

Hierarchical Modular Network for Video Captioning

Hanhua Ye¹ Guorong Li^{1*} Yuankai Qi² Shuhui Wang^{3,4}
 Qingming Huang^{1,3,4} Ming-Hsuan Yang⁵

¹University of Chinese Academy of Science, Beijing, China

²Australian Institute for Machine Learning, The University of Adelaide,

³Key Lab of Intell. Info. Process., Inst. of Comput. Tech., CAS, Beijing, China,

⁴Peng Cheng Laboratory, Shenzhen, China, ⁵University of California, Merced

{yehanhua20, liguorong}@mails.ucas.ac.cn, qykshr@gmail.com, wangshuhui@ict.ac.cn,
 qmhuang@ucas.ac.cn, mhyang@ucmerced.edu

Abstract

Video captioning aims to generate natural language descriptions according to the content, where representation learning plays a crucial role. Existing methods are mainly developed within the supervised learning framework via word-by-word comparison of the generated caption against the ground-truth text without fully exploiting linguistic semantics. In this work, we propose a hierarchical modular network to bridge video representations and linguistic semantics from three levels before generating captions. In particular, the hierarchy is composed of: (I) Entity level, which highlights objects that are most likely to be mentioned in captions. (II) Predicate level, which learns the actions conditioned on highlighted objects and is supervised by the predicate in captions. (III) Sentence level, which learns the global semantic representation and is supervised by the whole caption. Each level is implemented by one module. Extensive experimental results show that the proposed method performs favorably against the state-of-the-art models on the two widely-used benchmarks: MSVD 104.0% and MSR-VTT 51.5% in CIDEr score. Code will be made available at <https://github.com/MarcusNerva/HMN>.

1. Introduction

Video captioning aims to automatically generate natural language descriptions from videos, which plays an important role in numerous applications, such as assisting visually-impaired people, human-computer interaction, and video retrieval [7, 27, 36, 42, 43, 45, 49]. Despite recent advances in this field, it remains a challenging task as a video usually contains rich and diverse content, but only some information is relevant to caption (*e.g.*, two or three out of many objects are captured in a caption).

*Corresponding author.

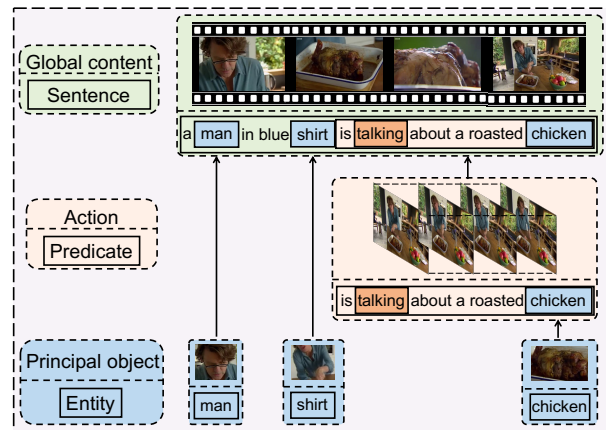


Figure 1. To effectively bridge video content and linguistic caption, we propose to supervise video representation learning in a three-level hierarchical structure, *i.e.*, the entity level, the predicate level, and the sentence level.

Existing methods aim to learn effective video representations to generate captions via recurrent decoders, which can be broadly categorized into two lines of work. The first one focuses on designing complex video encoders to learn better video representations [1, 5, 6, 26, 44, 50, 51]. For example, STG-KD [26] and ORG-TRL [51] build object relation graphs to reason the spatial and temporal relations between video objects. While GRU-EVE [1] applies Short Fourier Transform [25] to embed temporal dynamics in visual features, POS+CG [44] develops a cross gating block to fuse appearance and motion features and make a comprehensive representation. However, the optimization objectives of these methods are computed word-by-word as captions generating, disregarding the relevance between video representations and their linguistic counterpart. The other one focuses on narrowing the semantic gap between video representation and linguistic captions ahead of generating captions [28, 35, 52]. For example, Pan *et al.* [28] learn to align

the global representation of a video to the embedding of a whole caption. In contrast, Shen *et al.* [35] and Zheng *et al.* [52] associate *nouns* and *verbs* with visual features to explore the video-language correspondence on a fine-grained level. These approaches are able to generate more accurate captions as more representative video embedding is learned. However, they either focus on global sentence correspondence or local word correspondence, which disregard fine-grained details or global relevance.

In this work, we propose a hierarchical modular network to address the issues mentioned above. Our model aims to learn three kinds of video representations supervised by language semantics at different hierarchical levels as shown in Figure 1: (I) Entity level, which highlights objects that are most likely to be mentioned in captions and is supervised by *entities*¹ in the caption. (II) Predicate level, which learns the actions conditioned on highlighted objects and is supervised by the predicate in the caption. (III) Sentence level, which learns the global video representation supervised by the whole caption. Each level is implemented by one module. The motivation of our design is that objects usually serve as the cornerstone of a video caption, which can be the subject or object of an action as well as modifiers of subject and/or object. Instead of learning the visual representation of *verbs* along, we propose to learn video representation of *predicates* (*verb+noun*). This helps reduce the correspondence errors from a multi-meaning *verb* to a specific video action embedding, such as the *play* in *playing soccer* and *playing the piano*. The global video content embedding supervised by the embedding of a whole caption enables the generated caption to have a reasonable meaning.

It is worth noting that we propose a novel entity module. This module takes all the pre-extracted objects of a video as input and outputs a small set of principal objects that are most likely mentioned in a caption. Motivated by the success of DETR [3] for object detection, our entity module is designed with a transformer encoder-decoder architecture. In contrast to DETR, our queries are enhanced by video content and supervised by entities in captions, which enables the model to select principal objects according to video scenarios.

The contributions of this paper are summarized below:

- We propose a hierarchical modular framework to learn multi-level visual representations at different granularity by associating them with their linguistic counterparts: the *entity*, *predicate*, and *sentence*.
- We propose a transformer-based entity module to learn to select principal objects that are most likely to be mentioned in captions.
- Our method performs favorably against the state-

¹Note that *entities* are different from *nouns*. *Nouns* contain *abstract nouns*, such as *happiness* and *hunger*, while *entities* consist of object names, such as *onion* and *car*.

of-the-art models on two widely-used benchmarks: MSVD [4] and MSR-VTT [47].

