TACo: Token-aware Cascade Contrastive Learning for Video-Text Alignment

Jianwei Yang
Microsoft Research
jianwyan@microsoft.com

Yonatan Bisk
Carnegie Mellon University
ybisk@cs.cmu.edu

Jianfeng Gao
Microsoft Research
jfgao@microsoft.com

Abstract

Contrastive learning has been widely used to train transformer-based vision-language models for video-text alignment and multi-modal representation learning. This paper presents a new algorithm called ***T****oken-****A****ware* ***C****ascade c****o****ntrastive learning (****TACo****) that improves contrastive learning using two novel techniques. The first is the token-aware contrastive loss which is computed by taking into account the syntactic classes of words. This is motivated by the observation that for a video-text pair, the content words in the text, such as nouns and verbs, are more likely to be aligned with the visual contents in the video than the function words. Second, a cascade sampling method is applied to generate a small set of hard negative examples for efficient loss estimation for multi-modal fusion layers. To validate the effectiveness of* ***TACo****, in our experiments we finetune pretrained models for a set of downstream tasks including text-video retrieval (YouCook2, MSR-VTT and ActivityNet), video action step localization (CrossTask), video action segmentation (COIN). The results show that our models attain consistent improvements across different experimental settings over previous methods, setting new state-of-the-art on three public text-video retrieval benchmarks of YouCook2, MSR-VTT and ActivityNet.*

1. Introduction

Aligning or grounding language to videos is a challenging topic in the context of vision-language (VL) research as it requires the model to understand contents, dynamics, and causality presented in videos [3]. Inspired by the success of BERT [10] in natural language processing, there is a growing interest in applying transformer-based multi-modal models for video-text alignment and representation learning [40, 39, 59, 32, 14, 27]. These models are typically pretrained on large amounts of noisy video-text pairs using contrastive learning [34, 33], and then applied in a zero-shot manner or finetuned for various downstream tasks, such as text-video retrieval [51], video action step localization [60], video action segmentation [42], video question answering [43, 26] and video captioning [57].

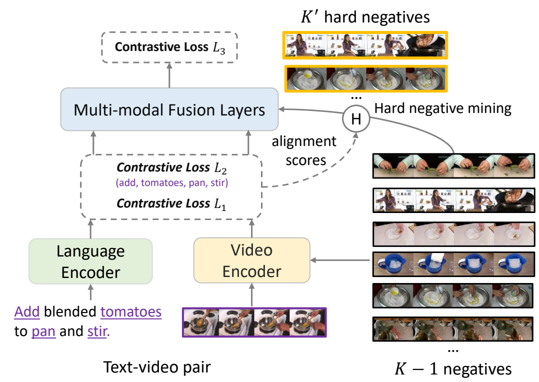

Figure 1: The proposed token-aware cascade contrastive learning pipeline. We compute three contrastive losses: 1) sentence-level loss L_1 over all negative examples; 2) token-level loss L_2 on content words (noun, verb) over all negative examples; 3) sentence-level loss L_3 over hard negative examples sampled based on L_1 and L_2 online.

In this paper, we present a new variant of contrastive learning, **T**oken-**A**ware **C**ascade c**o**ntrastive learning (**TACo**) to improve the video-text alignment for both large-scale pretraining and downstream specific tasks. As the name indicates, **TACo** makes two modifications to the conventional contrastive learning used in video-language domain. The first is the token-aware contrastive loss which is computed by taking into account the syntactic classes of words. This is motivated by the observation that, given a video and its corresponding text, content words, such as nouns and verbs, are more likely than function words to be aligned with (or grounded to) visual contents in the video. Conventional contrastive learning typically compute the loss after aggregating over all the words in the text and frames in the video (loss L_1 or L_3 in Fig. 1). In contrast, the token-aware contrastive loss is computed using only a subset of words whose syntactic classes belong to a predefined set (*e.g.*, nouns and verbs), which forces the grounding of individual words to the video (loss $L2$). For example, we pay particular attention to the words "add", "tomatos", "pan" and "stir" in Fig. 1.

The second technique we introduce is a cascade sampling method to find a small set of hard negative examples for training the multi-modal fusion layers. Consider a batch of K video-text pairs. For each of the video-text pairs, the ideal case is that we use the remaining $K-1$ negative videos or texts to compute the contrastive loss after multi-modal fusion. However, the cost of computing the contrastive loss quickly becomes prohibitive when it is coupled with multi-modal fusion layers, considering its high complexity $O(K^2 \times L^2)$ where L is total number of visual and textual tokens. A conventional way to address this is using random sampling to select a small subset of negative pairs. In this paper, instead of random sampling, we propose a cascade sampling method as shown in the top-right of Fig. 1 to efficiently select a small set of hard negative examples *on the fly* during training. It leverages the video-text alignment scores computed in L_1 and L_2 before multi-modal fusion layers, and helps to learn the multi-modal fusion layers more effectively without any extra overhead.

We perform a comprehensive empirical study to validate the effectiveness of **TACo** in both pretraining and dataset-specific scenarios. We apply **TACo** and different variants of contrastive losses to train or pretrain and finetune on various downstream tasks including text-video retrieval (YouCook2, MSR-VTT and ActivityNet) [57, 51, 12], video action step localization (CrossTask) [60] and action segmentation (COIN) [42]. Our results show that **TACo** improves the text-video retrieval performance over current state-of-the-art across three benchmarks. Furthermore, the learned multi-modal representation and video representation can be effectively transferred to CrossTask and COIN, and achieve better or comparable performance to current state-of-the-art methods.

2. Related work

Video-language pretraining. Realistic application scenarios around videos have prompted emergence of various video-language tasks, such as text-video retrieval [29, 54, 52], video question answering [20, 26], video captioning [53, 58], *etc*. Inspired by the success of BERT for large-scale pretraining in language domain [10], transformers have been employed in the video-language domain [40, 59, 32, 27] as well as image-language domain [41, 31, 56, 28]. Combined with large scale datasets, *e.g.* Howto100M [34] this approach has proven to be effective on various downstream tasks. Depending on the tasks of interest, some approaches train a multi-modal transformer using a combination of multiple losses including video-text alignment [40, 59, 32, 27], masked token (words/frames/objects) prediction [40, 59, 32], and frame order prediction [27], *etc*. Some other approaches exploited various contrastive learning techniques to directly optimize the feature space without multi-modal fusion [34, 33, 30, 14]. In most of previous works, these two approaches were explored separately. Very recently, an updated version of [32] used two independent alignment losses before and after multi-modal fusion in a single framework. In this paper, however, these two losses cooperate closely with each other during training in that the earlier stage helps to discover the hard negatives while the multi-modal layers with more capacity help to tackle those hard samples particularly.

