwhile, the value matrix $\mathbf{V}$ as well as the key matrix $\mathbf{K}$ in the input of the first CA of the sentence stream is computed from the self-attended boxes features $\hat{\mathbf{B}} = \{\hat{\mathbf{b}}_1, \cdots, \hat{\mathbf{b}}_N\}$. The output of the first CA in the sentence stream is the cross-attended words features $\bar{\mathbf{W}}^{(1)} = [\bar{\mathbf{w}}_1^{(1)}, \cdots, \bar{\mathbf{w}}_M^{(1)}]$. In a similar manner, the second CA of the video stream generates

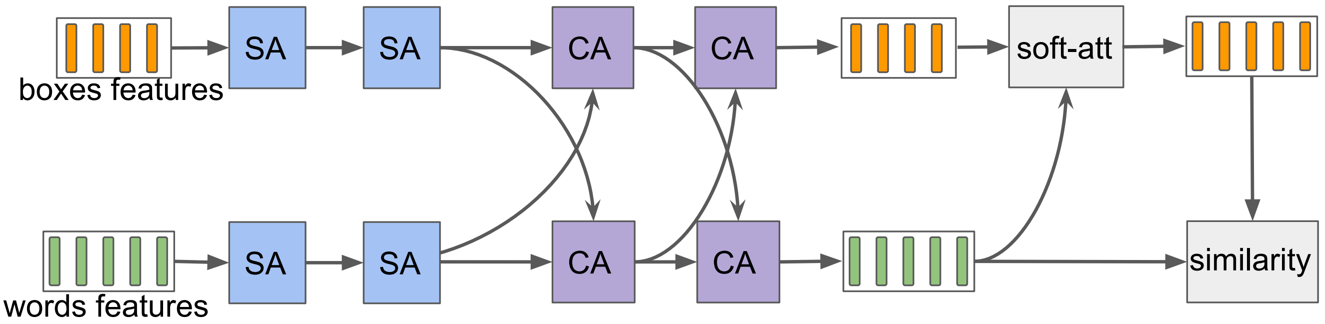


Figure 3: The architecture of the proposed model. Vertically, it can be divided into two streams. The upper stream processes the short video and the lower stream accounts for the sentence. Horizontally, it can be partitioned into three parts. The left part conducts the self attention on each stream individually, the middle part incorporates the cross-modal attention between two streams and the right part measures the similarity between the sentence and the short video.

$\bar{\mathbf{B}}^{(2)} = [\bar{\mathbf{b}}_1^{(2)}, \cdots, \bar{\mathbf{b}}_N^{(2)}]$ and the second CA of the sentence stream generates $\bar{\mathbf{W}}^{(2)} = [\bar{\mathbf{w}}_1^{(2)}, \cdots, \bar{\mathbf{w}}_N^{(2)}]$.

The similarity-computation part. Soft attention layer (soft-att) takes the cross-attended boxes features $\bar{\mathbf{B}}^{(2)} = [\bar{\mathbf{b}}_1^{(2)}, \cdots, \bar{\mathbf{b}}_N^{(2)}]$ as well as cross-attended words features $\bar{\mathbf{W}}^{(2)} = [\bar{\mathbf{w}}_1^{(2)}, \cdots, \bar{\mathbf{w}}_M^{(2)}]$ as input, and computes a similarity matrix S by

$$\mathbf{S} = (\bar{\mathbf{B}}^{(2)})^\top \bar{\mathbf{W}}^{(2)}. \tag{11}$$

For each column of S, $\mathbf{s}_i$, we conduct a soft-max operation on it and obtained a new vector:

$$\tilde{\mathbf{s}}_i = \text{softmax}(\mathbf{s}_i). \tag{12}$$

Then a new similarity matrix is obtained through $\tilde{\mathbf{S}} = [\hat{\mathbf{s}}_1, \cdots, \hat{\mathbf{s}}_M]$. The output of soft attention layer is computed by

$$\tilde{\mathbf{W}} = \bar{\mathbf{B}}^{(2)} \tilde{\mathbf{S}}. \tag{13}$$

The final similarity score is computed by

$$\mathbf{s} = \sum_{i=1}^{M} \langle \tilde{\mathbf{w}}_i, \bar{\mathbf{w}}_i^{(2)} \rangle, \tag{14}$$

where $\tilde{\mathbf{w}}_i$ denotes the i-th column of $\tilde{\mathbf{W}}$.