2. Related Work

From Template-based to CNN-based Methods. There are a number of methods have been proposed for the video captioning task. Kojima *et al.* [18] and Krishnamoorthy *et al.* [20] propose to first generate words for objects and actions, and then fit predicted words into predefined sentence templates to generate captions. However, template-based methods are hard to generate flexible descriptions. Inspired by the success of RNN and CNN, the encoder-decoder structure is widely utilized to generate descriptions with flexible syntactic structure [27, 43, 45]. In [42], Venugopalan *et al.* learn video representations by performing mean-pooling over CNN features of each frame, and exploit an LSTM [14] to generate captions. On the other hand, Yao *et al.* [49] design a temporal attention mechanism to aggregate relevant video segments given the state of text-generating RNN, modeling the global temporal structure of videos. In addition to the most frequently used image and motion features, Hori *et al.* [15] and Xu *et al.* [48] exploit audio features to enrich the video representation. In [7], Chen *et al.* propose a PickNet to select informative frames from the video, removing redundant visual information. Recently, Wang *et al.* [46] and Pei *et al.* [30] enhance the captioning quality by designing a memory network to organize multiple visual features.

Utilizing Detected Objects. Objects play an important role in generating captions, which are usually extracted via pre-trained object detectors (*e.g.* YOLO9000 [31], FasterRCNN [33], Mask RCNN [13]). Significant efforts have been made to use object information of videos for captions. In [50], Zhang *et al.* utilize a GRU [8] to capture object dynamic information from temporal trajectories. Aafaq *et al.* [1] exploit object labels (predicted by object detectors) to enhance the semantics of visual representation. On the other hand, Zheng *et al.* [52] adopt an dot-product attention mechanism to help determine the interactions between objects. In addition, Pan *et al.* [26] and Zhang *et al.* [51] employ graph convolutional networks [17] to perform relational reasoning among detected objects to enhance object-level representation. These approaches can generate more accurate captions as detailed video information is mined. However, all detected objects are used in these methods. As usually only a small set of objects are mentioned in captions, the large number of redundant objects may negatively affect the captioning performance. In contrast, we propose an entity module to highlight principal objects that are most likely to be mentioned in the caption in this work, reducing the noise brought by redundant objects.

Transformer in Vision. The success of transformer models [9, 23, 40] in natural language processing has attracted

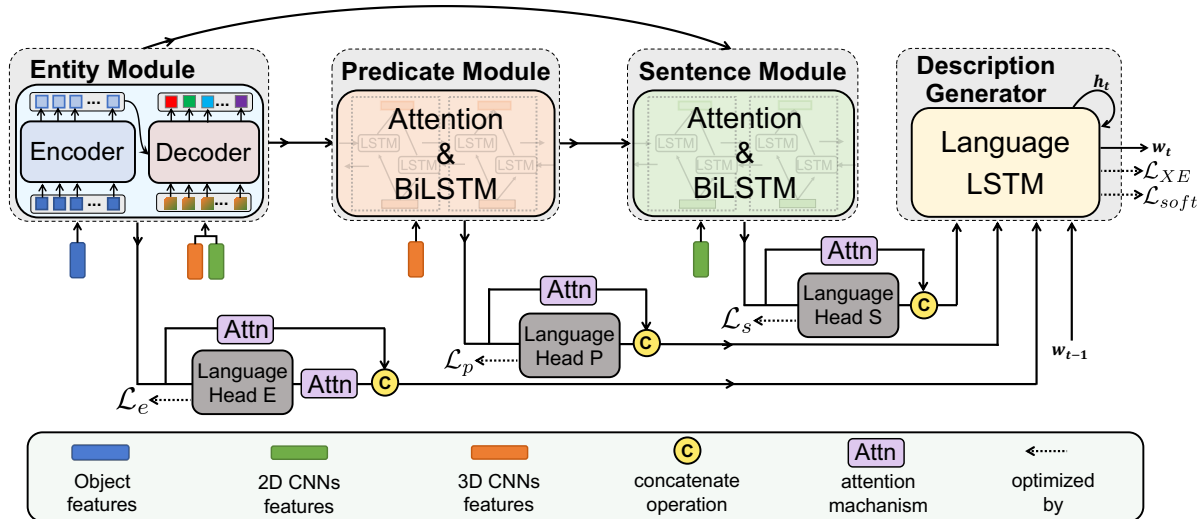


Figure 2. The proposed Hierarchical Modular Network serves as a strong video encoder, which bridges video representations and linguistic semantics from three levels via the entity (Sec. 3.1), predicate (Sec. 3.2), and sentence (Sec. 3.3) modules. Each module has its own input and linguistic supervision extracted from captions.

much interest in the computer vision community. Several methods have shown the effectiveness of transformers for vision tasks, such as image classification [10, 24, 39], object detection [3, 54], video understanding [11, 37], and semantic segmentation [53]. Motivated by DETR [3], which adaptively learn queries to detect objects, in this work we develop a transformer-based entity module to highlight principal objects out of a large number of candidates. Experiments and ablation studies show the effectiveness of this design.

3. Method

As shown in Figure 2, our model follows the conventional encoder-decoder paradigm, where our Hierarchical Modular Network (HMN) serves as the encoder. Our HMN consists of the entity, predicate, and sentence modules. Equipped with language heads, these modules are designed to bridge video representations and linguistic semantics from three levels. Our model operates as follows. First, taking all detected objects as input, the entity module outputs the features of principal objects. The predicate module encodes actions by combining features of principal objects and the video motion. Next, the sentence module encodes a global representation for the entire video content considering the global context and features of previously generated objects and actions. Finally, all features are concatenated together and fed into the decoder to generate captions.

3.1. Entity Module

Given a sequence of video frames, we uniformly select T frames as keyframes and collect short-range video frames

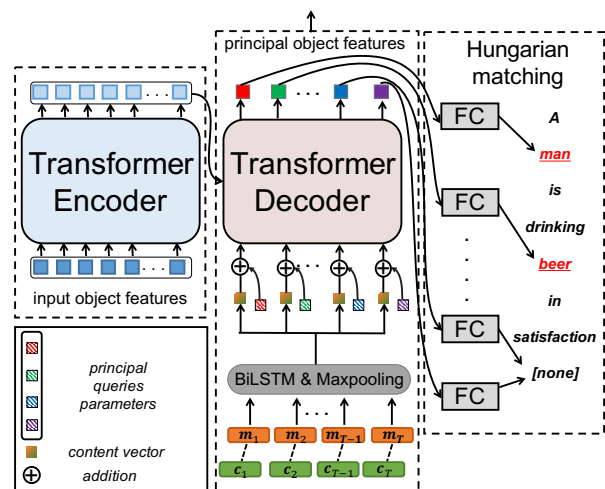


Figure 3. Main architecture of our entity module.

around keyframes as 3D cuboids. We exploit a pre-trained object detector [33] to capture object regions from each keyframe, and cluster these regions according to appearance and Intersection over Union (IoU) among bounding boxes. Then we apply the mean-pooling operation to these clusters to obtain the initial object features $\mathcal{O} = \{\mathbf{o}_i\}_{i=1}^L$, $\mathbf{o}_i \in \mathbb{R}^{d_o}$, where L and d_o denote the number of video objects and size of object features. As there are a large number of objects in a video, but only some are mentioned in captions, we design the entity module to learn to highlight these principal objects adaptively.