Video-text alignment. Aligning videos to text requires the model to understand motion and temporal coherence. Some works have relied on attention mechanisms to extract key information from videos [44, 54], while others preserve visual information by composing pairwise joint representation using 3D tensors [52] or use multi-level video encoders to separately encode the spatial and temporal cues [11]. These models usually rely on a rank or margin loss to learn the correct alignment for video-text pairs. Another line of work learns fine-grained or hierarchical alignment between videos and texts [55, 48, 6]. In [48], the authors proposed a fine-grained alignment by extracting the nouns and verbs from action phrase in a sentence and projecting them into a shared space with videos. Alternatively, the authors in [6] extract a hierarchical semantic graph and apply graph reasoning to achieve the alignment at different levels. Similar ideas have been also proposed in the image-text alignment by decomposing the images and texts into sub-tokens [25, 49]. Thus far, it has not been studied how these task-specific architectures can be integrated into large-scale pretraining. In this paper, we are the first to propose a simple yet effective token-aware contrastive loss for fine-grained alignment for pretraining and downstream tasks.

Negative sampling. Key to efficient contrastive training is a good source of negative examples. Most of current approaches use random sampling strategies for training video-text alignment [59, 32]. However, in the domain of image-text retrieval, a few works tried hard negative sampling to choose the hardest negatives for training. In [2, 13], the authors computed the alignment scores for all image-text pairs in a mini-batch and use the hardest negative sample to compute the marginal loss. However, this strategy can only be applied without multi-modal fusion. In those models which have multi-modal fusion layers for better representations [31, 8], the authors instead compute the matching score offline and then use it to sample hard negatives for finetuning image-text retrieval model, which however is difficult for large-scale pretraining due to the high computational cost. In this paper, our cascade hard negative mining is particularly designed to address these issues as we efficiently select the hard negative samples online before multi-modal fusion and send them to the fusion layers for computing the loss. As we will show in our experiments, this technique can be seamlessly applied to both large-scale pretraining and downstream tasks.

3. Method

3.1. Framework

As depicted in Fig. 1, our model has three components:

Video encoding module f_{θ_v}. It is implemented by a stack of self-attention layers parameterized by θ_v. Here, we assume the input video features have been already extracted using some pre-trained models such as 2D CNN (*e.g.*, ResNet [18]) or 3D CNN (*e.g.*, I3D [4], S3D [50]). Given the input video embeddings, video encoder starts with a linear layer to project them to the same dimension d as following self-attention layers. We denote the output of our video encoder for a video clip by a sequence of m features, $\boldsymbol{x} = \{x^1, ..., x^m\} \in \mathbb{R}^{m \times d}$. The number of features m depends on the choice of sampling frame rate and the video feature extractor, which we will discuss in Sec. 4.

Language encoding module f_{θ_t}. We use pretrained tokenizer [47] and BERT [10] to tokenize the input texts and extract textual features, respectively. Given a raw sentence, we append a "[CLS]" and "[SEP]" to the beginning and end, respectively. At the top, we can obtain a sequence of n textual features $\boldsymbol{y} = \{y^1, ..., y^n\} \in \mathcal{R}^{n \times d}$. We ensure the output feature dimension of video encoder to be identical to that of language encoder. During training, we update the parameters θ_t in our language encoder to adapt to the texts in specific domain, *e.g.,* cooking instructions in YouCook2 [57].

Multi-modal fusion module f_{θ_m}. It also consists of self-attention layers with learnable parameters θ_m. It takes video features $\boldsymbol{x} \in \mathbb{R}^{m \times d}$ and text features $\boldsymbol{y} \in \mathbb{R}^{n \times d}$ from two separate modalities as inputs and output the $(m+n)$ features $\boldsymbol{z} = \{z_1, ..., z_{(m+n)}\} \in \mathcal{R}^{(m+n) \times d}$. To help it to distinguish the video and language tokens, we use a token type embedding layer to learn two embeddings and add them to the visual and textual tokens, separately. Similar to original Transformer [46], we include a positional embedding layer to encode the absolute token positions in the input sequence.

The above three components comprise our video-text alignment model which is then trained with the proposed token-aware cascade contrastive loss. We start with a brief review of conventional contrastive learning and then introduce the proposed technique.

3.2. Contrastive learning: a revisit

Given a set of N video-text pairs $\{(v_i, t_i)\}_{i=1}^N$, our goal is to learn an optimal scoring function s such that paired video and text (v_i, t_i) have higher scores than all the other unmatched pairs $(v_j, t_k), j \neq k$. From the probabilistic perspective, aligning v_i to t_i is equivalent to maximizing the conditional probability $p(v_i|t_i)$ while minimizing the probability for all negative pairs $p(v_j|t_i), j \neq i$. According to [15, 36], $p(v_j|t_i)$ can be approximated by:

$$p(v_j|t_i) \sim \frac{\exp^{s(v_j,t_i)}}{\sum_{k=1}^N \exp^{s(v_k,t_i)}} \tag{1}$$

where $s(v,t)$ is the alignment score between v and t; the denominator is a sum over all possible videos, which is a partition function for normalization. Adding cross-entropy loss on $p(v_j|t_i)$, we can then derive the NCE loss [15]:

$$\begin{aligned} L_{nce} &= \sum_{i=1}^N -\log p(v_i|t_i) \\ &\sim \sum_{i=1}^N -\log \left(\frac{\exp^{s(v_i,t_i)}}{\exp^{s(v_i,t_i)} + \sum_{k \neq i} \exp^{s(v_k,t_i)}} \right) \end{aligned} \tag{2}$$

The denominator in Eq. 2 requires a sum over all videos in a dataset, which is intractable in practice. Therefore, we usually compute the NCE loss on a mini-batch of $K (K \ll N)$ video-text pairs sampled from the whole dataset. Ideally, we want to learn the parameters $\theta = \{\theta_v, \theta_t, \theta_m\}$ of the model to minimize the above NCE loss, such that $\Delta = s(v_i, t_i) - s(v_j, t_i)$ is maximized over all tuples $(t_i, v_i, v_j), j \neq i$. A number of previous works used the above formula for contrastive learning [33, 59]. Meanwhile, there are some variants of computing contrastive loss in video-langauge representation learning. For example, [27, 14] omits the denominator and incorporate a margin s.t. $s(v_i, t_i) > s(v_j, t_i) + \delta, \forall j \neq i$ in a mini-batch. [32] optimizes binary cross-entropy (BCE) by assigning (v_i, t_i) a positive label (1) and other pairs a negative label (0).