We summarize the pipeline of computing the relevance score between a video $\mathcal{V}$ and a sentence $\mathcal{S}$ in Algorithm 1. In the training phase, the similarities are further used for computing the loss. In the testing phase, the relevance scores are used for ranking.

Training loss. Let us define $s(i, j)$ as the similarity score of the video V_i and the sentence S_j. We seek to maximize the similarities between relevant sentence-video pairs and minimize the similarities between irrelevant sentence-video pairs. We construct each mini-batch by K ground-truth video-sentence pairs $\{(V_k, S_k)\}_{k=1}^K$. Meanwhile we set that each video V_k in the mini-batch is only relevant with the sentence in its ground-truth sentence-video pair, S_k, and irrelevant with other sentences. We define the loss $\mathcal{L}$ as

$$\mathcal{L} = \sum_{k=1}^{K} \Big[\sum_{j \neq k} [\alpha - s(k,k) + s(k,j)]_+ + \sum_{j \neq k} [\alpha - s(k,k) + s(j,k)]_+ \Big], \tag{15}$$

Algorithm 1 The pipeline of the proposed CAN.

```
Input: The set of word features $\mathbf{W} = [\mathbf{w}_1, \cdots, \mathbf{w}_M]$ extracted by Eq. (10) from a sentence $\mathcal{S}$. The set of bounding box features $\mathbf{B} = [\mathbf{b}_1, \cdots, \mathbf{b}_N]$ extracted by Eq. (8) from a video $\mathcal{V}$.
Output: $s(\mathbf{V},\mathbf{S})$, the relevance score of the pair $(\mathbf{V},\mathbf{S})$.
$\hat{\mathbf{W}}^{(0)} \leftarrow \mathcal{W}$, $\hat{\mathcal{B}}^{(0)} \leftarrow \mathcal{B}$
for $i \in [1, 2]$ do
    $\hat{\mathbf{W}}^{(i)} \leftarrow \text{SA}_{text}(\hat{\mathcal{W}}^{(i-1)})$
    $\hat{\mathbf{B}}^{(i)} \leftarrow \text{SA}_{video}(\hat{\mathbf{B}}^{(i-1)})$
end for
$\bar{\mathbf{W}}^{(0)} \leftarrow \hat{\mathbf{W}}^{(2)}$, $\bar{\mathbf{B}}^{(0)} \leftarrow \hat{\mathbf{B}}^{(2)}$
for $i \in [1, 2]$ do
    $\bar{\mathbf{W}}^{(i)} \leftarrow \text{CA}_{text}(\bar{\mathbf{W}}^{(i-1)}, \bar{\mathbf{B}}^{(i-1)})$
    $\bar{\mathbf{B}}^{(i)} \leftarrow \text{CA}_{video}(\bar{\mathbf{B}}^{(i-1)}, \bar{\mathbf{W}}^{(i-1)})$
end for
$\tilde{\mathbf{W}} \leftarrow \text{SoftAtt}(\bar{\mathbf{W}}^{(2)}, \bar{\mathbf{B}}^{(2)})$ as Eq. (11)-(13)
$\text{s}(\mathbf{V},\mathbf{S}) \leftarrow \sum_{i=1}^{M} \langle \tilde{\mathbf{w}}_i, \bar{\mathbf{w}}_i^{(2)} \rangle$
return $\text{s}(\mathbf{V},\mathbf{S})$
```