Architecture. Figure 3 illustrates the main architecture of our entity module, which consists of a transformer encoder and transformer decoder. This design is motivated

by DETR [3], which utilizes a transformer encoder-decoder architecture to learn a fixed set of object queries to directly predict object bounding boxes for the object detection task. Instead of simply detecting objects, we aim to determine the important ones in the video. Due to significant difference between these two tasks, a simple application of DETR architecture performs poorly (see Section 4.4). As the key of DETR is to learn queries, we design our own queries as detailed later.

The encoder maps input objects \mathcal{O} to a set of representations:

$$\mathcal{O}' = \{\mathbf{o}'_i\}_{i=1}^L = \mathbf{TransEncoder}(\mathcal{O}), \quad (1)$$

where $\mathbf{o}'_i \in \mathbb{R}^{d_{model}}$. Note that unlike conventional transformers, we discard positional encodings of objects because the object spatial location is meaningless after we apply mean-pooling on object features.

To enable the decoder to highlight principal objects, our decoder takes three types of inputs:

$$\mathcal{E} = \{\mathbf{e}_i\}_{i=1}^N = \mathbf{TransDecoder}(\mathcal{O}', \mathcal{Q}, \{\mathbf{v}\}_{\times N}), \quad (2)$$

where \mathcal{O}' is the encoder outputs, $\mathcal{Q} = \{\mathbf{q}_i\}_{i=1}^N$ is randomly initialized N parameters for queries. $\{\mathbf{v}\}_{\times N}$ are N identical video content vectors to put each query \mathbf{q}_i into a proper context and thus to facilitate decoding N video-specific principal object features. To construct the video content vector \mathbf{v} , we first utilize the pre-trained 2D CNNs [38] and pre-trained 3D CNNs [12] to extract the context features $\mathcal{C} = \{\mathbf{c}_i\}_{i=1}^T$ from keyframes, and motion features $\mathcal{M} = \{\mathbf{m}_i\}_{i=1}^T$ from 3D cuboids. Then, \mathcal{C} and \mathcal{M} are concatenated and fed into a BiLSTM to generate a set of hidden states $\mathcal{H} = \{\mathbf{h}_i\}_{i=1}^T = \mathbf{BiLSTM}(\{[\mathbf{c}_i; \mathbf{m}_i]\}_{i=1}^T)$. Next, we aggregate \mathcal{H} by max-pooling to obtain an embedding $\mathbf{v} = \mathbf{maxpool}(\mathcal{H})$ which represents video content.

As the decoder output, principal object features \mathcal{E} are later fed to the predicate and sentence modules, as well as the language head as shown in Figure 2. The language head projects \mathcal{E} to linguistic semantic space via a fully-connected layer

$$\bar{\mathcal{E}} = \{\bar{\mathbf{e}}_i\}_{i=1}^N = \mathbf{FC}(\mathcal{E}), \quad (3)$$

where $\bar{\mathbf{e}}_i \in \mathbb{R}^{d_s}$. This language head is supervised by embeddings of entities from captions as detailed in the next sub-section.

Loss function. We exploit the *entities* in captions to supervise our entity module. Specifically, we first obtain the ‘‘synonymy label’’ in *WordNet*² of each *noun* in ground-truth captions. And then, we choose *nouns* with labels ‘‘object.n.01’’, ‘‘causal_agent.n.01’’, and ‘‘matter.n.03’’ as *entities*, discarding *abstract nouns*.

Assume that M entities are extracted from a caption, we employ the pre-trained SBERT [32] as text encoder to

²<https://wordnet.princeton.edu>

compute the entities embeddings $\mathcal{N} = \{\mathbf{n}_i\}_{i=1}^M$, where $\mathbf{n}_i \in \mathbb{R}^{d_s}$. We set N larger than the typical number of *entities* in captions, and pad \mathcal{N} to size N with \emptyset (no *entity*). Then, we search for an optimal assignment $\hat{\sigma}$ between \mathcal{N} and $\bar{\mathcal{E}}$ with minimum distance cost:

$$\hat{\sigma} = \arg \min_{\sigma \in \Omega_N} \sum_i^N Dist(\mathbf{n}_i, \bar{\mathbf{e}}_{\sigma(i)}), \quad (4)$$

where the Ω_N is the assignment search space for N elements, and $Dist(\mathbf{n}_i, \bar{\mathbf{e}}_{\sigma(i)})$ is a pair-wise matching cost between the entity embedding \mathbf{n}_i and predicted linguistic embedding $\bar{\mathbf{e}}_{\sigma(i)}$ by

$$Dist(\mathbf{n}_i, \bar{\mathbf{e}}_{\sigma(i)}) = \mathbb{1}_{\{\mathbf{n}_i \neq \emptyset\}} \cdot \left(1 - \frac{\mathbf{n}_i \cdot \bar{\mathbf{e}}_{\sigma(i)}}{|\mathbf{n}_i| \cdot |\bar{\mathbf{e}}_{\sigma(i)}|}\right). \quad (5)$$

This optimal assignment is computed efficiently with Hungarian algorithm [21] as in DETR [3]. Finally, according to the optimal assignment $\hat{\sigma}$, we optimize our entity module by minimizing the distance between \mathcal{N} and $\bar{\mathcal{E}}$:

$$\mathcal{L}_e = \sum_i^N Dist(\mathbf{n}_i, \bar{\mathbf{e}}_{\hat{\sigma}(i)}). \quad (6)$$

3.2. Predicate Module

Our predicate module is designed to learn action representations that bridge the video action information and the *predicate* of the caption. As a *predicate* usually consists of a *verb* and its recipient, our predicate module concatenates the initial motion features $\mathcal{M} = \{\mathbf{m}_i\}_{i=1}^T$ and motion-related object features $\mathcal{M}^e = \{\mathbf{m}_i^e\}_{i=1}^T$ as input. For each motion \mathbf{m}_i , we compute its motion-related object features \mathbf{m}_i^e by attentively summarizing principal object features \mathcal{E} via

$$\begin{aligned} \mathbf{m}_i^e &= \sum_{k=1}^N \alpha_{i,k} \mathbf{e}_k, \\ \alpha_{i,k} &= \exp(\hat{\alpha}_{i,k}) / \sum_{j=1}^N \exp(\hat{\alpha}_{i,j}), \\ \hat{\alpha}_{i,k} &= \mathbf{w}_a^\top \tanh(\mathbf{W}_a^\top \mathbf{m}_i + \mathbf{U}_a^\top \mathbf{e}_k + \mathbf{b}_a), \end{aligned} \quad (7)$$

where $\alpha_{i,k}$ is the weight of k -th object \mathbf{e}_k regarding i -th motion; $\mathbf{W}_a, \mathbf{w}_a, \mathbf{U}_a$ and \mathbf{b}_a are learnable parameters.