3.3. **TACo**: our approach

The way of using contrastive learning in previous works has two issues. First, the loss is computed at sentence-level by taking '[CLS]' token [14] or the maximum over all tokens [33] in a sentence. Clearly, the content words (*e.g.*, nouns, verbs) are more likely to align with the visual contents or concepts in the videos compared with function words (*e.g.*, stop words). Second, the high computational cost in multi-modal fusion layers hinder the usage of large batch of negative samples, which however is essential to contrastive learning [33, 17, 7]. Motivated by these two issues, we introduce **TACo**, a simple yet effective method to improve the contrastive learning. We elaborate below how these contrastive losses are computed.

Given the K video-text pairs $\{(v_i, t_i)\}_{i=1}^K$ in a mini-batch, we first use our video encoder f_{θ_v} and language encoder f_{θ_t} to obtain a batch of video features $X = \{x_1, ..., x_K\} \in \mathcal{R}^{K \times m \times d}$ and text features $Y = \{y_1, ..., y_K\} \in \mathcal{R}^{K \times n \times d}$, respectively. Then, we average all tokens of a video clip v_i to get $\bar{x}_i \in \mathcal{R}^{1 \times d}$, and take the first '[CLS]' token for each text t_i to get $\bar{y}_i \in \mathcal{R}^{1 \times d}$. Based on $\bar{x}$ and $\bar{y}$, we compute the sentence-level contrastive loss:

$$L_1 = -\sum_{i=1}^{K} \log \left(\frac{\exp^{\bar{x}_i \cdot \bar{y}_i / \tau_1}}{\exp^{\bar{x}_i \cdot \bar{y}_i / \tau_1} + \sum_{j \neq i} \exp^{\bar{x}_j \cdot \bar{y}_i / \tau_1}} \right) \quad (3)$$

where τ_1 is a scalar temperature parameter. In Eq. 3, the computation is simply a number of dot-products between video and text features. Giving such efficiency, we can use all the $K-1$ negative samples in a mini-batch to compute the loss. Through this, we optimize θ_v and θ_t so as to project the video and text samples into an aligned feature space.

The '[CLS]' token and average of video tokens in Eq. 3 overlooks the differences across tokens and frames, and thus may not provide the pressure to push individual tokens (*e.g.*, nouns and verbs) to ground on the specific video contents. To encourage correct alignment, in addition to the sentence-level loss, we introduce a token-level contrastive loss:

$$L_2 = -\sum_{i=1}^{K} \sum_{p \in \mathcal{P}_i} \log \left(\frac{\exp^{s(x_i, y_i^p)/\tau_2}}{\exp^{s(x_i, y_i^p)/\tau_2} + \sum\limits_{j \neq i} \exp^{s(x_j, y_i^p)/\tau_2}} \right) \quad (4)$$

where τ_2 is another scalar temperature parameter; P_i is the indices of tokens of interest in i-th text, and y_i^p is the p-th token embedding in i-th text. $s(\cdot)$ measures the similarity between video features and specific token embedding y_i^p. It first computes the dot-product between $y_i^p \in \mathcal{R}^{1 \times d}$ and all m video tokens $x \in \mathcal{R}^{m \times d}$, and then take the maximum over m scores to get the final alignment score. Through Eq. 4, the model uses individual tokens as anchors to align with video, which is complementary to the sentence-level loss in Eq. 3. Similar to Eq. 3, we can compute this token-level contrastive loss efficiently, and thus use all the $K-1$ negative samples. As a whole, these two losses are used to optimize θ_v and θ_t in a token-aware manner.

Token of interest. In Eq. 4, we need to decide which tokens should be included in P_i. In this paper, we heuristically select *nouns* and *verbs* as the targets considering they are more "concrete" in the videos. In practice, *nouns* or *verbs* usually have different discriminativeness even if they are all the same type. For example, "man" is a *noun* but is less informative than "gymnast". To reflect this, we further assign different words with different weights by computing their inverse document frequency (idf) [21]. A higher idf means it is more unique across the corpus, and hence will weigh more when computing the token-level contrastive loss. Another practical issue for computing the loss is that the tokens are usually sub-words due to the BERT tokenizer. Hence, for all tokens that belongs to the same word, we will assign the same weights accordingly.

After computing the token-aware contrastive loss, we feed the features from separate modalities to multi-modal fusion layers to enable more interactions between them two. Similar to previous work [59], we take the feature corresponding to the "[CLS]" in the $(m+n)$ outputs. We regard this as the summary of two modalities and then compute the contrastive loss:

$$L_3 = -\sum_{i=1}^{K} \log \left(\frac{\exp^{w \cdot z_{i,i}^{cls}}}{\exp^{w \cdot z_{i,i}^{cls}} + \sum_{j \neq i} \exp^{w \cdot z_{j,i}^{cls}}} \right) \quad (5)$$

where $z_{j,i}^{cls}$ is the multi-modal fusion output for "[CLS]" token taking x_j and y_i as inputs; $w \in \mathcal{R}^{1 \times d}$ is the parameter in a linear layer[1]. Based on Eq. 5, we optimize all parameters in our model $\theta = \{\theta_v, \theta_t, \theta_m\}$ in collaboration with Eq. 3 and Eq. 4.

In Eq. 5, a practical challenge is that we can hardly use all $(K-1)$ negative samples in the mini-batch, due to the high computational and memory cost in the multi-modal fusion. The $O(d(m+n)^2)$ complexity of self-attention layer makes it intractable to pass all $K \times K$ pairs into the multi-modal layers. Previous work solved this by performing random sampling to cut the number of negative samples to K'. However, randomly choosing negative samples may result in sub-optimal learning since the pairs are scarce. We therefore introduce a cascade sampling strategy to find hard negatives instead of random ones.

Cascade hard negative sampling. To reduce the computational cost in Eq. 5, we choose among all possible video-text pairs a small subset which are most difficult. However, computing the alignment scores for all pairs using Eq. 5 and then select the hard negatives is a "chicken-and-egg" problem. Instead, we propose to use the similarities between all video-text pairs computed in Eq. 3 and Eq. 4 as the guidance. Specifically, for each text-video pair (v_j, t_i), we take their global similarity $\bar{x}_j \cdot \bar{y}_i$ computed in Eq. 3 and token-level similarity by aggregating $\sum_{p \in P_i} s(x_j, y_i^p)$ for all tokens of interest in t_i. Then we sum the two similarities as the alignment score for the given pair. For each text, we choose the top K' aligned negative videos and vice versa. The resulting $2K \times (K'+1)$ pairs are then fed into the multi-modal fusion layers. Through this strategy, we can effectively select the difficult negative samples on the fly at no extra cost. Since the multi-modal fusion layers has more capacity (parameters) to distinguish these hard negatives from positive ones, our sampling strategy naturally prompts the cooperation between the three contrastive losses.