where $[x]_+ = max(x, 0)$, and α is the margin which is a predefined constant. By default, we set $\alpha = 0.2$ in all experiments. Note that, the item $\sum_{j \neq k} [\alpha - s(k,k) + s(k,j)]_+$ in Eq. (15) targets to make the similarity between V_k and S_k larger by a margin α than similarities between V_k and other sentences in the mini-batch. In contrast, the item $\sum_{j \neq k} [\alpha - s(k,k) + s(j,k)]_+$ seeks to make the similarity between V_k and S_k larger by a margin α than similarities between S_k and other videos in the mini-batch. Note that, the loss function used in Eq. (15) takes all the negative triplets beyond the margin α into consideration, which is different from the hard negative mining used in VSE++ [9]. Our experiments show that, by replacing the loss function in Eq. (15) with the hard negative mining loss in VSE++ [9], the performance drops. This might be due to that the hard negative mining only counts from the hardest triplet, which is prone to modal collapse.

(a) how to use a washer

(b) draw a puppy sketch

(c) plant succulents in pots

(d) women handbag recommend

Figure 4: Visualization of some pairs of short videos and query sentences from our Daily700K dataset. Note that, in our Daily700K dataset, the texts are in Chinese. For the convenience of illustration, we translate the Chinese to English.

4 DEPLOYMENT

In this section, we introduce how the proposed CAN is deployed in the video retrieval system. Due to that our video database is large-scale, given a query text, limited by efficiency, it is unfeasible to get the relevance score of the text query with every video in the database through the proposed CAN. Therefore, we only deploy the CAN in the re-ranking stage. As shown in Figure 5, given a text query, we first conduct the title-based search. Benefited from the indexing of the sentence features, this step can be conducted very efficiently. Based on the ranking result of title-based retrieval, we select top M most relevant videos. The CAN is used to re-rank the selected M videos.

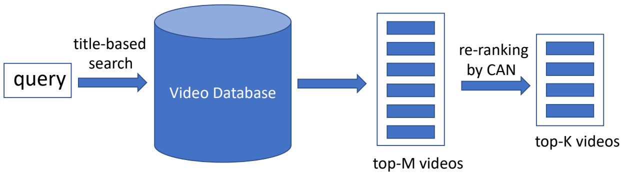


Figure 5: The deployment of CAN in the pipeline.

5 EXPERIMENTS

5.1 Datasets and Implementation Details

Since short-video APPs just emerge in recent years, we have not found a publicly released short-video retrieval dataset satisfying our demand. Therefore, we build a new dataset, Daily700K. It consists of 70, 000 pairs of short videos and query sentences, which are mainly about daily lives. We use 695, 000 pairs for training data and the rest 5, 000 pairs for testing. Since the relevance between a query and the short videos is relatively subjective to users, we collect ground-truth pairs by selecting the query-video pairs with high click rates, representing good ones for a large number of users. In Figure 4, we visualize some pairs of short videos and query sentences. In addition to evaluating the proposed CAN on our built Daily700K dataset, we also conduct experiments on a public benchmark dataset, VATEX [32]. It has 25, 991 training videos paired with 519, 820 captions and 3, 000 validation videos paired with 60, 000 captions. The dataset provided both Chinese captions and English Captions. We use the Chinese captions for experiments.

The bounding boxes are generated from Faster R-CNN [25] built on ResNet-101 [15] pre-trained on Visual Genomes [17]. For each detected region of the interest (ROI), *i.e.*, the bounding box, its feature is obtained by sum-pooling the convolutional features within the bounding box. The feature dimension is 2048. We use multi-head attention in all self-attention modules and combo-attention modules, and we set the number of head as 8. We evaluate the performance of algorithms based on two metrics, sentence-to-video (s2v) average recall@$\{1, 5, 10\}$ and video-to-sentence (v2s) average recall@$\{1, 5, 10\}$. We train the proposed CAN with ADAM optimizer. The initial learning rate 1×10^{-4} and decreases it to 1×10^{-5} after 30 epochs. The whole training process finishes in 50 epochs. All models are trained and deployed based on the PaddlePaddle deep learning framework developed by Baidu.

