Improving Video Captioning with Temporal Composition of a Visual-Syntactic Embedding *

Jesus Perez-Martin, Benjamin Bustos and Jorge Pérez
Department of Computer Science, University of Chile
{jeperez, jperez, bebustos}@dcc.uchile.cl

Abstract

Video captioning is the task of predicting a semantic and syntactically correct sequence of words given some context video. The most successful methods for video captioning have a strong dependency on the effectiveness of semantic representations learned from visual models, but often produce syntactically incorrect sentences which harms their performance on standard datasets. In this paper, we address this limitation by considering syntactic representation learning as an essential component of video captioning. We construct a visual-syntactic embedding *by mapping into a common vector space a visual representation, that depends only on the video, with a syntactic representation that depends only on Part-of-Speech (POS) tagging structures of the video description. We integrate this joint representation into an encoder-decoder architecture that we call Visual-Semantic-Syntactic Aligned Network (SemSynAN), which guides the decoder (text generation stage) by aligning temporal compositions of visual, semantic, and syntactic representations. We tested our proposed architecture obtaining state-of-the-art results on two widely used video captioning datasets: the Microsoft Video Description (MSVD) dataset and the Microsoft Research Video-to-Text (MSR-VTT) dataset.*

1. Introduction

A way of bridging vision and language is the automatic generation of natural language descriptions of videos, also known as video captioning [3, 17, 19, 24, 25, 35, 42, 45, 60]. This topic represents a fundamental challenge for several research areas like video analysis and understanding, human-computer interaction, and deep learning applications for vision. As a text generation task, video captioning is substantially more difficult than predicting a single sentence from an image (image captioning) [5, 9, 16, 27, 28, 32, 37, 53] since spatial-temporal information in videos introduces diversity and complexity regarding the visual content and the structure of associated textual descriptions.

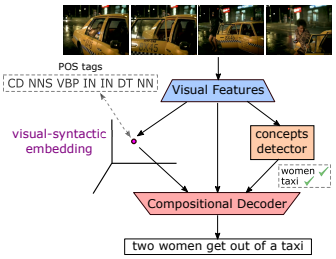

Figure 1. Example of video caption generation with Visual-Syntactic Embedding. The method computes high-level semantic and syntactic representations from the visual representation of the video. Next, the decoder generates a sentence from them.

The application of Deep Learning in both computer vision and natural language processing is gaining popularity due to its success in tasks like action recognition [15, 26, 30, 34] and machine translation [12, 52]. Specifically for video captioning, the state-of-the-art methods are based on *encoder-decoder* deep learning architectures. The encoder part of such architectures, usually based on Convolutional Neural Networks (CNN), compresses the video into feature representations, *e.g.*, appearance features, motion features, and high-level semantic representations. The decoder, usually based on Recurrent Neural Networks (RNN), generates the caption one word at a time given the encoder representations. Most successful architectures are focused on generating the correct words and concepts in the output description by including semantic features, but usually neglects the syntactic structure of the sentences describing the videos.

In this paper, we propose an encoder-decoder model (see Figure 1) that, besides considering visual and semantic features, incorporates in the decoder phase, a visual-syntactic representation extracted from the input video. The three types of representations (visual, semantic and syntactic representations) are combined with what we call *var-norm-compositional LSTM* and *adaptive fusion gates* that decide

*This work was funded by ANID - Millennium Science Initiative Program - Code ICN17_002 and ANID/Doctorado Nacional/2018-21180648.

when and how to include each feature type in the token generation phase. Specifically, the main contributions of this paper are as follows:

1. We propose a model to create *visual-syntactic embeddings* by exploiting the Part-of-Speech (POS) templates of video descriptions. We do this by learning two functions: $\phi(\cdot)$ that maps videos, and $\omega(\cdot)$ that maps (POS tags of) captions, both into a common vector space. The learning process is based on a *match and rank* strategy, and ensures that videos and their corresponding captions are mapped close together in the common space. Then, when producing features for the decoder architecture (see the next point), we can use the function $\phi(\cdot)$ to map the input video and generate our desired visual-syntactic embedding. To the best of our knowledge, this is the first approach to jointly learn embeddings from videos and (POS tags of) descriptions. Moreover, our proposal constitutes the first instance of an effective use of a ranking model to obtain syntactic representations of videos.

2. We propose the Visual-Semantic-Syntactic Aligned Network (SemSynAN) for video captioning that integrates global semantic and syntactic representations of the input video. It learns how to combine visual, semantic, and syntactic information in pairs (*i.e*., visual-semantic, visual-syntactic, and semantic-syntactic) while generating output tokens. As our results show, this process produces more accurate descriptions, both semantically and syntactically.

3. We evaluate our method on two widely used datasets: the Microsoft Video Description (MSVD) dataset [33] and the Microsoft Research Video-to-Text (MSR-VTT) dataset [56]. Except for one metric in MSR-VTT, we improve the state-of-the-art in both datasets in all metrics. For instance, in MSVD we obtain a relative improvement of 10.8% for METEOR and 8.2% for CIDEr, and in MSR-VTT a relative improvement of 2.6% for BLEU-4 and of 1.7% for METEOR.

2. Related Work

Previous work on video captioning employed template-based models [21, 29, 31, 48, 50, 57, 58], which aim to generate sentences within a reduced set of templates that assure grammatical correctness. These templates produce sentences organizing the results of the first stage of recognition of the relevant visual content (Subject-Verb-Object (SVO) triplets). However, the required complexity of rules and templates makes their manual design a time-consuming and highly expensive task for any sufficiently rich domain. Hence, these approaches soon become unsuitable for dealing with open-domain datasets.

2.1. Joint Embeddings And Semantic Guiding

For tasks like *video retrieval from descriptions* and *video descriptions retrieval from videos* [10, 11, 14, 38, 39], the joint visual-semantic embeddings have a successful application. These embeddings are constructed by combining two models: a *language model* that maps the captions to a language representation vector, and a *visual model* that obtains a visual representation vector from visual features. Both models are trained for projecting those representations into a joint space, minimizing a distance function. Dong *et al*. [11] obtain high-performance in retrieval tasks by using the same multi-level architecture for both models, and training with the *triplet-ranking-loss* function [13].

For video captioning, these embeddings have not been widely explored [18, 35, 42]. In LSTM-E [42], a joint embedding component is utilized to bridge the gap between visual content and sentence semantics. This embedding is trained by minimizing the *relevance loss* and *coherence loss* simultaneously. In SibNet [35], autoencoder for visual information, and a visual-semantic embedding for semantic information are exploited. These joint embeddings only consider the implicit contextual information of word vectors. To improve the perplexity and syntax correctness of generated sentences, we learn a new representation of videos with suitable syntactic information. We propose a joint visual-syntactic embedding, trained for retrieving POS tagging sequences from videos. We employ it to produce syntactic features and alleviate the syntactic inconsistency between the video content and the generated caption.