Finally, we present a comprehensive comparison to differentiate our model with previous works with respect to the used contrastive learning method in Table 1.

3.4. Objective

The training objective in our method is finding optimal $\theta = \{\theta_v, \theta_t, \theta_m\}$ by minimizing the combination of the above three contrastive losses:

$$\arg \min_{\theta_v, \theta_t, \theta_m} \sum_{i=1}^{N} (L_1 + \lambda_t L_2 + L_3) \quad (6)$$

[1] for clarity, we omit the bias term in the formula

Method	Token-aware	Early stage	Later stage	Cascade	Loss
VideoBert [40]	✗	✗	✓	✗	BCE
CBT [39]	✗	✗	✓	✗	NCE
TJVE [34]	✗	✓	✗	✗	Margin
MIL-NCE [33]	✗	✓	✗	✗	NCE
ActBert [59]	✗	✗	✓	✗	BCE
UniVL [32]	✗	✓	✓	✗	NCE
MMT [14]	✗	✓	✗	✗	Margin
TACo(Ours)	✓	✓	✓	✓	NCE

Table 1: A comparison of video-language pretraining methods regarding contrastive learning strategies. "Early stage" and "Later stage" mean computing the loss before and after multi-modal fusion, respectively. "Cascade" means using cascade hard negative sampling.

where λ_t is the weight of token-level loss (0.5 by default). During inference, we make the prediction by summing the alignment scores from all the three scoring functions.

4. Experimental setup

4.1. Datasets

In our experiments, we train and evaluate our model on the following established benchmarks:

• **YouCook2** [57] consists of 2k videos about routine cooking activities of 89 recipes. Each video contains multiple video clips annotated with text descriptions by human annotators. Following [34, 33], we train our models on the training split, and report the text-video retrieval performance on around 3.5k validation clips.

• **MSR-VTT** [51] contains 10k video clips associated with 200k sentences. There are two validation splits used in previous work. In [30, 14], the training set has 9k clip-text pairs with the remaining 1k pairs for evaluation, which we denote by *split1*. In [52, 34, 33], 1k clip-text pairs are sampled from the 3k pairs in test set for evaluation, while the original 7k pairs are used for training. We denote this by *split2*. We report text-video retrieval results using both splits.

• **ActivityNet** [24]. It consists of 20K YouTube videos, each of which is associated with multiple human-annotated captions. Following [55, 14], we concatenate all the captions for a video into a paragraph and evaluate the paragraph-video retrieval on the "val1" split.

• **Howto100M** [34]. We compare with previous work under the pretraining protocol on Howto100M [34, 33, 59, 32]. It was collected from YouTube and contains over 1.2M narrated videos associated with automatically generated transcripts. Each video contains over 100 clips on average.

To further verify the transferrability or our learned multi-modal representation from Howto100M, we also evaluate the action step localization and action segmentation on CrossTask [60] and COIN [42], respectively.

4.2. Settings

Previous work use a variety of different video and language representations which we find significantly affect the final performance. We summarize different choices below:

• **Video representations**. For 2D CNN, Resnet-152 [18] is used to extract feature map and then globally pooled to 2048-d [34, 32]. For 3D features, commonly used models are I3D [5], R(2+1)D [45] and S3D [50]. In [59], the authors further extract objects from the video clips. In [30, 14], the authors use collaborative experts to extract features from audio, scene, OCR, face, speech, etc.

• **Language representations**. There are primarily four variants: 1) GoogleNews pretrained word2vec (w2v) [35] used in [30, 34, 33]; 2) LSTM or Bidirectional LSTM [19]; 3) pretrained BERT [10] used in [40, 59, 32, 14] and 4) OpenAI-GPT [37] used in [30].

In this paper, we use a pretrained BERT-base model for language representation as in [59, 32]. For video features, following [34, 33, 32], we extract 2D CNN features using Resnet-152 (R-152) pretrained on ImageNet [9]. For 3D CNN features, we use I3D (with Resnext-101 backbone) pretrained on Kinetics-400 [22] and S3D [50] pretrained on Howto100M [33]. The off-the-shelf pretrained weights are provided by [16] and [33]. For simplicity, we denote them by I3D-X101 and S3D-HM in the following.

Another discrepancy among different methods is the number of self-attention layers used in the model. In [59], the authors use 12 multi-modal self-attention layers while 6 video encoder layers and 2 multi-modal fusion layers are used in [32]. Differently, 4 multi-modal self-attention layers are used in [14]. In this paper, for all our ablation studies below, we use 1 and 2 self-attention layers for our video encoder and multi-modal fusion, respectively. To compare with previous work on specific dataset, we use 2 video encoding layers. While pretraining the model with large-scale dataset Howto100M [34], we increase to 4 video encoding layers for comparable model capacity to previous works [59, 32, 14]. Note that this largest model is still smaller than or on par with the aforementioned methods.

4.3. Implementation details

For YouCook2 and MSR-VTT, the maximum number of video and text tokens are set to 48 and 30, respectively. For paragraph-video retrieval on ActivityNet, we set them both to 256. The 2D R-152 feature is extracted for one frame per second, and then globally pooled to 2048-d. For 3D CNN features, we follow [34] to sample video frames at 24 fps and extract an I3D-X101 feature every 16 frames. This results in 1.5 2048-d feature per second. For Eq. 3 and 4, we set the temperatures τ_1 and τ_2 both equal to 1.