5.2 Ablation Study

Global versus local. We compare our method with the method based on global feature. To be specific, we compare ours with three baselines: 1) the global video feature with local words features, 2) the global sentence feature with local bounding boxes features and 3) the global sentence feature with the global video feature. The global video feature is obtained by sum-pooling bounding boxes features. The global sentence feature is obtained by sum-pooling words features. We use 32 local bounding boxes features per video and 10 local words features per sentence.

Table 1 shows the retrieval results of ours based on local features and methods based on global features. As shown in the table, using the global video feature and the global sentence features, it only achieves a 12.8 recall@1 for sentence-to-video search and a 13.9 recall@1 for video-to-sentence search, which is considerably worse

Table 1: Comparisons between global and local features.

		s2v recall			v2s recall		
sentence	video	@1	@5	@10	@1	@5	@10
global	global	12.8	57.7	62.7	13.9	58.3	61.6
global	local	13.9	60.5	64.0	16.1	58.8	70.5
local	global	32.6	73.6	85.4	28.9	76.6	90.2
local	local	51.6	81.9	89.4	52.4	82.3	90.5

than ours based on local bounding boxes features and local words features. Meanwhile, the recall@1 of sentence-to-video search when using the global video feature and local words features is only 32.6 and the recall@1 of sentence-to-video search when using local bounding boxes feature and the global sentence feature is only 13.9. Both of them are lower than our 51.6 recall@1 of sentence-to-video search based on local bounding boxes features and text words features. As for video-to-sentence search, using the global video feature and local words features only achieves a 28.9 recall@1, and using local bounding boxes feature and the global sentence feature only achieves a 16.1 recall@1. Both of them are also worse than ours. The inferior performance achieved by methods based on the global sentence or video feature validate the effectiveness of using local bounding boxes features and local words features.

Table 2: The influence of bounding boxes # per video. The experiments are conducted on Daily700K dataset.

	s2v recall			v2s recall		
# box	@1	@5	@10	@1	@5	@10
4	44.4	80.6	89.0	45.5	81.7	90.2
8	47.5	81.6	89.2	48.3	82.2	90.3
16	49.1	81.7	89.4	50.1	82.3	90.3
32	51.6	81.9	89.4	52.4	82.3	90.5

Impact of the number of bounding boxes. We evaluate the impact of the number of bounding boxes on the performance of the proposed CAN. We set the number of bounding boxes per frame as 10 and sample 20 frames per video, and thus the total bounding boxes per video is 200. To improve the efficiency of training, we further conduct k-medians clustering on 200 bounding boxes to select a more compact set of bounding boxes. We vary the number of selected bounding boxes among $\{4, 8, 16, 32\}$. We testify the influence of the number of selected bounding boxes on the retrieval recall. The experiments are conducted on Daily700K dataset. As shown in Table 2, the retrieval recall consistently improves as the number of selected bounding boxes increases. For example, when the number of selected bounding boxes per frame is 4, it only achieves a 44.1 recall@1 for sentence-to-video search and a 45.5 recall@1 for video-to-sentence search. In contrast, using 32 bounding boxes, it achieves a 51.6 recall@1 for sentence-to-video search and a 52.4 recall@1 for video-to-sentence search. Despite that the retrieval recall might be improved with more selected bounding boxes, we use only 32 boxes per video due to the limited computing resources.

Impact of the number of frames. For each video, we sample uniformly key frames from it for further processing. We vary the number of sampled frame among $\{2, 4, 8, 16, 32\}$ to testify the influence of the number of selected key frames on the retrieval performance. The experiments are conducted on VATEX dataset. As shown in Table 3, the recall@1 generally improves as the number of sampled key frames increases. For instance, using two frames per video, it only achieves a 15.0 recall@1 for the sentence-to-video search and a 16.4 recall@1 for the video-to-sentence search. In contrast, using 32 frames, we achieve a 33.2 recall@1 for sentence-to-video search and a 33.2 recall@1 for video-to-sentence search.