Then, we use a bi-directional LSTM to encode actions:

$$\mathcal{A} = \{\mathbf{a}_i\}_{i=1}^T = \mathbf{BiLSTM}(\{[\mathbf{m}_i; \mathbf{m}_i^e]\}_{i=1}^T). \quad (8)$$

where $\mathbf{a}_i \in \mathbb{R}^{d_{model}}$. The action features \mathcal{A} are later used as inputs for the sentence module and predicate language head. Formulated as a fully-connected layer, the predicate language head projects video action to linguistic semantic space:

$$\bar{\mathbf{a}} = \mathbf{FC}(\mathbf{maxpool}(\mathcal{A})) \quad (9)$$

where $\bar{\mathbf{a}} \in \mathbb{R}^{d_s}$.

Loss function. We exploit the *predicate* of ground-truth captions to supervise our predicate module. Likewise, we use SBERT [32] to encode the *predicate* into embedding $\mathbf{p} \in \mathbb{R}^{d_s}$. Then, our predicate module is optimized by minimizing the distance between \mathbf{p} and $\bar{\mathbf{a}}$:

$$\mathcal{L}_p = 1 - \frac{\mathbf{p} \cdot \bar{\mathbf{a}}}{|\mathbf{p}| \cdot |\bar{\mathbf{a}}|}. \quad (10)$$

3.3. Sentence Module

Our sentence module is designed to learn global video representations that bridge global visual content and the entire linguistic caption. Since a caption comprises *entities*, the *predicate*, and other context information, our sentence module takes initial video context features $\mathcal{C} = \{\mathbf{c}_i\}_{i=1}^T$, context-related action features $\mathcal{C}^a = \{\mathbf{c}_i^a\}_{i=1}^T$, and context-related object features $\mathcal{C}^e = \{\mathbf{c}_i^e\}_{i=1}^T$ as input.

We obtain \mathbf{c}_i^a and \mathbf{c}_i^e by attentively summarizing action features $\mathcal{A} = \{\mathbf{a}_i\}_{i=1}^T$ and principal object features $\mathcal{E} = \{\mathbf{e}_i\}_{i=1}^N$ according to video context \mathbf{c}_i . Here we take \mathbf{c}_i^a as an example:

$$\begin{aligned} \mathbf{c}_i^a &= \sum_{k=1}^T \beta_{i,k} \mathbf{a}_k, \\ \beta_{i,k} &= \exp(\hat{\beta}_{i,k}) / \sum_{j=1}^T \exp(\hat{\beta}_{i,j}), \\ \hat{\beta}_{i,k} &= \mathbf{w}_g^\top \tanh(\mathbf{W}_g^\top \mathbf{c}_i + \mathbf{U}_g^\top \mathbf{a}_k + \mathbf{b}_g), \end{aligned} \quad (11)$$

where $\beta_{i,k}$ is the weight of the k -th action regarding the i -th context; \mathbf{w}_g , \mathbf{W}_g , \mathbf{U}_g and \mathbf{b}_g are learnable parameters.

Then, we compute the global video features:

$$\mathcal{G} = \{\mathbf{g}_i\}_{i=1}^T = \text{BiLSTM}(\{\{\mathbf{c}_i; \mathbf{c}_i^a; \mathbf{c}_i^e\}\}_{i=1}^T), \quad (12)$$

where $\mathbf{g}_i \in \mathbb{R}^{d_{model}}$. The global video feature \mathcal{G} is later used as input of the caption generator and the language head of the sentence module.

The language head of this module is implemented via a fully-connected layer. It takes \mathcal{G} as input and predicts global video linguistic embedding under supervision of the embedding of captions

$$\bar{\mathbf{g}} = \text{FC}(\text{maxpool}(\mathcal{G})), \quad (13)$$

where $\bar{\mathbf{g}} \in \mathbb{R}^{d_s}$.

Loss function. We exploit the embedding of the whole caption to supervise the sentence module. Similarly, the caption embedding $\mathbf{s} \in \mathbb{R}^{d_s}$ is computed by the pretrained SBERT. Then we minimize the distance between \mathbf{s} and $\bar{\mathbf{g}}$ to optimize our sentence module:

$$\mathcal{L}_s = 1 - \frac{\mathbf{s} \cdot \bar{\mathbf{g}}}{|\mathbf{s}| \cdot |\bar{\mathbf{g}}|}. \quad (14)$$

3.4. Description Generation

We employ an LSTM as our description generator to produce accurate captions in steps. To generate accurate descriptions for videos, our description generator takes all three levels of video representations, their linguistic predictions, and the previous word w_{t-1} as inputs

$$\mathbf{h}_t^{lang} = \text{LSTM}_{lang}([\hat{\mathbf{g}}_t^l; \hat{\mathbf{a}}_t^l; \hat{\mathbf{e}}_t^l; \mathbf{E}(w_{t-1}); \mathbf{h}_{t-1}^{lang}), \quad (15)$$

where $\mathbf{E}(w_{t-1}) \in \mathbb{R}^{d_w}$ is the word embedding of w_{t-1} , and d_w is the embedding size. $\hat{\mathbf{g}}_t^l$, $\hat{\mathbf{a}}_t^l$, and $\hat{\mathbf{e}}_t^l$ are concatenation of visual features and linguistic predictions, *i.e.*, $\hat{\mathbf{g}}_t^l = [\mathbf{g}_t^l; \bar{\mathbf{g}}]$, $\hat{\mathbf{a}}_t^l = [\mathbf{a}_t^l; \bar{\mathbf{a}}]$, $\hat{\mathbf{e}}_t^l = [\mathbf{e}_t^l; \bar{\mathbf{e}}]$. $\bar{\mathbf{g}}$ and $\bar{\mathbf{a}}$ are predicted linguistic embeddings of global content and actions as described in Section 3.3 and Section 3.2. \mathbf{g}_t^l , \mathbf{a}_t^l , \mathbf{e}_t^l , and $\bar{\mathbf{e}}_t^l$ are obtained via attentively summarizing \mathcal{G} , \mathcal{A} , \mathcal{E} , and entity linguistic prediction $\bar{\mathcal{E}}$ according to the history hidden state \mathbf{h}_{t-1}^{lang} . These four representations are computed in a similar way. Here we take \mathbf{e}_t^l as an example:

$$\begin{aligned} \mathbf{e}_t^l &= \sum_{k=1}^N \gamma_{t,k} \mathbf{e}_k, \\ \gamma_{t,k} &= \exp(\hat{\gamma}_{t,k}) / \sum_{j=1}^N \exp(\hat{\gamma}_{t,j}), \\ \hat{\gamma}_{t,k} &= \mathbf{w}_d^\top \tanh(\mathbf{W}_d^\top \mathbf{h}_{t-1}^{lang} + \mathbf{U}_d^\top \mathbf{e}_k + \mathbf{b}_d), \end{aligned} \quad (16)$$

where $\gamma_{t,k}$ is the weight of object \mathbf{e}_k at the t -th decoding step; and \mathbf{w}_d , \mathbf{W}_d , \mathbf{U}_d and \mathbf{b}_d are learnable parameters.