Training on separate datasets. In this setting, we train models from scratch using the training set provided in YouCook2, MSR-VTT and ActivityNet separately. We train

Video Representation	YouCook2 R1↑	R5↑	R10↑	MR↓	MSR-VTT (*split1*) R1↑	R5↑	R10↑	MR↓
R-152, Baseline	4.1	13.2	19.4	81.0	16.4	42.6	55.8	8.0
R-152, Ours	**4.6**	**14.1**	**20.4**	**71.0**	**18.9**	**46.2**	**58.8**	**7.0**
I3D-X101, Baseline	2.1	8.1	12.7	125.0	14.7	40.83	53.2	9.0
I3D-X101, Ours	**2.6**	**8.9**	**13.2**	**115.0**	**20.6**	**44.0**	**56.9**	**7.0**
R-152+I3D-X101, Baseline	4.2	13.5	20.0	75.0	16.6	45.4	58.5	7.0
R-152+I3D-X101, Ours	**4.7**	**14.3**	**21.9**	**68.0**	**23.1**	**50.5**	**64.0**	**5.0**
S3D-HM, Baseline	13.8	37.2	51.1	10.0	18.7	47.2	62.2	6.0
S3D-HM, Ours	**16.1**	**40.3**	**52.2**	**9.0**	**23.9**	**51.4**	**65.0**	**5.0**
R-152+S3D-HM, Baseline	13.3	35.8	48.9	11.0	21.4	48.1	61.5	6.0
R-152+S3D-HM, Ours	**15.8**	**39.8**	**52.4**	**10.0**	**24.5**	**52.8**	**65.5**	**5.0**

Table 2: Text-video retrieval performance on YouCook2 and MSR-VTT with different feature types. S3D pretrained on HowTo100M outperforms others with large margin.

the model for 30k iterations with batch size 128. For each training sample, we use our cascade sampling strategy to sample 8 hard negatives. We use Adam [23] as the optimizer with initial learning rate $1e^{-4}$. A linear learning rate decay is applied after 5k warm-up iterations. The weight decay is set to $1e^{-5}$.

Pretraining and finetuning. We pretrain our model on Howto100M [34]. Since the original annotated video clips in Howto100M are usually short with a few seconds, we merge the adjacent clips so that the resulted text has at least 10 words. We use Adam [23] as the optimizer with initial learning rate $1e^{-4}$. We train the model for 500k iterations with batch size 64, and also sample 8 hard negatives for each sample using our cascade sampling strategy. After pretraining, we finetune the pretrained models on different datasets using the same setting as above except for a lower initial learning rate $2e^{-5}$ and less finetuning iterations 20k.

Evaluation metrics. For text-video retrieval, we use Recalls at different points (Recall@n or Rn, with n as a specific number) and Median Rank (MR) as the metrics following previous works [59, 32]. In all tables, we use ↑ or ↓ to indicate higher or lower is better, respectively.

5. Results

We first evaluate text-video retrieval performance and then study whether the learned representations can be transferred to other tasks on CrossTask and COIN.

5.1. Text-video retrieval

5.1.1 Comparing with baselines

We first show the comparisons with baselines to inspect the effects of different components in our model.

Video representations. We train our model with different video representations as described above and compare it with the baseline model which has identical architecture but merely trained with L_3 as depicted in Eq. 5. The baseline

Losses	Cascade	YouCook2 R1↑	R5↑	R10↑	MR↓	MSR-VTT (*split1*) R1↑	R5↑	R10↑	MR↓
L_1	n/a	14.1	35.7	48.8	11.0	22.9	49.7	61.7	6.0
L_3	n/a	13.3	35.8	48.9	11.0	21.4	48.1	61.5	6.0
$L_1 + L_3$	✗	13.9	37.4	50.7	10.0	22.5	50.8	64.1	5.0
$L_1 + L_3$	✓	15.0	38.7	51.3	10.0	23.7	51.3	63.9	5.0
$L_1 + L_2 + L_3$	✓	**15.8**	**39.8**	**52.4**	**10.0**	**24.5**	**52.8**	**65.5**	5.0

Table 3: Text-video retrieval performance with different technique ensembles. It shows that using our proposed two techniques produce best results. All experiments use R-152+S3D-HM video features.

Token of Interest	YouCook2 R1↑	R5↑	R10↑	MR↓	MSR-VTT (*split1*) R1↑	R5↑	R10↑	MR↓
None	15.0	38.7	51.3	10.0	23.7	51.3	63.9	5.0
det+adp	14.7	38.5	51.2	10.0	23.3	51.0	63.5	5.0
noun	15.4	39.3	51.8	10.0	24.0	51.8	65.1	5.0
verb	15.3	39.0	51.4	10.0	23.9	52.1	64.8	5.0
noun+verb	**15.8**	**39.8**	**52.4**	10.0	**24.5**	**52.8**	**65.5**	5.0

Table 4: Text-video retrieval performance with different tokens of interest for computing token-level contrastive loss. "det" means determiner; "adp" means adposition. We use the same video features as in Table 3.

contrastive learning method has been adopted in a number of previous works [59, 32]. This comparison can verify the effectiveness of our proposed contrastive learning method considering two models have exactly the same number of parameters. In Table 2, we can see our proposed method outperforms baseline across all feature types introduced in Sec. 4.2 on both YouCook2 and MSR-VTT. Note that our model uses exactly the same number of parameters to the baseline model. These consistent improvements demonstrate the effectiveness and generalization ability of our proposed method. As mentioned above, we also observe the text-video retrieval performance significantly depends on the feature types. We can find 3D features (I3D-X101 and S3D-HM) in general outperform 2D feature (R-152), which is expected since 2D feature does not capture the motions in the videos. Among all three feature types, S3D-HM outperforms the other two with large margin, which demonstrates the potential to learn good video representation by pretraining on large-scale noisy dataset (Howto100M [34]). Because Howto100M mainly contains instructional videos, it is more close to YouCook2 than MSR-VTT, and hence we see more gain on YouCook2. These comparisons indicate video representations matter much to the final performance.

Component Analysis. In our method, we combine L_1, L_2, and L_3 during training and inference. Here, we study how they perform separately and contribute to the final performance. In Table 2, we use R-152+S3D-HM as the video feature and report the results with different loss combina-

Model	Lang.	Video	YouCook2			
			R1↑	R5↑	R10↑	MR↓
Random	–	–	0.0	0.2	0.3	1675
TVJE [34]	w2v	R-152+I3D-X101	4.2	13.7	21.5	65
UniVL(v1) [32]	BERT	R-152+I3D-X101	3.4	10.8	17.8	76
TACo (Ours)	BERT	R-152+I3D-X101	**4.9**	**14.7**	**21.7**	**63**
UniVL(v3) [32]	BERT	S3D-HM	7.7	23.9	34.7	21
TACo (Ours)	BERT	S3D-HM	**16.6**	**40.3**	**53.1**	**9.0**

Table 5: Comparing text-video retrieval on YouCook2.