Another way of exploiting informative semantics is by learning to ensemble the result of visual perception models [3, 17, 36, 43, 59]. Pan *et al*. [43] incorporated the transferred semantic attributes learned from two sources (images and videos) inside a CNN-RNN framework. For this, they integrated a *transfer unit* to dynamically control the impact of each source's semantic attributes as an additional input to LSTM. Gan *et al*. [17] included the semantic meaning via a *semantic-concept-detector* model, which computes the probability of each concept appears in the video. They incorporated concept-dependent weight tensors in LSTM for composing the semantic representations. More recently, Yuan *et al*. [59] proposed the *Semantic Guiding LSTM*, which jointly explores visual and semantic features using two semantic guiding layers. Chen *et al*. [3] improved the model proposed by Gan *et al*. [17] by using the hyperbolic tangent $\tanh(\cdot)$ to activate the raw cell input instead of the logistic sigmoid $\sigma(\cdot)$, and integrating the global semantic feature at each step instead of only at the beginning.

These methods show the benefits of describing the videos according to dynamic visual and semantic information. However, these models' performance has a strong dependence on the quality of semantic concept detection models. This strong dependence can be alleviated by in-

cluding an adaptive mechanism that selectively determines the visual and semantic information required to generate each word. In this sense, we combine two *fusion gates* for adaptively mix three aligned temporal compositions of three sources: visual, semantic, and syntactic.

2.2. Syntactic Guiding

Although some recent works in image captioning [8, 23] and video captioning [25, 54] have explored the use of syntactic information in the generation process, its impact has not been widely explored. For the video captioning task, Hou *et al*. [25] created a model that first generates a sequence of POS tags and a sequence of words, and then learns the joint probability of both sequences using a probabilistic directed acyclic graph. Wang *et al*. [54] integrated a syntactic representation in the decoder, also learned by a POS sequence generator. A weakness of these models is that they do not directly exploit the relationship neither between syntactic and visual representations nor between syntactic and semantic representations. In contrast, we propose a method to learn a visual-syntactic embedding and obtain syntactic representations of videos, instead of learning to generate a sequence of POS tags. We also include two compositional layers to obtain visual-syntactic-related and semantic-syntactic-related representations.

3. Our Approach

As a learning problem, video captioning can be formalized as follows. Given a set $\mathcal{D}$ of pairs (x, y), where $x = (x_1, x_2, \ldots, x_n)$ is a sequence representing frames from a video, and $y = (w_1, w_2, \ldots, w_m)$ is a sequence of words that describes the information in x, we want to construct a model that maximizes

$$p(w_1, w_2, \ldots, w_m | x_1, x_2, \ldots, x_n)$$

over all pairs $(x, y) \in \mathcal{D}$. That is, the model maximizes the probability of the descriptive information (y) given the input video (x). Usually, in video caption datasets, there are several descriptions for the same video, that is, for every video x exists at least two descriptions y_1 and y_2 such that $y_1 \neq y_2$ and $(x, y_1), (x, y_2) \in \mathcal{D}$. In the experiments we use datasets in which every video has at least 20 descriptions.

One way of solving a video-caption problem is to train a model that first constructs an *encoder* representation from x, say $\mathrm{enc}(x)$ and then, using $\mathrm{enc}(x)$ as input, *decodes* it to produce the sequence y each word at a time. In this paper, we propose a Deep Learning *encoder-decoder* architecture that uses classical building blocks such as CNNs and RNNs but with some crucial additions to mix visual, semantic, and syntactic features from the training data to boost its performance (Figure 2). We developed our method under the assumption that integrating semantic-concept representations with syntactic representations can improve the quality of generated sentences. Compared with previous work, our main contribution is the addition of the syntactic part. As we show in the experimental section below, this component allows us to improve the state of the art in video captioning.

Now, we describe our model for predicting a video's global syntactic representation. Next, we introduce an effective way to fuse semantic and syntactic temporal representations into our video captioning method's decoder.

3.1. Encoder and Visual-Syntactic Embedding

For encoding the input video x, we propose an architecture of three stages. The first stage consists in compressing the video into a global representation that we denote by $\rho(\cdot)$, which combines two standard visual features extractor. Specifically, we sample p frames from x and extract 2D-CNN feature vectors $\{a_1, a_2, \ldots, a_p\}$ and 3D-CNN feature vectors $\{m_1, m_2, \ldots, m_p\}$ intuitively representing the appearance and motion information of the video, respectively. Then, these features are concatenated and averaged to produce $\rho(x)$, that is, $\rho(x) = \frac{1}{p}\sum_{i=1}^{p}[a_i, m_i]$. We emphasize that in our proposal, the feature extractors (2D- and 3D-CNNs) are fixed. In other words, we leverage pre-trained feature extractors and do not train them in our architecture.

The second stage in our encoder consists of producing a semantic representation of the video. Based on video captioning studies like Chen *et al*. [3] and Gan *et al*. [17], we use a standard concept detector. The concept detector is essentially a multi-class classifier that gives probabilities for possible keywords that are relevant to describe a specific video. Recall that our task is a video-captioning task, so we need alternative data to train such a classifier. To do this, we first construct a set T of keywords by considering the K most frequent words in all the descriptions in our dataset $\mathcal{D}$ ($|T| = K$). Then, for every video, say $x^\star$, in $\mathcal{D}$, we consider the set of all keywords that appear in the descriptions for $x^\star$, that is, the set

$$\mathrm{tags}(x^\star) = \{w \in T \mid \exists\, y, (x^\star, y) \in \mathcal{D} \text{ and } w \text{ occurs in } y\}.$$

Then, for training the classifier, we use the dataset $\mathcal{D}_T$ composed of all the pairs $(x, \mathrm{tags}(x))$ such that x is a video in $\mathcal{D}$. We consider a standard MLP architecture that has $\rho(x)$ as input, ReLU activation in hidden layers and a sigmoid activation at the output, producing a vector $\mathrm{S}(\rho(x)) \in [0, 1]^K$. We train it with a component-wise binary cross-entropy loss as it is customary for multi-class classifiers. The idea is that the i-th component of $\mathrm{S}(\rho(x))$ is the probability assigned by the model to the fact that $w_i \in \mathrm{tags}(x^\star)$. We use $\mathrm{S}(\rho(x))$ as the concept detector vector.