Table 3: The influence of number of frames. The experiments are conducted on VATEX dataset.

	s2v recall			v2s recall		
# frame	@1	@5	@10	@1	@5	@10
2	15.0	68.8	82.6	16.4	73.4	85.9
4	20.8	72.0	84.3	21.3	74.8	86.2
8	27.3	74.6	85.6	27.6	76.7	86.3
16	30.0	75.4	85.9	29.9	77.1	86.5
32	33.2	75.9	85.9	33.2	77.1	86.5

Table 4: The influence of modules on the proposed CAN.

CA	SA		s2v recall			v2s recall		
	text	video	@1	@5	@10	@1	@5	@10
	✓	✓	48.9	80.9	88.3	49.0	81.5	89.8
✓		✓	49.2	81.2	89.7	50.1	82.1	90.6
✓	✓		50.9	81.5	89.2	51.3	82.0	90.2
✓	✓	✓	51.6	81.9	89.4	52.4	82.3	90.5

Impact of the modules. We evaluate the influence of modules on the performance of the proposed CA in the retrieval accuracy. To be specific, we evaluate the performance of the proposed CAN by removing the CA block, SA blocks on the text side and SA blocks on the video side, respectively. The experiments are conducted on Daily700K dataset. As shown in Table 4, after removing the CA blocks, the performance becomes considerably worse, which validates the effectiveness of combo-attention modules. Meanwhile, by removing the text-side SA blocks, the considerably deteriorate the performance of the proposed CAN. To be specific the sentence-to-video recall@1 drops from 51.6 to 49.2. In contrast, the influence of the video-side SA blocks is relatively limited. This is due to that the features of bounding box are extracted from ResNet, which has already possessed good discriminating capability.

Influence of the number of CA blocks. Recall from Figure 3 that, we stacks two CA blocks in both video and sentence streams. We investigate the influence of the number of CA blocks on the retrieval performance. We vary the number of CA blocks among $\{0, 1, 2, 3\}$. As shown in Table 5, on the Daily700K dataset, the retrieval recall increases as the number of CA blocks increases. To be specific, the sentence-to-video recall@1 increases from 48.9 to 53.3 as the number of CA blocks increases from 0 to 3. In contrast, on VATEX dataset, the best retrieval recall is achieved when the number of CA blocks is 2. The worse performance using 3 CA blocks on VATEX dataset might be due to over-fitting as VATEX is relatively small. By default, we use 2 CA blocks on both datasets.

Table 5: The influence of the number of CA blocks.

(a) Daily700K

	s2v recall			v2s recall		
# CA	@1	@5	@10	@1	@5	@10
0	48.9	80.9	88.3	49.0	81.5	89.8
1	50.9	81.6	89.4	51.3	81.9	90.0
2	51.6	81.9	89.4	52.4	82.3	90.5
3	53.3	84.5	91.9	54.1	85.2	92.7

(b) VATEX

	s2v recall			v2s recall		
# CA	@1	@5	@10	@1	@5	@10
0	30.6	72.6	85.1	31.8	72.0	83.7
1	31.6	74.4	83.9	32.6	74.0	84.3
2	33.2	75.9	85.9	33.2	77.1	86.5
3	32.6	74.2	85.7	31.2	73.0	86.7

Influence of the number of heads. Since the SA blocks and CA blocks all adopt a multi-head settings. We further evaluate the influence of the number of head on the performance of the proposed CAN model. We conduct experiment on Daily700K dataset. In experiments, we vary the number of heads among $\{1, 2, 4, 8, 16\}$. As shown in Table 6, when the number of heads increases from 1 to 4, the performance of the proposed CAN becomes better. Meanwhile, the performance of SCAN is stable when the number of heads changes among $\{4, 8, 16\}$. By default, we use 8 heads.