Model	Lang.	Video	MSR-VTT			
			R1↑	R5↑	R10↑	MR↓
Random	–	–	0.1	0.5	1.0	500.0
JSFusion [52]	BiLSTM	R-152	10.2	31.2	43.2	13.0
JPoSE [48]	w2v	TSN+Flow	14.3	38.1	53.0	9.0
TVJE [34]	w2v	R-152+I-101	12.1	35.0	48.0	12.0
UniVL(v1)* [32]	BERT	R-152+I-101	14.6	39.0	52.6	10.0
TACo (Ours)	BERT	R-152+I-101	**19.2**	**44.7**	**57.2**	**7.0**
CE [30]	GPT	Collaborative Experts	20.9	48.8	62.4	6.0
MMT [14]	BERT	Collaborative Experts	24.6	54.0	67.1	**4.0**
TACo (Ours)	BERT	R-152+S3D-HM	**26.7**	**54.5**	**68.2**	**4.0**

Table 6: Comparing text-video retrieval on MSR-VTT. The upper block and bottom block use *split2* and *split1*, respectively. We report them separately for fair comparison.

Model	Lang.	Video	ActivityNet			
			R1↑	R5↑	R10↑	MR↓
Random	-	-	0.02	0.1	1.02	2458
DenseCap [24]	LSTM	C3D	14.0	32.0	65.0	34
FSE [55]	GRU	C3D+TSN-Inception	18.2	44.8	89.1	7.0
CE [30]	GPT	Collaborative Experts	18.2	47.7	91.4	6.0
MMT [14]	BERT	Collaborative Experts	22.7	54.2	93.2	5.0
TACo (Ours)	BERT	R-152+S3D-HM	**25.8**	**56.3**	**93.8**	**4.0**

Table 7: Comparing text-video retrieval on ActivityNet.

tions. As we can see, solely using L_1 (row 1) or L_2 (row 2) for contrastive learning results in sub-optimal video-text alignment. Simply combining them together (row 3) improves the performance on two datasets. This implies that different levels of contrastive learning can be complementary to each other, which supports our earlier hypothesis that these two losses are synergistic with each other for a better video-text alignment. When incorporating the hard negative mining via our cascade sampling strategy (row 4), it further improves the performance. Finally, we can see adding token-level contrastive loss L_3 can further improve the performance across all settings (row 5).

Tokens of Interest. We further study the effect of different tokens of interest on the model performance. By default, our model uses the noun and verb as the tokens of interest to compute the token-level contrast loss. Here, we vary them to other types such as adposition (adp) and determiner (det) for investigation. In Table 4, we replace "noun+verb"

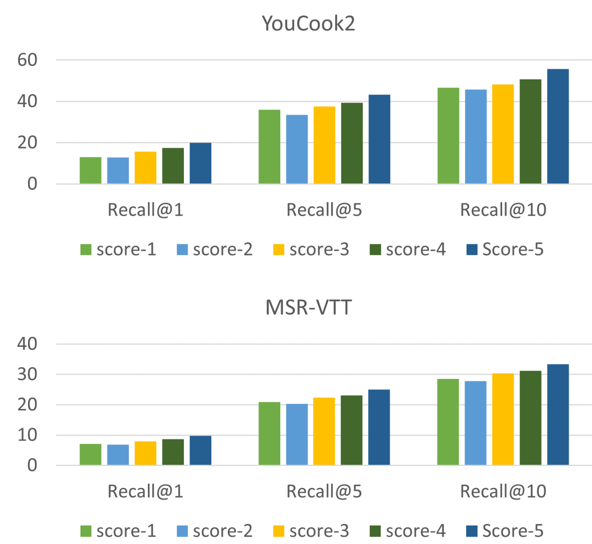

Figure 2: Zero-shot performance on YouCook2 and MSR-VTT for different settings. score-1-5 correspond to the five settings in Table 3 from top to bottom.

with "det+adp", "noun" and "verb" and report the numbers on two text-video retrieval datasets. As we can see, using "det+adp" as the target tokens is worse than the baseline without any token-level contrastive loss. "noun" and "verb" can both improve the performance while "noun" is slightly better than "verb". Finally, combining noun and verb together achieves the best performance. These results align with our intuition to use nouns and verbs as the target token for fine-grained alignment between texts and videos considering they are usually grounded to video contents.

5.1.2 Comparing with state-of-the-art

We compare with previous works under three protocols: 1) training and evaluating on separate datasets; 2) pretraining on Howto100M and evaluating zero-shot performance and 3) finetuning pretrained model on separate datasets.

Results on separate datasets. We separately show the comparisons on YouCook2, MSR-VTT and ActivityNet in Table 5, 6 and 7. For a fair comparison with previous works, we use the same or similar features as listed in the tables. As we can see, our method outperforms all previous work across all datasets. These results validates its effectiveness to learn video-text alignment. Note that previous works either use a variety of loss functions [32, 27] or a collection of multiple features [30, 14]. In contrast, we achieve the best performance using a *simpler* contrastive learning pipeline with smaller model size. This supports our earlier claim on the efficiency. Comparing the numbers in Table 2, Table 5 and Table 6, we can find our model achieves better performance with the same video features when using deeper video encoder (2 layers *v.s.* 1 layer).

	Model	Video	YouCook2				MSR-VTT				ActivityNet			
			R1↑	R5↑	R10↑	MR↓	R1↑	R5↑	R10↑	MR↓	R1↑	R5↑	R50↑	MR↓
Zero-shot	TJVE [34]	R-152+I-101	6.1	17.3	24.8	46.0	7.5	21.2	29.6	38.0	–	–	–	–
	ActBERT [59]	O-101+ R(2+l)D	9.6	26.7	38.0	19.0	8.6	23.4	33.1	36.0	–	–	–	–
	MIL-NCE [33]	S3D-HM	15.1	38.0	51.2	10.0	**9.9**	24.0	32.4	29.5	–	–	–	–
	TACo (Ours)	S3D-HM	**19.9**	**43.2**	**55.7**	**8.0**	9.8	**25.0**	**33.4**	**29.0**	–	–	–	–
Finetuned	TJVE [34]	R-152+I3D-X101	8.2	24.5	35.3	24.0	14.9	40.2	52.8	9.0	–	–	–	–
	UniVL(v3) [32]	S3D-HM	28.9	57.6	70.0	**4.0**	21.2	49.6	63.1	6.0	–	–	–	–
	TACo (Ours)	S3D-HM	**29.6**	**59.7**	**72.7**	**4.0**	**24.8**	**52.1**	**64.0**	**5.0**	28.3	56.8	92.6	4.0
	MMT [14]	Collaborative Experts	–	–	–	–	26.6	57.1	69.6	**4.0**	28.7	**61.4**	**94.5**	3.3
	TACo (Ours)	R-152+S3D-HM	27.3	56.5	68.8	**4.0**	**28.4**	**57.8**	**71.2**	**4.0**	**30.4**	61.2	93.4	**3.0**

Table 8: A complete comparison of **TACo** under zero-shot and finetuning evaluation protocols. Note that the zero-shot and upper part of finetuned performance for MSR-VTT is on *split2*, while the bottom is on *split1* for fair comparison.