3.1.1 Visual-Syntactic Embedding

The third stage in our encoder architecture produces what we call *visual-syntactic embedding*. We next explain the

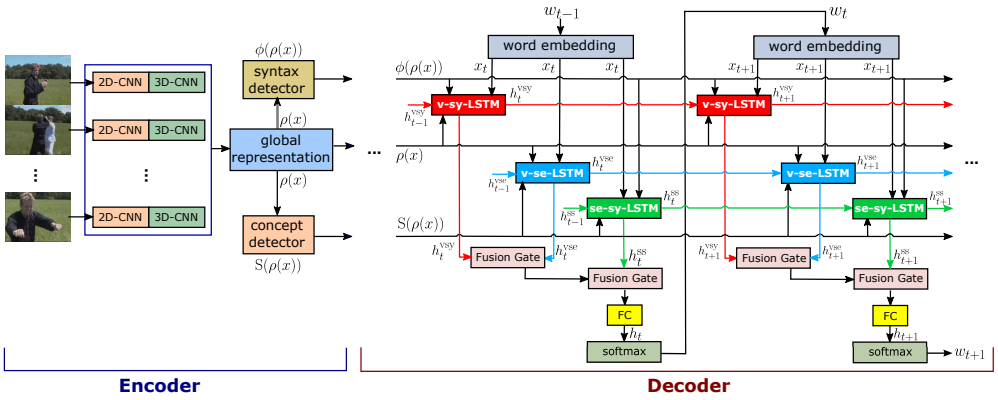

Figure 2. Proposed video captioning model. Firstly, we extract 2D-CNN and 3D-CNN visual features and a global representation $\rho(x)$. Next, the method predicts semantic and syntactic representations of the video by $\mathrm{S}(\rho(x))$ and $\phi(\rho(x))$, respectively. Then, the decoder generates the t-th word dynamically combining these three vectors in pairs. A different VNC_L layer processes each pair.

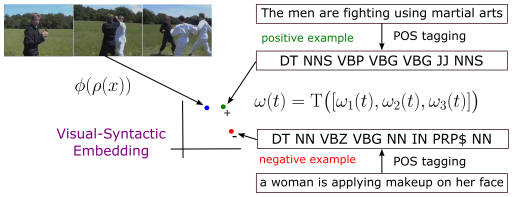

Figure 3. Visual-Syntactic Embedding. The model learns to map from video and POS sequences to a common space by the functions $\phi(\cdot)$ and $\omega(\cdot)$, preserving the relationship between visual content and positive syntactic structures.

main intuition and how they are constructed from the data.

We claim that cues about the syntactic structure of the video's descriptions can be directly extracted from a video without necessarily extracting explicit information about the entities or objects participating. For example, there may be several videos in our dataset that, because of their structure, share a description pattern of the form

$$\langle \text{object1} \rangle \quad \langle \text{object2} \rangle \quad \langle \text{action} \rangle \quad \langle \text{object3} \rangle$$

as in a description like "$\langle$The dog$\rangle$ and $\langle$the cat$\rangle$ $\langle$are lying$\rangle$ on $\langle$the floor$\rangle$." We propose to train a model to compute a suitable syntactic representation of descriptions directly from the input video. We attack this representation learning problem as a *Part-Of-Speech template retrieval* problem.

Given a pair $(x, y) \in \mathcal{D}$, our strategy is to learn how to map the feature vector $\rho(x)$ and the sequence of POS tags of y into a d-dimensional common space (Figure 3). Specifically, the aim is to learn two mapping functions $\phi(\cdot)$ and $\omega(\cdot)$ (encoders) that map from visual features and POS template vectors, respectively, to the joint embedding space.

Before explaining the specific architecture that we use for mappings $\phi(\cdot)$ and $\omega(\cdot)$, we describe how we train them to construct embeddings into a common space by using the *triplet-ranking-loss approach* [11]. For now it is enough to assume that both $\phi(\cdot)$ and $\omega(\cdot)$ produces vectors in $\mathbb{R}^d$. Assume that $\mathrm{dist}(\cdot,\cdot)$ is a distance function in $\mathbb{R}^d$. Let $(x, y) \in \mathcal{D}$ and assume that $y^\star$ is an arbitrary description in the dataset not associated with x (a negative example). We would like the following to hold

$$\mathrm{dist}\big(\phi(\rho(x)),\, \omega(y)\big) + \alpha < \mathrm{dist}\big(\phi(\rho(x)),\, \omega(y^\star)\big),$$

where α is a margin that is enforced between positive and negative pairs. This gives rise to a natural optimization problem in which one wants to minimize

$$\max\Big\{0,\ \mathrm{dist}\big(\phi(\rho(x)),\, \omega(y)\big) + \alpha - \mathrm{dist}\big(\phi(\rho(x)),\, \omega(y^\star)\big)\Big\}$$

over all triples $(x, y, y^\star)$ where $(x, y) \in \mathcal{D}$ and $y^\star$ is a negative example. This formulation is the classic version of triple-ranking loss. In our architecture, we use an improved version, which penalizes the model taking into account the hardest negative examples [11, 13]. In this improved version, one considers a tuple $(x, y, x^\star, y^\star)$ where $(x, y) \in \mathcal{D}$, $y^\star$ is the closest negative example for x, and $x^\star$ is the closest negative example for y. We refer the reader to Dong *et al.* [11] and Faghiri *et al.* [13] for details.

Given the way for training the mappings, we can explain the architecture of both encoders $\phi(\cdot)$ and $\omega(\cdot)$. On the one hand, we define the visual encoder $\phi(\cdot) \to \mathbb{R}^d$ similar to the concept detector model, but removing the sigmoid activation function and incorporating a Batch Normalization layer. On the other hand, to map the sequence of POS tags, we define $\omega(\cdot) \to \mathbb{R}^d$ extending the encoding proposed in [11] for sentences. Specifically, $\omega(\cdot)$ encodes a POS tags sequence t by concatenating three intermediate representations of the sequence that we call *global*, *temporal*, and *local-enhancing*, denoted by $\omega_1(t)$, $\omega_2(t)$, and $\omega_3(t)$, respectively. Then, this concatenation is projected in the embedding space by another MLP T, that is,

$\omega(t) = \mathrm{T}\big([\omega_1(t), \omega_2(t), \omega_3(t)]\big)$. We next describe how $\omega_1(t)$, $\omega_2(t)$, and $\omega_3(t)$ are computed.

The global representation $\omega_1(t)$ is computed simply as a bag-of-words representation of the POS tags in t. Given that we consider 28 POS tags, $\omega_1(t)$ is a 28-dimensional vector. The temporal representation $\omega_2(t)$ is constructed by using a recurrent network over the POS tag sequence. More specifically, we first compute the sequence $H = (\hat{h}_1, \hat{h}_2, \ldots, \hat{h}_m)$ as the output of a bidirectional GRU network over the POS tags $t = (t_1, t_2, \ldots, t_m)$. Then $\omega_2(t)$ is computed as the average of the vectors in H. Finally, the local-enhancing representation $\omega_3(t)$ is computed as a one-dimensional convolution over the sequence H and a final max pooling.