According to the current hidden state of LSTM \mathbf{h}_t^{lang} , the probability distribution \mathbf{P}_t over a vocabulary of D words is computed by a fully-connected layer and the softmax operation

$$\mathbf{P}_t = \text{softmax}(\text{FC}(\mathbf{h}_t^{lang})). \quad (17)$$

Loss function. Given a video with the ground-truth words $[w_1, \dots, w_{L_s}]$, where L_s is the caption length, we compute the Cross-Entropy loss to optimize our description generator:

$$\mathcal{L}_{XE} = - \sum_{t=1}^{L_s} \delta(w_t)^\top \log \mathbf{P}_t, \quad (18)$$

where $\delta(w_t) \in \mathbb{R}^D$ is the one-hot encoding of word w_t .

However, the long-tailed word distribution problem³ in the video captioning corpus likely causes training issues with unbalanced data. To alleviate this problem, Zhang *et al.* [51] propose a plug-in method, where the abundant linguistic knowledge is transferred from a pretrained external language model (ELM) to the description generator. Given

³The function words and common words (such as ‘a’, ‘the’, and ‘man’) are far more than the content specific words (such as ‘lion’, ‘onion’, and ‘bicycle’) in the corpus.

the previous words $w_{<t}$, the probability distribution of ELM at time step t is:

$$Q_t = \text{ELM}(w_{<t} | \Theta_{\text{ELM}}). \quad (19)$$

To transfer the knowledge from ELM to the description generator, the KL divergence between P_t and Q_t is minimized during training:

$$D_{KL} = - \sum_{d=1}^D Q_t^d \cdot \log \frac{P_t^d}{Q_t^d}, \quad (20)$$

where Q_t^d and P_t^d denote the d -th element of Q_t and P_t . In this work, we treat Q_t as ‘‘soft target’’ of our description generator, and use D_{KL} as \mathcal{L}_{soft} . More details about the ‘‘ELM’’ and ‘‘soft-target’’ can be found in [51].

3.5. Training

Our model is trained in an end-to-end fashion via optimizing the sum of all losses

$$\mathcal{L} = \lambda_e \mathcal{L}_e + \lambda_p \mathcal{L}_p + \lambda_s \mathcal{L}_s + \mathcal{L}_{XE} + \lambda_{soft} \mathcal{L}_{soft}. \quad (21)$$

4. Experimental Results

In this section, we evaluate our model on two widely-used datasets: MSVD [4] and MSR-VTT [47], with four widely-used metrics: BLEU@4 [29], METEOR [2], ROUGE-L [22], and CIDEr [41] (denoted by B@4, M, R, and C, respectively). We present the results compared to state-of-the-art methods and report ablation studies on both benchmarks.

4.1. Datasets

MSVD is a widely-used video captioning dataset composed of 1970 short video clips collected from YouTube. Each clip is annotated with up to 40 English sentences. Similar to existing methods [6,26,51,52], we take 1200 video clips for training, 100 clips for validation, and 670 clips for testing. **MSR-VTT** is a large-scale dataset for video captioning. It is collected from a commercial video website and covers the most diverse visual contents so far. MSR-VTT contains 10k video clips from 20 categories, such as music, people, and gaming. Each video clip has 20 English captions, leading to total 28k unique words. We use the same setup as existing methods [6,26,51,52] for experiments (6513, 497, and 2990 videos for training, validation and testing).

4.2. Implementation Details

Feature Extraction and Text Processing. We uniformly sample $T = 15$ clips for each video. Each clip consists of 16 consecutive frames. We take the 8-th frame of each clip as keyframe and detected 10 objects for each keyframe. Similar to most recent methods [1,5,6,51,52], we

extract context features using the InceptionResNetV2 [38] model and motion features using the C3D [12] model. In addition, we use Faster-RCNN [33] pretrained on Visual Genome [19] to detect object regions. All the above-mentioned visual features are projected to a 512-dimension space before being fed into our model. Likewise, the dimension of encoded visual features d_{model} is set to 512.

For captions, we first remove punctuations and convert all letters into lowercase. Captions are truncated at 20 words and tokenized. We exploit off-the-shelf Constituency Parsing tools provided by *AllenNLP*⁴ to extract *nouns* and the *predicate* from the processed caption, where we utilize the *WordNet* to distinguish *entities* from *nouns* as described in Section 3.1. The embedding size d_s of *entities*, *predicates*, and *sentences* is set to 768 using the off-the-shelf pre-trained SBERT [32]. The size of word embedding d_w is set to 300.

Other details. We set the number of queries N in Eq.(2) to 8 for our entity module. Both the encoder and decoder of the entity module have 2 transformer layers for MSVD (3 for MSR-VTT), 8 attention heads, and 512 hidden state size. Empirically, λ_e , λ_p , λ_s , and λ_{soft} are set to 0.6, 0.3, 1.0, and 0.5 respectively. We adopt the Adam [16] optimizer with a learning rate of 1e-4 to optimize our model. The batch size is set to 64, and the training epochs is set to 20. The hidden size of description generator is set to 512. During testing, we use beam search with size 5 to generate captions. Our entire system is implemented with PyTorch and all experiments are conducted on $1 \times$ RTX-2080Ti GPU.

4.3. Comparison to State-of-The-Art Methods

We evaluate the proposed model against the state-of-the-art methods. The main results are presented in Table 1. Our method achieves the best performance on the MSVD benchmark in terms of all the four metrics. In particular, our method achieves 59.2 of BLUE@4 (with +2.3 improvement) and 104.0 of CIDEr (with +5.5 improvement). On the MSR-VTT benchmark, our method achieves the best results on 3 out of 4 metrics. Note that CIDEr captures human judgment of consensus better than other metrics [41]. The significant improvements under CIDEr on both benchmarks demonstrate that our model can generate semantically more accurate captions.