Method	CrossTask	COIN
Alayrac *et al.* [1]	13.3	–
Zhukov *et al.* [60]	22.4	–
Supervised [60]	31.6	–
NN-Viterbi [38]	–	21.2
CBT [39]	–	53.9
TVJE [34]	33.6	–
MIL-NCE [33]	40.5	61.0
ActBert [59]	41.4	57.0
UniVL(v3) [32]	42.0	**70.0**
TACo (Ours)	**42.5**	68.4

Table 9: Action step localization on CrossTask (avg. recall) and action segmentation on COIN (acc.).

Zero-shot and finetuned performance. In Table 8, we show the comparisons across different models pretrained on Howto100M. In the upper part of the table, we compare the zero-shot performance on YouCook2 and MSR-VTT. We do not evaluate on ActivityNet since it has different number of input video tokens compared with the pretrained model and thus is not directly compatible to the pretrained model. As we can see, **TACo** outperforms previous works significantly on YouCook2 and slightly on MSR-VTT. Since YouCook2 has closer domain gap to Howto100M than MSR-VTT, the improvement brought by large-scale pretraining is more significant. However, on MSR-VTT, our model still outperforms MIL-NCE [33] which uses the same video features. In Fig. 2, we show the zero-shot performance on YouCook2 and MSR-VTT when pretraining our models with different contrastive losses as listed in Table 3. Accordingly, it shows our proposed contrastive losses gradually improve the performance, and combining all techniques achieves the best performance. Based on the pretrained model, we further finetune it on specific datasets. In our experiments, we use two feature S3D-HM and R-152+S3D-HM, to compare with the methods with the same/similar settings. As we can see, our model using S3D-HM outperforms UniVL [32] using the same feature but more video encoder layers (6). Different from zero-shot results, we observe more improvement on MSR-VTT than YouCook2 after finetuning. This implies that finetuning on specific datasets can compensate the domain gap to the pretraining datasets. To compare with the methods using features extracted from collaborative experts [14], we enrich our video representation by adding 2D R-152 feature, which achieves better performance on MSR-VTT, and better Recall@1 and Median Rank on ActivityNet. Note that this combination hurts the performance on YouCook2, and we witnessed a similar trend for models without pretraining in Table 2. Finally, comparing with the results without pretraining in Table 5, 6 and 7, we clearly find large-scale pretraining and finetuning brings substantial improvements consistently.

5.2. Other video-related tasks

Following [34, 59, 32], we evaluate action step localization performance on CrossTask dataset [60]. It covers 18 tasks and each video contains multiple video segments annotated with action steps and natural language descriptions. Similar to [34, 59, 32], we use our model to compute the similarity between each frame and the action step descriptions, which results in a score matrix. Using the official algorithm provided by [60], we can find the optimal frame-wise order of action steps for a video. By comparing it with the ground-truth annotations, we compute the recall for each task and then do the average. According to the results in Table 9, our model achieves the best performance compared with previous works. This indicates that our model can learn good video-language representations.

We further evaluate our pretrained model on action segmentation task on COIN dataset, following [33, 59]. Unlike the above task, action segmentation does not rely on texts, and thus can be used to evaluate the learned video representation. As shown in Table 9, our method significantly outperforms MIL-NCE and ActBert, and achieves comparable performance to UniVL. This indicates that our model is also a good video representation learner.

6. Conclusion

In this paper, we introduced **TACo**, a simple yet effective contrastive learning method for learning video-text alignment. It is aimed at addressing two existing issues in current contrastive learning pipelines: missing fine-grained alignment and inefficient sampling for multi-modal fusion. Without introducing any extra parameters, our method achieved promising results on three text-video retrieval benchmarks under various evaluation protocols. We further demonstrated the learned representations can be effectively transferred to other tasks such as action step localization and segmentation. Based on all these encouraging results, we believe **TACo** is a good alternative to conventional contrastive learning pipeline.

References

[1] Jean-Baptiste Alayrac, Piotr Bojanowski, Nishant Agrawal, Josef Sivic, Ivan Laptev, and Simon Lacoste-Julien. Unsupervised learning from narrated instruction videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4575–4583, 2016. 8

[2] Srikar Appalaraju and Vineet Chaoji. Image similarity using deep cnn and curriculum learning. *arXiv preprint arXiv:1709.08761*, 2017. 2

[3] Yonatan Bisk, Ari Holtzman, Jesse Thomason, Jacob Andreas, Yoshua Bengio, Joyce Chai, Mirella Lapata, Angeliki Lazaridou, Jonathan May, Aleksandr Nisnevich, Nicolas Pinto, and Joseph Turian. Experience grounds language. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020. 1

[4] Joao Carreira and Andrew Zisserman. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 05 2017. 3

[5] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017. 5

[6] Shizhe Chen, Yida Zhao, Qin Jin, and Qi Wu. Fine-grained video-text retrieval with hierarchical graph reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10638–10647, 2020. 2

[7] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020. 3

[8] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Universal image-text representation learning. In *European Conference on Computer Vision*, pages 104–120. Springer, 2020. 2

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 5

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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, Volume 1 (Long and Short Papers)*, 10 2019. 1, 2, 3, 5

[11] Jianfeng Dong, Xirong Li, Chaoxi Xu, Shouling Ji, Yuan He, Gang Yang, and Xun Wang. Dual encoding for zero-example video retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9346–9355, 2019. 2

[12] Bernard Ghanem Fabian Caba Heilbron, Victor Escorcia and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 961–970, 2015. 2

[13] Fartash Faghri, David J Fleet, Jamie Ryan Kiros, and Sanja Fidler. Vse++: Improving visual-semantic embeddings with hard negatives. *arXiv preprint arXiv:1707.05612*, 2017. 2

[14] Valentin Gabeur, Chen Sun, Karteek Alahari, and Cordelia Schmid. Multi-modal transformer for video retrieval. In *European Conference on Computer Vision (ECCV)*, volume 5. Springer, 2020. 1, 2, 3, 5, 7, 8

[15] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 297–304, 2010. 3

[16] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6546–6555, 2018. 5

[17] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020. 3

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. 12 2015. 3, 5

[19] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 5

[20] Yunseok Jang, Yale Song, Youngjae Yu, Youngjin Kim, and Gunhee Kim. Tgif-qa: Toward spatio-temporal reasoning in visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2758–2766, 2017. 2

[21] Karen Spärck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28:11–21, 1972. 4