3.2. The Decoder

Given the output of our encoder, *i.e.*, the averaged feature representation $\rho(x)$, the concept detector vector $\mathrm{S}(\rho(x))$ and the visual-syntactic encoding $\phi(\rho(x))$, our decoder network, that we denote by ψ, generates the natural language description $y = \psi\big(\rho(x), \mathrm{S}(\rho(x)), \phi(\rho(x))\big)$. We define ψ as a recurrent architecture that generates the tokens in y each word at a time. This recurrent architecture has four components to dynamically decide when to use visual-semantic, visual-syntactic, or semantic-syntactic temporal information in the generation process (Figure 2). Our decoder deals with the usual overfitting of video captioning models by incorporating dropout and layer normalization strategies.

In detail, our decoder has three specialized recurrent layers based on compositional-LSTM network [3, 17], and an additional layer to combine the outputs of the three recurrent layers defined by two levels of what we call *fusion gates*. The role of this combination layer is to adaptively mix the outputs of the three recurrent layers, while the role of the recurrent layers is to capture temporal states related to a specific pair of feature information (*i.e.*, visual-semantic, visual-syntactic, and semantic-syntactic). Unlike other architectures [2, 19, 41], our recurrent layers are not deeply connected (the output of one is not the input of another). Intuitively, each one is in charge of combining two different information channels separately. So, we compute the three layers in parallel without increasing the execution time.

Var-Norm-Compositional LSTM (VNC_L). As a building block, we use a recurrent layer that we call *Var-Norm-Compositional LSTM* that is defined as follows. We first define a general operator F over matrices and vectors as

$$F(u, v, m_u, m_v, U, V, W) = W\big(U(u \odot m_u) \odot V(v \odot m_v)\big).$$

Now consider as input the vectors q, r and the sequence $x_1, \ldots, x_N$. We define two sequences $h_1, \ldots, h_N$ and $c_1, \ldots, c_N$ by using F in an LSTM-like way recursively as follows. Let $*$ represent i (input), f (forget), o (output) and c (cell), as in the gates of an LSTM. We first define the intermediate vectors $\hat{z}_*$, $\hat{x}_{*,t}$ and $\hat{h}_{*,t}$ by

$$\begin{aligned}
\hat{z}_* &= F(q, r, m_{*,q}, m_{*,r}, C_{*,1}, C_{*,2}, C_{*,3}),\\
\hat{x}_{*,t} &= F(x_t, r, m_{*,x}, m_{*,r}, W_{*,1}, W_{*,2}, W_{*,3}),\\
\hat{h}_{*,t} &= F(h_{t-1}, r, m_{*,h}, m_{*,r}, U_{*,1}, U_{*,2}, U_{*,3}),
\end{aligned}$$

where $W_{*,j}$, $U_{*,j}$ and $C_{*,j}$ with $j = 1, 2, 3$ are weight matrices to be learned, and $m_{*,q}$, $m_{*,r}$, $m_{*,x}$ and $m_{*,h}$ are the dropout masks applied to q, r, x_t and h_{t-1}, one for each gate. We note that these dropout masks are the same for every step t and, as every dropout, are picked randomly for every training step. Then, we compute the gates as

$$\begin{aligned}
\hat{i}_t &= \sigma(\mathrm{BN}(\hat{x}_{i,t} + \hat{z}_i + \hat{h}_{i,t} + b_i)),\\
\hat{f}_t &= \sigma(\mathrm{BN}(\hat{x}_{f,t} + \hat{z}_f + \hat{h}_{f,t} + b_f)),\\
\hat{o}_t &= \sigma(\mathrm{BN}(\hat{x}_{o,t} + \hat{z}_o + \hat{h}_{o,t} + b_o)),\\
\hat{c}_t &= \tanh(\mathrm{BN}(\hat{x}_{c,t} + \hat{z}_c + \hat{h}_{c,t} + b_c)),
\end{aligned}$$

where b_* is a bias vector to be learned for each gate $* \in \{i, f, o, c\}$ and BN denotes a Batch Normalization layer applied before the activation functions. Finally, we have that

$$\begin{aligned}
c_t &= \hat{f}_t \odot c_{t-1} + \hat{i}_t \odot \hat{c}_t,\\
h_t &= \hat{o}_t \odot \tanh(c_t).
\end{aligned}$$

We denote the above recursive process simply as

$$(h_t, c_t) = \mathrm{VNC}_L(h_{t-1}, c_{t-1}, x_t, q, r) \tag{1}$$

Now, given the VNC_L definition, we can define each layer in our recurrent decoder architecture.

The **Visual-Semantic layer (v-se-LSTM)** incorporates semantic information focusing on the meaning of the visual and language context information. v-se-LSTM takes into account the global visual representation $\rho(x)$ and the result of our concept detector $\mathrm{S}(\rho(x))$.

$$(h_t^{\mathrm{vse}}, c_t^{\mathrm{vse}}) = \mathrm{VNC}_L\big(h_{t-1}^{\mathrm{vse}}, c_{t-1}^{\mathrm{vse}}, x_t, \rho(x), \mathrm{S}(\rho(x))\big)$$

The **Visual-Syntactic layer (v-sy-LSTM)** incorporates syntactic information focusing on the structure of the visual and POS tagging information. v-sy-LSTM takes into account the visual information $\rho(x)$ and its projection in the visual-syntactic embedding $\phi(\rho(x))$.

$$(h_t^{\mathrm{vsy}}, c_t^{\mathrm{vsy}}) = \mathrm{VNC}_L\big(h_{t-1}^{\mathrm{vsy}}, c_{t-1}^{\mathrm{vsy}}, x_t, \rho(x), \phi(\rho(x))\big)$$

The **Semantic-Syntactic layer (se-sy-LSTM)** processes the semantic primitives $\mathrm{S}(\rho(x))$ and the syntactic representation $\phi(\rho(x))$, regardless of the visual features information. The temporal semantic-syntactic-related information allows the decoder to generate words without considering the visual content, *e.g.*, linking verbs.

$$h_t^{\mathrm{ss}}, c_t^{\mathrm{ss}} = \mathrm{VNC}_L\big(h_{t-1}^{\mathrm{ss}}, c_{t-1}^{\mathrm{ss}}, x_t, \mathrm{S}(\rho(x)), \phi(\rho(x))\big)$$

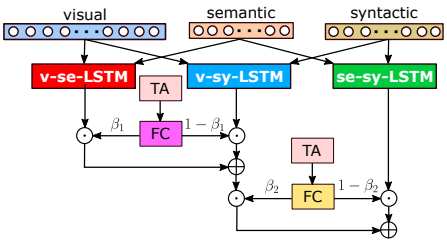

Figure 4. Hierarchical fusion strategy. Two adaptive gates are connected in a cascade way. TA represents our Temporal Attention module. β_1 and β_2 are weight-vectors computed by the sigmoid of two fully-connected layers (FC). $\odot$ and $\bigoplus$ are the element-wise multiplication and addition.