Note that GRU-EVE, STG-KD and ORG-TRL focus on designing complex video encoders to learn video representations, but leave the video-language relevance unexploited. SAAT associates the *subject*, *verb*, and *object* with visual information, focusing on local word correspondence. However, our model outperforms them by a significant margin (+8.8~25.9 of CIDEr on MSVD) via jointly modeling the video-language correspondence at three different levels,

⁴<https://demo.allennlp.org/constituency-parsing>

Models	Year	Context	Features Motion	Object	MSVD				MSR-VTT			
					B@4	M	R	C	B@4	M	R	C
M ³ [46]	2018	VGG	C3D	-	51.8	32.5	-	-	38.1	26.6	-	-
RecNet [45]	2018	Inception-V4	-	-	52.3	34.1	69.8	80.3	39.1	26.6	59.3	42.7
PickNet [7]	2018	ResNet-152	-	-	52.3	33.3	69.6	76.5	41.3	27.7	59.8	44.1
MARN [30]	2019	ResNet-101	C3D	-	48.6	35.1	71.9	92.2	40.4	28.1	60.7	47.1
OA-BTG [50]	2019	ResNet-200	-	Mask-RCNN	56.9	36.2	-	90.6	41.4	28.2	-	46.9
POS-CG [44]	2019	InceptionResnetV2	OpticalFlow	-	52.5	34.1	71.3	88.7	42.0	28.2	61.6	48.7
MGSA [5]	2019	InceptionResnetV2	C3D	-	53.4	35.0	-	86.7	42.4	27.6	-	47.5
GRU-EVE [1]	2019	InceptionResnetV2	C3D	YOLO	47.9	35.0	71.5	78.1	38.3	28.4	60.7	48.1
STG-KD [26]	2020	ResNet-101	I3D	Faster-RCNN	52.2	36.9	73.9	93.0	40.5	28.3	60.9	47.1
SAAT [52]	2020	InceptionResnetV2	C3D	Faster-RCNN	46.5	33.5	69.4	81.0	40.5	28.2	60.9	49.1
ORG-TRL [51]	2020	InceptionResnetV2	C3D	Faster-RCNN	54.3	36.4	73.9	95.2	43.6	28.8	62.1	50.9
SGN [34]	2021	ResNet-101	C3D	-	52.8	35.5	72.9	94.3	40.8	28.3	60.8	49.5
MGRMP [6]	2021	InceptionResnetV2	C3D	-	55.8	36.9	74.5	98.5	41.7	28.9	62.1	51.4
HMN (ours)	2022	InceptionResnetV2	C3D	Faster-RCNN	59.2	37.7	75.1	104.0	43.5	29.0	62.7	51.5

Table 1. Comparison with state-of-the-art methods on MSVD and MSR-VTT benchmarks. The best results are shown in bold.

which indicates the advantage of further exploring the corresponding relationship between the video and sentence.

4.4. Ablation Study

Effectiveness of Proposed Modules. We evaluate the effectiveness of each module by removing them and retraining the model. The results are shown in the first group of Table 2. All the proposed modules contribute significantly to the overall performance. The entity module contributes the most, followed by the sentence and predicate modules.

Effectiveness of Links Between Modules. In our model, we adapt side connections between modules. To evaluate their effectiveness, we cut off such side connections separately. We use “cut e2p” to denote removing the connection between the entity module and the predicate module, “cut e2s” for removing the connection between the entity module and the sentence module, and “cut p2s” for removing the connection between the predicate module and the sentence module.

The results are presented in the second group of Table 2. The performance drops without these side connections, which indicates their contribution to the overall performance. Compared to the performance when removing each module completely, the performance is relatively better. This indicates that the three modules themselves contribute more than the connection design between them.

Noun. v.s. Entity and Verb. v.s. Predicate During the training, we supervise the entity and predicate modules with *entities* and the *predicate*, respectively. Here we analyze the alternatives by replacing the entity supervision with broader *noun* supervision, and replacing predicate with *verb* supervision. The results are presented in the third group of Table 2. The alternative approaches perform worse than our final model, which indicates *entities* and the *predicate* provide better supervisions for video captioning.

Models	MSVD				MSR-VTT			
	B@4	M	R	C	B@4	M	R	C
w/o Entity	51.5	34.4	71.8	88.3	40.9	27.3	60.6	46.6
w/o Predicate	56.1	36.6	73.7	98.9	43.4	28.0	61.8	49.8
w/o Sentence	51.8	34.8	72.2	94.8	41.5	27.5	61.0	49.4
Cut e2p	56.9	36.7	73.9	101.3	43.3	28.0	61.7	50.4
Cut e2s	56.7	36.4	73.3	100.6	43.1	28.3	61.9	50.8
Cut p2s	56.5	37.0	73.4	100.6	42.5	28.0	61.7	50.7
noun supervision	53.2	35.8	72.3	98.1	42.8	28.1	61.7	50.4
verb supervision	55.7	36.8	73.7	102.1	42.4	28.2	61.6	50.1
all objects	53.6	35.9	72.8	92.7	42.6	28.0	61.4	48.6
w/o v.c.q.	53.6	35.7	72.8	94.4	41.8	27.8	61.1	48.5
Full model	59.2	37.7	75.1	104.0	43.5	29.0	62.7	51.5

Table 2. Ablation studies on MSVD and MSR-VTT.

Principal Objects v.s. All Objects To evaluate the effectiveness of selecting principal objects, we replace our entity module with original embeddings of all objects. As row “all objects” of Table 2 shows, the model performance drops significantly. This is likely caused by the fact that many objects are redundant for video captioning.

Video Content Enhanced Object Query In our entity module, we enhance object queries with video content vectors. To verify its effectiveness, we carry out experiments by removing these vectors. The results are presented in the “w/o v.c.q.” row of Table 2. The model performance drops significantly without video content for queries.

Number of Queries The number of queries N can significantly affect captioning performance as shown in Figure 5. Too many or Too few principal object queries can degrade the performance. This is because too many queries may introduce noisy objects and too few queries cannot provide sufficient information.