Table 6: The influence of the head number on our CAN. The experiments are conducted on Daily700K dataset.

	s2v recall			v2s recall		
head #	@1	@5	@10	@1	@5	@10
1	50.5	81.0	88.6	51.4	81.4	89.1
2	50.9	80.9	88.9	51.7	81.4	89.2
4	51.4	81.8	89.5	52.2	82.3	90.1
8	51.6	81.9	89.4	52.4	82.3	90.5
16	51.1	81.8	89.6	51.5	82.2	90.1

5.3 Comparisons with Other Methods.

We compare the sentence-to-video retrieval performance of the proposed method with two recent state-of-the-art methods, SCAN [18] and VSE++ [9]. We use the codes provided by the authors of SCAN and VSE++, respectively. We test their performance on our built Daily700K as well as VATEX [31] dataset using identical features. As shown in Table 7, our CAN consistently outperforms SCAN and VSE++ on both datasets. To be specific, on Daily700K dataset, SCAN only achieves a 48.5 recall@1 for the sentence-to-video retrieval and 49.2 recall@1 for the video-to-sentence retrieval. Meanwhile, VSE++ SCAN only achieves a 43.8 recall@1 for the sentence-to-video retrieval and 45.5 recall@1 for the video-to-sentence retrieval. In contrast, sentence-to-video recall@1 of our CAN is 51.6, and video-to-sentence recall@1 of our CAN is 52.4.

Table 7: Comparisons with state-of-the-art methods on Daily700K and VATEX datasets.

(a) Daily700K

	s2v recall			v2s recall		
method	@1	@5	@10	@1	@5	@10
VSE++ [9]	43.8	78.5	88.0	45.5	79.5	88.6
SCAN [18]	48.5	80.8	89.3	49.2	81.1	90.0
CAN (ours)	51.6	81.9	89.4	52.4	82.3	90.5

(b) VATEX

	s2v recall			v2s recall		
method	@1	@5	@10	@1	@5	@10
VSE++ [9]	29.4	71.4	81.5	31.4	70.8	82.3
SCAN [18]	30.0	72.4	82.3	32.0	72.8	83.3
CAN (ours)	33.2	75.9	85.9	33.2	77.1	86.5

5.4 Online Results

We evaluate the proposed CAN in Baidu dynamic video advertising platform. Two online metrics are used to measure the performance: impression rate (IR) and conversion rate (CVR) defined as follows:

$$\mathrm{IR} = \frac{\text{\# of impressions}}{\text{\# of queries}}, \qquad \mathrm{CVR} = \frac{\text{revenue}}{\text{\# of cicks}}. \tag{16}$$

We compare the IR and CVR of Baidu dynamic video advertising platform before and after launching the proposed CAN. Note that, before launching CAN, the video search is based on title-based retrieval. As shown in Table 8, after launching the proposed CAN, the IR achieves a 11.08% and CVR achieves a 5.47% increase.

Table 8: Online results from Dec. 20th to Dec. 25th, 2019 in Baidu dynamic video advertising platform.

metric	IR	CVR
improvement	11.08%	5.47%

6 CONCLUSION

In this paper, we present the combo-attention network (CAN) launched in Baidu dynamic video adverting platform. CAN formulates the sentence-to-video search into a matching problem between a set of bounding boxes and a set of words. It exploits the cross-modal attentions besides self attentions. To evaluate the performance of CAN in short-video retrieval tasks, we built a video dataset consisting of 700K short videos collected from Haokan APP and label them based on users' clicks. Experiments conducted on our built Daily700K and the public VATEX datasets demonstrate the effectiveness of the proposed method. Meanwhile, the online experiments show the launch of CAN considerable boosts the revenue of Baidu dynamic video adverting platform.

ACKNOWLEDGMENTS

We are deeply grateful to the contributions of many colleagues from Baidu. A few names are Jie Liu, Shujing Wang, Lin Liu, Dingcheng Li, Hongliang Fei, Zhiqiang Xu; but there are certainly many more who have contributed to this large project.

REFERENCES

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. *Layer normalization.* Technical Report. arXiv:1607.06450.

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*. San Diego, CA.

[3] James Bradbury, Stephen Merity, Caiming Xiong, and Richard Socher. 2017. Quasi-Recurrent Neural Networks. In *Proceedings of the 5th International Conference on Learning Representations (ICLR)*. Toulon, France.