[22] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The Kinetics Human Action Video Dataset. *arXiv:1705.06950*, 05 2017. 5

[23] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[24] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *Proceedings of the IEEE international conference on computer vision*, pages 706–715, 2017. 5, 7

[25] Kuang-Huei Lee, Xi Chen, Gang Hua, Houdong Hu, and Xiaodong He. Stacked cross attention for image-text matching. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 201–216, 2018. 2

[26] Jie Lei, Licheng Yu, Mohit Bansal, and Tamara L Berg. Tvqa: Localized, compositional video question answering. *arXiv preprint arXiv:1809.01696*, 2018. 1, 2

[27] Linjie Li, Yen-Chun Chen, Yu Cheng, Zhe Gan, Licheng Yu, and Jingjing Liu. Hero: Hierarchical encoder for video+language omni-representation pre-training. *arXiv preprint arXiv:2005.00200*, 2020. 1, 2, 3, 7

[28] Xiujun Li, Xi Yin, Chunyuan Li, Xiaowei Hu, Pengchuan Zhang, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, Yejin Choi, and Jianfeng Gao. Oscar: Object-semantics aligned pre-training for vision-language tasks. *arXiv preprint arXiv:2004.06165*, 2020. 2

[29] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 2

[30] Yang Liu, Samuel Albanie, Arsha Nagrani, and Andrew Zisserman. Use what you have: Video retrieval using representations from collaborative experts. *arXiv preprint arXiv:1907.13487*, 2019. 2, 5, 7

[31] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks. In *11 pages, 5 figures*, 08 2019. 2

[32] Huaishao Luo, Lei Ji, Botian Shi, Haoyang Huang, Nan Duan, Tianrui Li, Xilin Chen, and Ming Zhou. Univilm: A unified video and language pre-training model for multimodal understanding and generation. *arXiv:2002.06353*, 2020. 1, 2, 3, 5, 6, 7, 8

[33] A. Miech, J. B. Alayrac, L. Smaira, I. Laptev, J. Sivic, and A. Zisserman. End-to-end learning of visual representations from uncurated instructional videos. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9876–9886, June 2020. 1, 2, 3, 5, 8

[34] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. HowTo100M: Learning a Text-Video Embedding by Watching Hundred Million Narrated Video Clips. In *ICCV*, 06 2019. 1, 2, 5, 6, 7, 8

[35] Tomas Mikolov, Kai Chen, Greg S. Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *International Conference on Learning Representations*, 2013. 5

[36] Andriy Mnih and Yee Whye Teh. A fast and simple algorithm for training neural probabilistic language models. *arXiv preprint arXiv:1206.6426*, 2012. 3

[37] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training, 2018. 5

[38] Alexander Richard, Hilde Kuehne, Ahsan Iqbal, and Juergen Gall. Neuralnetwork-viterbi: A framework for weakly supervised video learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7386–7395, 2018. 8

[39] Chen Sun, Fabien Baradel, Kevin Murphy, and Cordelia Schmid. Learning video representations using contrastive bidirectional transformer. *arXiv preprint arXiv:1906.05743*, 2019. 1, 5, 8

[40] C. Sun, A. Myers, C. Vondrick, K. Murphy, and C. Schmid. Videobert: A joint model for video and language representation learning. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7463–7472, 2019. 1, 2, 5

[41] Hao Tan and Mohit Bansal. LXMERT: Learning Cross-Modality Encoder Representations from Transformers. In *EMNLP*, 08 2019. 2

[42] Yansong Tang, Dajun Ding, Yongming Rao, Yu Zheng, Danyang Zhang, Lili Zhao, Jiwen Lu, and Jie Zhou. Coin: A large-scale dataset for comprehensive instructional video analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1207–1216, 2019. 1, 2, 5

[43] Makarand Tapaswi, Yukun Zhu, Rainer Stiefelhagen, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. Movieqa: Understanding stories in movies through question-answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4631–4640, 2016. 1

[44] Atousa Torabi, Niket Tandon, and Leonid Sigal. Learning language-visual embedding for movie understanding with natural-language. *arXiv preprint arXiv:1609.08124*, 2016. 2

[45] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6450–6459, 2018. 5

[46] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. 06 2017. 3

[47] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771, 2019. 3

[48] Michael Wray, Diane Larlus, Gabriela Csurka, and Dima Damen. Fine-grained action retrieval through multiple parts-of-speech embeddings. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 450–459, 2019. 2, 7

[49] Hao Wu, Jiayuan Mao, Yufeng Zhang, Yuning Jiang, Lei Li, Weiwei Sun, and Wei-Ying Ma. Unified visual-semantic embeddings: Bridging vision and language with structured meaning representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6609–6618, 2019. 2

[50] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 305–321, 2018. 3, 5

[51] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5288–5296. IEEE, 2016. 1, 2, 5

[52] Youngjae Yu, Jongseok Kim, and Gunhee Kim. A joint sequence fusion model for video question answering and re-

trieval. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 471–487, 2018. 2, 5, 7

[53] Youngjae Yu, Hyungjin Ko, Jongwook Choi, and Gunhee Kim. Video captioning and retrieval models with semantic attention. *arXiv preprint arXiv:1610.02947*, 6(7), 2016. 2

[54] Youngjae Yu, Hyungjin Ko, Jongwook Choi, and Gunhee Kim. End-to-end concept word detection for video captioning, retrieval, and question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3165–3173, 2017. 2

[55] Bowen Zhang, Hexiang Hu, and Fei Sha. Cross-modal and hierarchical modeling of video and text. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 374–390, 2018. 2, 5, 7

[56] Luowei Zhou, Hamid Palangi, Lei Zhang, Houdong Hu, Jason J. Corso, and Jianfeng Gao. Unified vision-language pretraining for image captioning and vqa. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*, 2019. 2

[57] Luowei Zhou, Chenliang Xu, and Jason J Corso. Towards automatic learning of procedures from web instructional videos. In *AAAI 2018*, 2017. 1, 2, 3, 5

[58] Luowei Zhou, Yingbo Zhou, Jason J Corso, Richard Socher, and Caiming Xiong. End-to-end dense video captioning with masked transformer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8739–8748, 2018. 2

[59] Linchao Zhu and Yi Yang. Actbert: Learning global-local video-text representations. In *CVPR*, 2020. 1, 2, 3, 4, 5, 6, 8

[60] Dimitri Zhukov, Jean-Baptiste Alayrac, Ramazan Gokberk Cinbis, David Fouhey, Ivan Laptev, and Josef Sivic. Cross-task weakly supervised learning from instructional videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3537–3545, 2019. 1, 2, 5, 8