4.5. Qualitative Results

We show some qualitative results in Figure 4. The first row shows attended objects by our entity module. A clearer

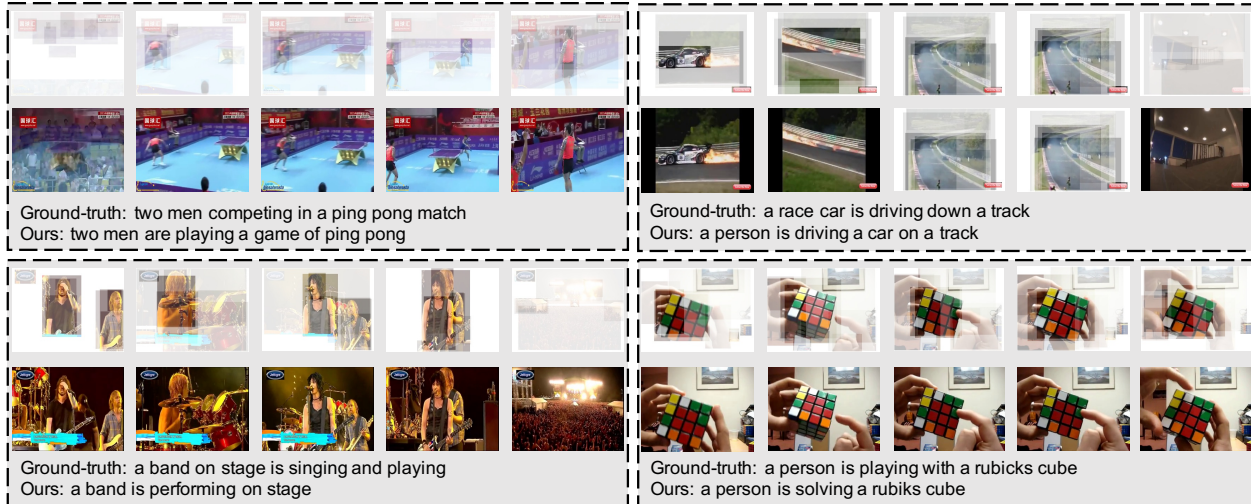


Figure 4. Qualitative results on MSR-VTT. The first row is the principal object regions attended by the our entity module. A clearer region denotes a higher attention weights. The second row shows keyframes for each video.

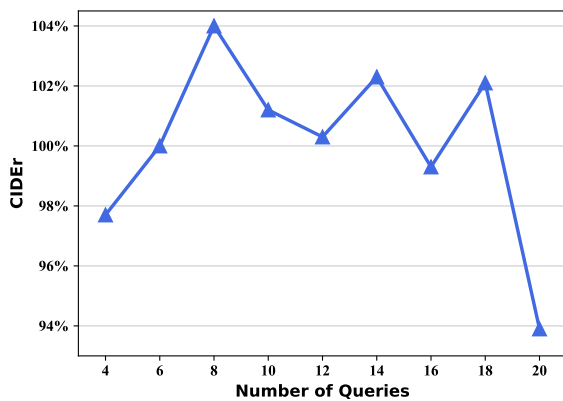


Figure 5. Performance (CIEDr score) comparison under different number of queries.

region denotes a higher attention weight. The second row shows the keyframes for reference. As shown in the figure, our model can generate high-quality captions. The regions of principal objects can be chosen correctly most of the time, which shows the entity module has the ability to distinguish principal objects from all the detected objects. We also note that the entity module is able to neglect some redundant frames. For instance, the example at the top-left shows that our entity module mainly focuses on the two ping-pong players rather than those audiences in the first frame. Similarly, in examples at the top-right and bottom-left, our entity module ignores the objects that appeared in the last frames because their content is not closely related to the video caption. Due to space limitation, more qualitative results are placed in supplementary materials.

4.6. Limitations

Similar to existing methods [28, 34, 52], our approach is suitable for single-action videos. However, there exist many multi-action videos in the MSR-VTT dataset. For instance, *a man walks in a room and talks to others* has two separate actions, i.e., *walks in a room* and *talks to others*. Existing methods and the proposed model are not able to handle these complex scenes that require multiple predicates. This is also the main reason why the improvement of our model on MSR-VTT is not as significant as that on MSVD.

5. Conclusion

In this paper, we proposed a hierarchical modular network for video captioning, which bridges video representations and linguistic semantics from three levels: entity level, predicate level, and sentence level. By constructing three modules and associating them with their linguistic counterparts, we can obtain effective visual representations for captioning. Furthermore, our entity module is able to identify principal video objects that carry important semantic meanings without processing all mundane ones for generating accurate captions. Our proposed model has achieved the state-of-the-art performance on both MSVD and MSR-VTT benchmarks. Extensive experiments and qualitative results have demonstrated the effectiveness of each module.

Acknowledgements: This work was supported in part by the Italy–China Collaboration Project TALENT under Grant 2018YFE0118400; in part by the National Natural Science Foundation of China under Grant 61836002, 61902092, 61976069, 61872333, 62022083, and 61931008; in part by the Youth Innovation Promotion Association CAS; in part by the Fundamental Research Funds for Central Universities. M.-H. Yang is supported in part by NSF CAREER grant 1149783.