[4] Andrei Broder. 2002. A taxonomy of web search. *SIGIR Forum* 36, 2 (2002), 3–10.

[5] João Carreira and Andrew Zisserman. 2017. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. In *Proceedingts of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Honolulu, HI, 4724–4733.

[6] Harm de Vries, Florian Strub, Jérémie Mary, Hugo Larochelle, Olivier Pietquin, and Aaron C. Courville. 2017. Modulating early visual processing by language. In *Advances in Neural Information Processing Systems (NIPS)*. Long Beach, CA, 6594–6604.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*. Minneapolis, MN, 4171–4186.

[8] Jeff Donahue, Lisa Anne Hendricks, Marcus Rohrbach, Subhashini Venugopalan, Sergio Guadarrama, Kate Saenko, and Trevor Darrell. 2017. Long-Term Recurrent Convolutional Networks for Visual Recognition and Description. *IEEE Trans. Pattern Anal. Mach. Intell.* 39, 4 (2017), 677–691.

[9] Fartash Faghri, David J. Fleet, Jamie Ryan Kiros, and Sanja Fidler. 2018. VSE++: Improving Visual-Semantic Embeddings with Hard Negatives. In *Proceedings of the 2018 British Machine Vision Conference 2018 (BMVC)*. Newcastle, UK, 12.

[10] Daniel C. Fain and Jan O. Pedersen. 2006. Sponsored Search: A Brief History. In *SSA Workshop*. Ann Arbor, Michigan.

[11] Miao Fan, Jiacheng Guo, Shuai Zhu, Shuo Miao, Mingming Sun, and Ping Li. 2019. MOBIUS: Towards the Next Generation of Query-Ad Matching in Baidu's Sponsored Search. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, (KDD)*. Anchorage, AK, 2509–2517.

[12] Ali Farhadi, Seyyed Mohammad Mohsen Hejrati, Mohammad Amin Sadeghi, Peter Young, Cyrus Rashtchian, Julia Hockenmaier, and David A. Forsyth. 2010. Every Picture Tells a Story: Generating Sentences from Images. In *Computer Vision - ECCV 2010, Proceedings of the 11th European Conference on Computer Vision (ECCV)*. Heraklion, Greece, 15–29.

[13] Andrea Frome, Gregory S. Corrado, Jonathon Shlens, Samy Bengio, Jeffrey Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. 2013. DeViSE: A Deep Visual-Semantic Embedding Model. In *Advances in Neural Information Processing Systems (NIPS)*. Lake Tahoe, NV, 2121–2129.

[14] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional Sequence to Sequence Learning. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*. Sydney, Australia, 1243–1252.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas, NV, 770–778.

[16] Nal Kalchbrenner, Lasse Espeholt, Karen Simonyan, Aaron van den Oord, Alex Graves, and Koray Kavukcuoglu. 2016. *Neural machine translation in linear time.* Technical Report. arXiv:1610.10099.

[17] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. 2017. Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations. *Int. J. Comput. Vis.* 123, 1 (2017), 32–73.

[18] Kuang-Huei Lee, Xi Chen, Gang Hua, Houdong Hu, and Xiaodong He. 2018. Stacked Cross Attention for Image-Text Matching. In *Proceedings of the 15th European Conference on Computer Vision (ECCV)*. Munich, Germany, 212–228.

[19] Dingcheng Li, Xu li, Jun Wang, and Ping Li. 2020. Video Recommendation with Multi-gate Mixture of Experts Soft Actor Critic. In *Proceedings of the 43th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. Xi'an, China.

[20] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*. Vancouver, Canada, 13–23.

[21] Lin Ma, Zhengdong Lu, Lifeng Shang, and Hang Li. 2015. Multimodal Convolutional Neural Networks for Matching Image and Sentence. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*. Santiago, Chile, 2623–2631.

[22] Niluthpol Chowdhury Mithun, Juncheng Li, Florian Metze, and Amit K. Roy-Chowdhury. 2018. Learning Joint Embedding with Multimodal Cues for Cross-Modal Video-Text Retrieval. In *Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval (ICMR)*. Yokohama, Japan, 19–27.