3.2.1 Combination with Hierarchic Fusion Gates

For combining the output of the three recurrent units, we propose a *hierarchical fusion* strategy before generating the word in each step (Figure 4). This strategy is an essential component of our model. It consists of a temporal attention mechanism (TA) and two adaptive gates that intuitively decide how and when to use (or forget) the visual-related information from v-se-LSTM and v-sy-LSTM layers. For TA, we base our mechanism on the *soft attention* [1]. With this strategy, the decoder learns to dynamically weight the temporal feature vectors in each step. For each video, with visual features $[a_i, m_i]$ at i-th frames segment, we compute the TA in step t by $a_t = \frac{1}{p}\sum_{i=1}^{p}\alpha_i[a_i, m_i]$, where α_i is the weight related to the i-th frames segment. We compute these weights considering the three recurrent layers' hidden states by $\alpha_i = \text{softmax}\big(W_{a,2}\cdot(W_{a,1}\cdot[h_t^{\text{vse}}, h_t^{\text{vsy}}, h_t^{\text{ss}}]+b_a)\big)$, where $W_{a,1}$, $W_{a,2}$ and b_a are parameters to be learned.

Given the TA in step t and the decoder's output in the previous step h_{t-1}, the fusion gates select the most accurate information between h_t^{vse}, h_t^{vsy} and h_t^{ss}. The model first fuses the visual-related layers (h_t^{vse} and h_t^{vsy}) in a temporal vector. Then, this vector is fused with the semantic-syntactic-related information h_t^{ss}, increasing the chance of generating captions with correct semantic meaning and syntactic structure. We define this component as follows:

$$h_t = \beta_2 \odot \big(\beta_1 \odot h_t^{\text{vse}} + (1-\beta_1)\odot h_t^{\text{vsy}}\big) + (1-\beta_2)\odot h_t^{\text{ss}},$$

such that,

$$\beta_1 = \sigma\big(W_{h,1}\cdot[h_{t-1}, a_t] + b_{h,1}\big),$$
$$\beta_2 = \sigma\big(W_{h,2}\cdot[h_{t-1}, a_t] + b_{h,2}\big),$$

where $W_{h,1}$, $W_{h,2}$, $b_{h,1}$ and $b_{h,2}$ are learnable parameters.

3.3. Syntax-weighted Loss

As a language generation task, the video captioning models are usually trained by the principle of Maximum Likelihood Estimation, also known as Cross-Entropy minimization (CELoss) [20]. However, the weak relationship of the CELoss function with the popular evaluation metrics of video captioning [45, 47], constitutes a limitation of its use.

To overcome this limitation while aiming to consider syntactic information in the training phase, we propose the syntax-weighted loss function. Our function improves the loss used by Chen *et al*. [3], considering the distance between the syntactic representation and the POS structure of the generated description. Thus, given a video x, the ground-truth caption $y = (y_1, y_2, \ldots, y_L)$ of x, and the POS tagging t of the generated description, we define the weight $w = \max\big\{1, L^{\beta} - \big(\text{dist}\big(\phi(\rho(x)), \omega(t)\big) + 1\big)^{\gamma}\big\}$, and we minimize

$$-\frac{1}{w}\sum_{i=1}^{L}\log p_\theta(y_i|y_{z<i}), \tag{2}$$

where $\beta \in [0, 1]$ and $\gamma \in [0, 1]$ are hyperparameters used to manage the balance between the length (conciseness) and syntactic correctness of generated descriptions. Greater β implies longer captions, and greater γ implies better syntax.

4. Experimental Evaluation

For the evaluation, we use two widely used benchmark datasets that are publicly available: MSVD [33] and MSR-VTT [56]. On the one hand, the MSVD dataset contains 1,970 videos with 41 descriptions per video on average and a total of 12,859 unique words. We use the MSVD's standard split, *i.e*., 1,200, 100, and 670 videos for training, validation, and testing. On the other hand, the MSR-VTT dataset contains about 50 hours, with 200,000 clip-sentence pairs (approximately 20 descriptions per clip), covering a broad range of categories and diverse visual content. We also use the MSR-VTT's standard split, *i.e*., 6,512, 498, and 2,990 clips for training, validation, and testing. For comparing, we report results on four popular evaluation metrics using the Microsoft COCO evaluation server [4].

4.1. Training Setup

To extract 2D-CNN features of the video, we use ResNet-152 [22] feature extractor pre-trained on ImageNet [7, 49]. For 3D-CNN, we use ECO [62] and R(2+1)D [51] feature extractors, both pre-trained on Kinetics-400 dataset. On details, for frame-level representations, we concatenate the ResNet-152, and ECO features vectors, resulting in 3584-dimensional feature vectors. Concerning global representation, we average these features and concatenate it with the 512-dimensional R(2+1)D feature, obtaining a 4096-dimension global representation. To represent text descriptions, we obtain the vocabulary from the training set of each dataset. Next, we map each description to a sequence of indices in the vocabulary, putting the $\langle$eos$\rangle$ and $\langle$unk$\rangle$ tokens at the end and in the positions of unknown words.

Table 1. Ablation study on the testing set of MSVD and MSR-VTT datasets. Each row reports the results by changing only one aspect of the method, *e.g.*, the architecture *−(v-se, se-sy)* shows the results obtained by the model without the v-se-LSTM and se-sy-LSTM layers.

Architecture	MSVD				MSR-VTT			
	BLEU-4	METEOR	CIDEr	ROUGE$_L$	BLEU-4	METEOR	CIDEr	ROUGE$_L$
SemSynAN (ours)	**64.4**	**41.9**	**111.5**	**79.5**	**46.4**	**30.4**	**51.9**	**64.7**
−v-sy	59.4	39.4	107.2	77.0	45.3	29.0	48.0	62.4
−se-sy	58.3	39.5	106.8	76.3	44.6	29.6	48.9	62.6
−(v-se, se-sy)	48.5	34.3	75.8	72.1	40.7	27.6	42.6	60.5
−(v-se, v-sy)	56.7	37.2	92.0	74.9	45.3	28.8	47.7	62.4
−hfg	60.8	41.3	103.9	75.9	45.8	29.7	48.0	63.0
−R(2+1)D	61.9	39.7	105.4	77.0	43.7	29.6	50.0	61.6
−wl	50.5	39.8	98.1	73.5	43.9	26.9	43.0	59.8
−vd	49.2	39.4	101.4	74.0	43.6	27.2	45.2	61.9
−max	63.7	41.2	108.1	79.0	45.6	30.1	48.0	63.1

For the visual-syntactic embedding, we set the dimension of the common space to 512, the hidden sizes of the *visual model* to 2048 and 1024, and the hidden size of the biGRU layer of the *syntactic model* to 1024. We trained the model on the MSR-VTT dataset using the *cosine distance* as the dist$(\cdot,\cdot)$ function, a learning rate of 1×10^{-5} and a margin parameter of 0.1. Some methods like LSTM-E [42] use all ground-truth captions while others like LJRV [40] and Dong *et al*. [11] randomly sample five ground-truth captions per video. We follow the latter strategy. By sampling a random subset, the most frequent syntactic structure is most likely to be selected. Our results demonstrate that, in MSVD and MSR-VTT, five samples are sufficient for learning the cues about the syntactic structure of video captions.