References

- [1] Nayyer Aafaq, Naveed Akhtar, Wei Liu, Syed Zulqarnain Gilani, and Ajmal Mian. Spatio-temporal dynamics and semantic attribute enriched visual encoding for video captioning. In *CVPR*, pages 12487–12496, 2019. 1, 2, 6, 7
- [2] Satanjeev Banerjee and Alon Lavie. METEOR: an automatic metric for MT evaluation with improved correlation with human judgments. In *ACL Workshops*, pages 65–72, 2005. 6
- [3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, pages 213–229, 2020. 2, 3, 4
- [4] David L. Chen and William B. Dolan. Collecting highly parallel data for paraphrase evaluation. In *ACL*, pages 190–200, 2011. 2, 6
- [5] Shaoxiang Chen and Yu-Gang Jiang. Motion guided spatial attention for video captioning. In *AAAI*, pages 8191–8198, 2019. 1, 6, 7
- [6] Shaoxiang Chen and Yu-Gang Jiang. Motion guided region message passing for video captioning. In *ICCV*, pages 1543–1552, 2021. 1, 6, 7
- [7] Yangyu Chen, Shuhui Wang, Weigang Zhang, and Qingming Huang. Less is more: Picking informative frames for video captioning. In *ECCV*, pages 367–384, 2018. 1, 2, 7
- [8] Kyunghyun Cho, Bart van Merriënboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *EMNLP*, pages 1724–1734, 2014. 2
- [9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, pages 4171–4186, 2019. 2
- [10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 3
- [11] Rohit Girdhar, João Carreira, Carl Doersch, and Andrew Zisserman. Video action transformer network. In *CVPR*, pages 244–253, 2019. 3
- [12] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *CVPR*, pages 6546–6555, 2018. 4, 6
- [13] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. In *ICCV*, pages 2980–2988, 2017. 2
- [14] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, 1997. 2
- [15] Chiori Hori, Takaaki Hori, Teng-Yok Lee, Ziming Zhang, Bret Harsham, John R. Hershey, Tim K. Marks, and Kazuhiko Sumi. Attention-based multimodal fusion for video description. In *ICCV*, pages 4203–4212, 2017. 2
- [16] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 6
- [17] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017. 2
- [18] Atsuhiko Kojima, Takeshi Tamura, and Kunio Fukunaga. Natural language description of human activities from video images based on concept hierarchy of actions. *IJCV*, 50:171–184, 2002. 2
- [19] 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. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *IJCV*, 123:32–73, 2017. 6
- [20] Niveda Krishnamoorthy, Girish Malkarnkar, Raymond J. Mooney, Kate Saenko, and Sergio Guadarrama. Generating natural-language video descriptions using text-mined knowledge. In *AAAI*, 2013. 2
- [21] Harold W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2:83–97, 1955. 4
- [22] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004. 6
- [23] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019. 2
- [24] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, pages 10012–10022, 2021. 3
- [25] Alan Oppenheim and Ronald Schaffer. *Discrete-time Signal Processing*. Prentice-Hall, 1999. 1
- [26] Boxiao Pan, Haoye Cai, De-An Huang, Kuan-Hui Lee, Adrien Gaidon, Ehsan Adeli, and Juan Carlos Niebles. Spatio-temporal graph for video captioning with knowledge distillation. In *CVPR*, pages 10867–10876, 2020. 1, 2, 6, 7
- [27] Pingbo Pan, Zhongwen Xu, Yi Yang, Fei Wu, and Yueting Zhuang. Hierarchical recurrent neural encoder for video representation with application to captioning. In *CVPR*, pages 1029–1038, 2016. 1, 2
- [28] Yingwei Pan, Tao Mei, Ting Yao, Houqiang Li, and Yong Rui. Jointly modeling embedding and translation to bridge video and language. In *CVPR*, pages 4594–4602, 2016. 1, 8
- [29] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, pages 311–318, 2002. 6
- [30] Wenjie Pei, Jiyuan Zhang, Xiangrong Wang, Lei Ke, Xiaoyong Shen, and Yu-Wing Tai. Memory-attended recurrent network for video captioning. In *CVPR*, pages 8347–8356, 2019. 2, 7
- [31] Joseph Redmon and Ali Farhadi. YOLO9000: better, faster, stronger. In *CVPR*, pages 6517–6525, 2017. 2
- [32] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *EMNLP-IJCNLP*, pages 3980–3990, 2019. 4, 5, 6

- [33] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. In *NeurIPS*, pages 91–99, 2015. [2](#), [3](#), [6](#)
- [34] Hobin Ryu, Sunghun Kang, Haeyong Kang, and Chang D. Yoo. Semantic grouping network for video captioning. In *AAAI*, pages 2514–2522, 2021. [7](#), [8](#)
- [35] Zhiqiang Shen, Jianguo Li, Zhou Su, Minjun Li, Yurong Chen, Yu-Gang Jiang, and Xiangyang Xue. Weakly supervised dense video captioning. In *CVPR*, pages 5159–5167, 2017. [1](#), [2](#)
- [36] Jingkuan Song, Lianli Gao, Zhao Guo, Wu Liu, Dongxiang Zhang, and Heng Tao Shen. Hierarchical LSTM with adjusted temporal attention for video captioning. In *IJCAI*, pages 2737–2743, 2017. [1](#)
- [37] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning. In *ICCV*, pages 7463–7472, 2019. [3](#)
- [38] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, pages 4278–4284, 2017. [4](#), [6](#)
- [39] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*, pages 10347–10357, 2021. [3](#)
- [40] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, pages 5998–6008, 2017. [2](#)
- [41] Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *CVPR*, pages 4566–4575, 2015. [6](#)
- [42] Subhashini Venugopalan, Marcus Rohrbach, Jeffrey Donahue, Raymond J. Mooney, Trevor Darrell, and Kate Saenko. Sequence to sequence - video to text. In *ICCV*, pages 4534–4542, 2015. [1](#), [2](#)
- [43] Subhashini Venugopalan, Huijuan Xu, Jeff Donahue, Marcus Rohrbach, Raymond J. Mooney, and Kate Saenko. Translating videos to natural language using deep recurrent neural networks. In *NAACL*, pages 1494–1504, 2015. [1](#), [2](#)
- [44] Bairui Wang, Lin Ma, Wei Zhang, Wenhao Jiang, Jingwen Wang, and Wei Liu. Controllable video captioning with POS sequence guidance based on gated fusion network. In *ICCV*, pages 2641–2650, 2019. [1](#), [7](#)
- [45] Bairui Wang, Lin Ma, Wei Zhang, and Wei Liu. Reconstruction network for video captioning. In *CVPR*, pages 7622–7631, 2018. [1](#), [2](#), [7](#)
- [46] Junbo Wang, Wei Wang, Yan Huang, Liang Wang, and Tieniu Tan. M3: multimodal memory modelling for video captioning. In *CVPR*, pages 7512–7520, 2018. [2](#), [7](#)
- [47] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. MSR-VTT: A large video description dataset for bridging video and language. In *CVPR*, pages 5288–5296, 2016. [2](#), [6](#)
- [48] Jun Xu, Ting Yao, Yongdong Zhang, and Tao Mei. Learning multimodal attention LSTM networks for video captioning. In *ACMMM*, pages 537–545, 2017. [2](#)
- [49] Li Yao, Atousa Torabi, Kyunghyun Cho, Nicolas Ballas, Christopher J. Pal, Hugo Larochelle, and Aaron C. Courville. Describing videos by exploiting temporal structure. In *ICCV*, pages 4507–4515, 2015. [1](#), [2](#)
- [50] Junchao Zhang and Yuxin Peng. Object-aware aggregation with bidirectional temporal graph for video captioning. In *CVPR*, pages 8327–8336, 2019. [1](#), [2](#), [7](#)
- [51] Ziqi Zhang, Yaya Shi, Chunfeng Yuan, Bing Li, Peijin Wang, Weiming Hu, and Zheng-Jun Zha. Object relational graph with teacher-recommended learning for video captioning. In *CVPR*, pages 13275–13285, 2020. [1](#), [2](#), [5](#), [6](#), [7](#)
- [52] Qi Zheng, Chaoyue Wang, and Dacheng Tao. Syntax-aware action targeting for video captioning. In *CVPR*, pages 13093–13102, 2020. [1](#), [2](#), [6](#), [7](#), [8](#)
- [53] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip H. S. Torr, and Li Zhang. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *CVPR*, pages 6881–6890, 2021. [3](#)
- [54] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable DETR: deformable transformers for end-to-end object detection. In *ICLR*, 2021. [3](#)