[23] Mayu Otani, Yuta Nakashima, Esa Rahtu, Janne Heikkilä, and Naokazu Yokoya. 2016. Learning Joint Representations of Videos and Sentences with Web Image Search. In *Proceedings of the 10th European Conference on Computer Vision (ECCV Workshops)*. Amsterdam, The Netherlands, 651–667.

[24] Yingwei Pan, Tao Mei, Ting Yao, Houqiang Li, and Yong Rui. 2016. Jointly Modeling Embedding and Translation to Bridge Video and Language. In *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas, NV, 4594–4602.

[25] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. 2017. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* 39, 6 (2017), 1137–1149.

[26] Karen Simonyan and Andrew Zisserman. 2014. Two-Stream Convolutional Networks for Action Recognition in Videos. In *Advances in Neural Information Processing Systems (NIPS)*. Montreal, Canada, 568–576.

[27] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. 2019. VideoBERT: A Joint Model for Video and Language Representation Learning. In *Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. Seoul, Korea (South), 7463–7472.

[28] Du Tran, Lubomir D. Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. 2015. Learning Spatiotemporal Features with 3D Convolutional Networks. In *Proceedigns of the 2015 IEEE International Conference on Computer Vision (ICCV)*. Santiago, Chile, 4489–4497.

[29] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems (NIPS)*. Long Beach, CA, 5998–6008.

[30] Xiaolong Wang, Ross B. Girshick, Abhinav Gupta, and Kaiming He. 2018. Non-Local Neural Networks. In *Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Salt Lake City, UT, 7794–7803.

[31] Xiaolong Wang and Abhinav Gupta. 2018. Videos as Space-Time Region Graphs. In *Proceedings of the 15th European Conference on Computer Vision (ECCV)*. Munich, Germany, 413–431.

[32] Xin Wang, Jiawei Wu, Junkun Chen, Lei Li, Yuan-Fang Wang, and William Yang Wang. 2019. VaTeX: A Large-Scale, High-Quality Multilingual Dataset for Video-and-Language Research. In *Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. Seoul, Korea (South), 4580–4590.

[33] Ran Xu, Caiming Xiong, Wei Chen, and Jason J. Corso. 2015. Jointly Modeling Deep Video and Compositional Text to Bridge Vision and Language in a Unified Framework. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI)*. Austin, TX, 2346–2352.

[34] Youngjae Yu, Hyungjin Ko, Jongwook Choi, and Gunhee Kim. 2017. End-to-End Concept Word Detection for Video Captioning, Retrieval, and Question Answering. In *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Honolulu, HI, 3261–3269.

[35] Zhou Yu, Jun Yu, Yuhao Cui, Dacheng Tao, and Qi Tian. 2019. Deep Modular Co-Attention Networks for Visual Question Answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Long Beach, CA, 6281–6290.

[36] Weijie Zhao, Shulong Tan, and Ping Li. 2020. SONG: Approximate Nearest Neighbor Search on GPU. In *35th IEEE International Conference on Data Engineering (ICDE)*. Dallas, TX.

[37] Weijie Zhao, Deping Xie, Ronglai Jia, Yulei Qian, Ruiquan Ding, Mingming Sun, and Ping Li. 2020. Distributed Hierarchical GPU Parameter Server for Massive Scale Deep Learning Ads Systems. In *Proceedings of the 3rd Conference on Third Conference on Machine Learning and Systems (MLSys)*. Huston, TX.

[38] Weijie Zhao, Jingyuan Zhang, Deping Xie, Yulei Qian, Ronglai Jia, and Ping Li. 2019. AIBox: CTR Prediction Model Training on a Single Node. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management (CIKM)*. Beijing, China, 319–328.

[39] Zhixin Zhou, Shulong Tan, Zhaozhuo Xu, and Ping Li. 2019. Möbius Transformation for Fast Inner Product Search on Graph. In *Advances in Neural Information Processing Systems (NeurIPS)*. Vancouver, Canada, 8216–8227.