We use Adam optimizer with an initial learning rate of 4×10^{-5} for the MSR-VTT dataset and 2×10^{-5} for MSVD and a batch-size of 64. We trained for at least 50 epochs with early-stopping criteria of 10 epochs. Each VNC$_L$ layer has a hidden size of 1024, and we use a keep probability of 0.8 for their dropout masks and 0.5 in all other cases. For our *syntax-weighted loss* function, we set the parameters $\beta = 0.7$ and $\gamma = 0.9$ and use the *cosine distance*.We fine-tune the hyperparameters on the validation sets and select the best checkpoint for testing according to a linear combination of BLEU-4, METEOR, CIDEr, and ROUGE$_L$ measures. We implemented our models and training methods on PyTorch[1] [46]. They are publicly available on GitHub[2].

4.2. Results and Analysis

Ablation Study. Table 1 shows the results of nine ablated experiments that we performed on the MSVD and MSR-VTT datasets. Specifically, we evaluate our SemSynAN model by removing, in separate runs, one or two of our VNC$_L$ layers, the *fusion gates*, the *weighted-loss* function, the dropout masks and the maximum sampling strategy.

−v-sy and *−se-sy*. These two runs figure out the contribution of each VNC$_L$ layers that process syntactic information. These runs' models have two VNC$_L$ layers only and combine them by only one fusion gate.

[1] https://pytorch.org

[2] Our code is available at https://github.com/jssprz/visual_syntactic_embedding_video_captioning

Table 2. Performance comparison with the state-of-the-art methods on the testing set of MSVD dataset.

Approach	BLEU-4	METEOR	CIDEr	ROUGE$_L$
LSTM-E [42]	45.3	31.0	-	-
SCN-LSTM [17]	51.1	33.5	77.7	-
TDDF [60]	45.8	33.3	73.0	69.7
MTVC [44]	54.5	36.0	92.4	72.8
BAE [2]	42.5	32.4	63.5	-
MFATT-TM-SP [36]	52.0	33.5	-	-
ECO [62]	53.5	35.0	85.8	-
SibNet [35]	54.2	34.8	88.2	71.7
Joint-VisualPOS [25]	52.8	36.1	87.8	71.5
GFN-POS_RL(IR+M) [54]	53.9	34.9	91.0	72.1
hLSTMat [19]	54.3	33.9	73.8	-
SAVCSS [3]	61.8	37.8	103.0	76.8
DSD-3 DS-SEM [24]	50.1	34.7	76.0	73.1
ORG-TRL [61]	54.3	36.4	95.2	73.9
SemSynAN (ours)	**64.4**	**41.9**	**111.5**	**79.5**

−(v-se, se-sy) and *−(v-se, v-sy)*. These runs also evaluate the effectiveness of introducing syntactic information but removing two VNC$_L$ layers. These runs' models are based on only one syntactic-related layer.

−hfg. In this experiment, the decoder computes the output in each step by concatenating the output of each layer instead of using our hierarchical gated fusion.

−wl. In this run, we trained the model with the CELoss function for optimizing the model. CELoss cannot extract the effectiveness of our model as well as our *syntax-weighted loss* in MSVD and MSR-VTT.

−vd. In this run, we trained the model without using our dropout masks as part of VNC$_L$ layers. Using the same dropout masks for every step has a high impact on the model performance in both datasets.

−max. In this run, we sampled the word from the output multinomial probability distribution in the training phase, making the higher probabilities more likely to be sampled. For testing, we always chose the argmax.

The first four rows of Table 1 demonstrate that our model is significantly enhanced by including the syntactic information on both datasets, proving the proposed method's effectiveness. Overall, the performance of our model is improved with the incorporation of each component.

Comparison with State of the Art on MSVD. Table 2 shows the performance of the proposed approach and other state-of-the-art methods on the MSVD dataset. The SCN-

Table 3. Performance comparison with the state-of-the-art methods on the testing set of MSR-VTT dataset. * denotes results that were obtained by reinforcement learning of that metric.

Approach	BLEU-4	METEOR	CIDEr	ROUGE$_L$
TDDF [60]	37.3	27.8	43.8	59.2
MTVC [44]	40.8	28.8	47.1	60.2
CIDEnt_RL [45]	40.5	28.4	51.7*	61.4
HRL [55]	41.3	28.7	48.8*	61.7
PickNet [6]	38.9	27.2	42.1	59.5
MFATT-TM-SP [36]	39.1	26.7	-	-
SibNet [35]	40.9	27.5	47.5	60.2
Joint-VisualPOS [25]	42.3	29.7	49.1	62.8
GFN-POS_RL(IR+M) [54]	41.3	28.7	**53.4***	62.1
hLSTMat [19]	39.7	27.0	43.4	-
SAVCSS [3]	43.8	28.9	51.4*	62.4
DSD-3 DS-SEM [24]	45.2	29.9	51.1	64.2
ORG-TRL [61]	43.6	28.8	50.9	62.1
SemSynAN (ours)	**46.4**	**30.4**	51.9	**64.7**

LSTM [17], and SAVCSS [3] methods process a semantic representation by visual-semantic compositional LSTM decoders, without considering the syntactic information. The incorporation of syntactic representation with our compositional modules improves the performance in comparison to those approaches. Likewise, the superior performance of our sentence generator framework is demonstrated in comparison to model that exploit fixed encoding based on 2D-CNN and 3D-CNN features, such as LSTM-E, SCN-LSTM, SAVCSS. Two recent approaches [25, 54] use the syntactic information from the POS tagging structure but do not directly consider temporal relations between the visual, semantic, and syntactic representations. In the proposed approach, the semantic and syntactic representations are adaptively fused with the visual features, determining the most accurate information for generating each word. Hence, it is seen that our SemSynAN provides better scores than the previous syntax-based approaches. Specifically, our method has a relative BLEU-4 improvement of 4.2% $\left(\frac{64.4-61.8}{61.8}\right)$, METEOR of 10.8% $\left(\frac{41.9-37.8}{37.8}\right)$, CIDEr of 8.2% $\left(\frac{111.5-103.0}{103.0}\right)$, and ROUGE$_L$ of 3.5% $\left(\frac{79.5-76.8}{76.8}\right)$.

Comparison with State of the Art on MSR-VTT. Table 3 compares the performance of our SemSynAN model with the recently published results on the MSR-VTT dataset. Our approach surpasses the methods that exploit the POS tagging structure of video captions [25, 54], and the approaches based on visual-semantic embeddings [35] and compositions [3, 17]. Unlike CIDEnt_RL [45], HRL [55], GFN-POS_RL(IR+M) [54], and SAVCSS [3], we do not use reinforcement learning to directly maximize any metric. However, our approach improves the results in terms of all metrics except CIDEr, where GFN-POS_RL(IR+M) [54] rich better score by reinforcing this score. Specifically, our model has a relative BLEU-4 improvement of 2.6% $\left(\frac{46.4-45.2}{45.2}\right)$, METEOR of 1.7% $\left(\frac{30.4-29.9}{29.9}\right)$, and ROUGE$_L$ of 0.8% $\left(\frac{64.7-64.2}{64.2}\right)$. While, in terms of relative CIDEr, our approach outperforms the models without reinforcement learning by 1.6% $\left(\frac{51.9-51.1}{51.1}\right)$.

Figure 5. Three representative samples from the test split of MSVD, which cover ground-truth captions and their POS structure, two of our ablation models, and our proposal. Highlighted, the words and POS tags that the model predicted correctly.

4.2.1 Qualitative Analysis

Figure 5 shows the predictions of our model for three video examples of the MSVD dataset. To observe the improvement in the captions generated by our model, we compared these predictions with the outputs of two of our ablated models, *i.e.*, *−v-sy* and *−(v-se, se-sy)*. We highlighted some words and POS tags where the model combined the semantic and syntactic information correctly. In these three examples, we can notice that our proposal generates better descriptions than the ablated models.

In the first example, the approach proposed in Section 3 generates the syntactic pattern "NN CC NN". In the second and third examples, different to the ablated models, our approach predicts the syntactic patterns "NN IN DT NN" and "NN IN PRP$ NN" respectively. In the last example, *−v-sy* and *−(v-se, se-sy)* fail generating the noun "face".

5. Conclusions

In this paper, we presented an encoder-decoder model for video captioning named SemSynAN capable of generating sentences with more precise semantics and syntax. As part of this model, we proposed a technique to retrieve POS tagging structures of video descriptions while obtaining a high-level syntactic representation from visual information. We show that paying more attention to syntax improves the quality of descriptions. Our method guarantees the contextual relation between the words in the sentence, controlling the semantic meaning and syntactic structure of generated captions. The experimental results demonstrate that our approach improves the state of the art on two of the most utilized evaluation benchmarks on video captioning.

References

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations*, 9 2015.

[2] Lorenzo Baraldi, Costantino Grana, and Rita Cucchiara. Hierarchical Boundary-Aware Neural Encoder for Video Captioning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3185–3194. IEEE, 7 2017.

[3] Haoran Chen, Ke Lin, Alexander Maye, Jianming Li, and Xiaolin Hu. A Semantics-Assisted Video Captioning Model Trained with Scheduled Sampling. 8 2019.

[4] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollar, and C. Lawrence Zitnick. Microsoft COCO Captions: Data Collection and Evaluation Server, 4 2015.

[5] Xinlei Chen and C. Lawrence Zitnick. Mind's eye: A recurrent visual representation for image caption generation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 07-12-June, pages 2422–2431. IEEE, 6 2015.

[6] Yangyu Chen, Shuhui Wang, Weigang Zhang, and Qingming Huang. Less Is More: Picking Informative Frames for Video Captioning. In *Computer Vision – ECCV 2018*, pages 367–384. Springer International Publishing, 2018.

[7] 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, Miami, FL, US, 2009. IEEE.

[8] Aditya Deshpande, Jyoti Aneja, Liwei Wang, Alexander G. Schwing, and David Forsyth. Fast, Diverse and Accurate Image Captioning Guided by Part-Of-Speech. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10687–10696. IEEE, 6 2019.

[9] Jeff Donahue, Lisa Anne Hendricks, Marcus Rohrbach, Subhashini Venugopalan, Sergio Guadarrama, Kate Saenko, and Trevor Darrell. Long-Term Recurrent Convolutional Networks for Visual Recognition and Description. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(4):677–691, 2015.

[10] Jianfeng Dong, Xirong Li, and Cees G. M. Snoek. Predicting Visual Features From Text for Image and Video Caption Retrieval. *IEEE Transactions on Multimedia*, 20(12):3377–3388, 12 2018.

[11] Jianfeng Dong, Xirong Li, Chaoxi Xu, Shouling Ji, Yuan He, Gang Yang, and Xun Wang. Dual Encoding for Zero-Example Video Retrieval. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9338–9347. IEEE, 6 2019.

[12] Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. Understanding Back-Translation at Scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500, Stroudsburg, PA, USA, 2018. Association for Computational Linguistics.

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

[14] Hao Fang, Saurabh Gupta, Forrest Iandola, Rupesh K. Srivastava, Li Deng, Piotr Dollar, Jianfeng Gao, Xiaodong He, Margaret Mitchell, John C. Platt, C. Lawrence Zitnick, and Geoffrey Zweig. From captions to visual concepts and back. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 07-12-June, pages 1473–1482. IEEE, 6 2015.

[15] Christoph Feichtenhofer, Axel Pinz, and Richard P. Wildes. Spatiotemporal Multiplier Networks for Video Action Recognition. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7445–7454. IEEE, 7 2017.

[16] Chuang Gan, Zhe Gan, Xiaodong He, Jianfeng Gao, and Li Deng. StyleNet: Generating Attractive Visual Captions with Styles. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 955–964. IEEE, 7 2017.

[17] Zhe Gan, Chuang Gan, Xiaodong He, Yunchen Pu, Kenneth Tran, Jianfeng Gao, Lawrence Carin, and Li Deng. Semantic Compositional Networks for Visual Captioning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2017-Janua, pages 1141–1150. IEEE, 7 2017.

[18] Lianli Gao, Zhao Guo, Hanwang Zhang, Xing Xu, and Heng Tao Shen. Video Captioning with Attention-Based LSTM and Semantic Consistency. *IEEE Transactions on Multimedia*, 19(9), 2017.

[19] Lianli Gao, Xiangpeng Li, Jingkuan Song, and Heng Tao Shen. Hierarchical LSTMs with Adaptive Attention for Visual Captioning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–19, 1 2019.

[20] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. 2016.

[21] Sergio Guadarrama, Niveda Krishnamoorthy, Girish Malkarnenkar, Subhashini Venugopalan, Raymond Mooney, Trevor Darrell, and Kate Saenko. YouTube2Text: Recognizing and Describing Arbitrary Activities Using Semantic Hierarchies and Zero-Shot Recognition. In *2013 IEEE International Conference on Computer Vision*, volume 1, pages 2712–2719. IEEE, 12 2013.

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2016-Decem, pages 770–778. IEEE, 6 2016.

[23] Xinwei He, Baoguang Shi, Xiang Bai, Gui Song Xia, Zhaoxiang Zhang, and Weisheng Dong. Image Caption Generation with Part of Speech Guidance. *Pattern Recognition Letters*, 119:229–237, 3 2019.

[24] M Hemalatha and C Chandra Sekhar. Domain-Specific Semantics Guided Approach to Video Captioning. In *The IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1587–1596, 3 2020.

[25] Jingyi Hou, Xinxiao Wu, Wentian Zhao, Jiebo Luo, and Yunde Jia. Joint Syntax Representation Learning and Visual

Cue Translation for Video Captioning. In *The IEEE International Conference on Computer Vision (ICCV)*, 2019.

[26] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3D Convolutional Neural Networks for Human Action Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):221–231, 1 2013.

[27] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3128–3137. IEEE, 6 2015.

[28] Ryan Kiros, Ruslan Salakhutdinov, and Richard S. Zemel. Unifying visual-semantic embeddings with multimodal neural language models, 12 2014.

[29] Atsuhiro Kojima, Takeshi Tamura, and Kunio Fukunaga. Natural Language Description of Human Activities from Video Images Based on Concept Hierarchy of Actions. *International Journal of Computer Vision*, 50(2):171–184, 2002.

[30] Yu Kong and Yun Fu. Human Action Recognition and Prediction: A Survey, 6 2018.

[31] Niveda Krishnamoorthy, Girish Malkarnenkar, Raymond Mooney, Kate Saenko, and Sergio Guadarrama. Generating Natural-Language Video Descriptions Using Text-Mined Knowledge. *NAACL HLT Workshop on Vision and Language*, pages 10–19, 2013.

[32] Polina Kuznetsova, Vicente Ordonez, Tamara Berg, and Yejin Choi. TREETALK: Composition and Compression of Trees for Image Descriptions. *Transactions of the Association of Computational Linguistics*, 2(1):351–362, 2014.

[33] David L. Chen and William B. Dolan. Collecting highly parallel data for paraphrase evaluation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, volume 1, pages 190–200. Association for Computational Linguistics, 2011.

[34] Ji Lin MIT, Chuang Gan, and Song Han MIT. TSM: Temporal Shift Module for Efficient Video Understanding. In *The IEEE International Conference on Computer Vision (ICCV)*, 2019.

[35] Sheng Liu, Zhou Ren, and Junsong Yuan. SibNet: Sibling convolutional encoder for video captioning. In *MM 2018 - Proceedings of the 2018 ACM Multimedia Conference*, pages 1425–1434. Association for Computing Machinery, Inc, 10 2018.

[36] Xiang Long, Chuang Gan, and Gerard De Melo. Video Captioning with Multi-Faceted Attention. In *Transactions of the Association for Computational Linguistics*, pages 173–184, 2018.

[37] Junhua Mao, Wei Xu, Yi Yang, Jiang Wang, Zhiheng Huang, and Alan Yuille. Deep Captioning with Multimodal Recurrent Neural Networks (m-RNN), 12 2014.

[38] Antoine Miech, Ivan Laptev, and Josef Sivic. Learning a Text-Video Embedding from Incomplete and Heterogeneous Data, 4 2018.

[39] Niluthpol C. Mithun, Juncheng Li, Florian Metze, and Amit K. Roy-Chowdhury. Joint embeddings with multimodal cues for video-text retrieval. *International Journal of Multimedia Information Retrieval*, 8(1):3–18, 3 2019.

[40] Mayu Otani, Yuta Nakashima, Esa Rahtu, Janne Heikkilä, and Naokazu Yokoya. Learning Joint Representations of Videos and Sentences with Web Image Search. In *Computer Vision – ECCV 2016*, pages 651–667. Springer International Publishing, 2016.

[41] Pingbo Pan, Zhongwen Xu, Yi Yang, Fei Wu, and Yueting Zhuang. Hierarchical Recurrent Neural Encoder for Video Representation with Application to Captioning. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1029–1038. IEEE, 6 2016.

[42] Yingwei Pan, Tao Mei, Ting Yao, Houqiang Li, and Yong Rui. Jointly Modeling Embedding and Translation to Bridge Video and Language. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4594–4602. IEEE, 6 2016.

[43] Yingwei Pan, Ting Yao, Houqiang Li, and Tao Mei. Video Captioning with Transferred Semantic Attributes. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2017-Janua, pages 984–992. IEEE, 7 2017.

[44] Ramakanth Pasunuru and Mohit Bansal. Multi-Task Video Captioning with Video and Entailment Generation. In *55th Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 1273–1283, Stroudsburg, PA, USA, 2017. Association for Computational Linguistics.

[45] Ramakanth Pasunuru and Mohit Bansal. Reinforced Video Captioning with Entailment Rewards. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 979–985, Stroudsburg, PA, USA, 2017. Association for Computational Linguistics.

[46] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary Devito Facebook, A I Research, Zeming Lin, Alban Desmaison, Luca Antiga, Orobix Srl, and Adam Lerer. Automatic differentiation in PyTorch. 2017.

[47] Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence Level Training with Recurrent Neural Networks. In *International Conference on Learning Representations*, 11 2016.

[48] Marcus Rohrbach, Wei Qiu, Ivan Titov, Stefan Thater, Manfred Pinkal, and Bernt Schiele. Translating Video Content to Natural Language Descriptions. In *2013 IEEE International Conference on Computer Vision*, number December, pages 433–440. IEEE, 12 2013.

[49] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3):211–252, 12 2015.

[50] Jesse Thomason, Subhashini Venugopalan, Sergio Guadarrama, Kate Saenko, and Raymond Mooney. Integrating Language and Vision to Generate Natural Language Descriptions of Videos in the Wild. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers,*, pages 1218–1227, Dublin, Ireland, 2014.

[51] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A Closer Look at Spatiotemporal Convolutions for Action Recognition. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6450–6459. IEEE, 6 2018.

[52] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhim. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6000–6010, Long Beach, California, USA, 2017. Curran Associates Inc.

[53] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 07-12-June, pages 3156–3164. IEEE, 6 2015.

[54] 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 *The IEEE International Conference on Computer Vision (ICCV)*, 2019.

[55] Xin Wang, Wenhu Chen, Jiawei Wu, Yuan-Fang Wang, and William Yang Wang. Video Captioning via Hierarchical Reinforcement Learning. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4213–4222. IEEE, 6 2018.

[56] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. MSR-VTT: A Large Video Description Dataset for Bridging Video and Language. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5288–5296, 2016.

[57] Ran Xu, Caiming Xiong, Wei Chen, and Jason J Corso. Jointly Modeling Deep Video and Compositional Text to Bridge Vision and Language in a Unified Framework. 2015.

[58] Haonan Yu, Jeffrey Mark Siskind, and West Lafayette. Learning to Describe Video with Weak Supervision by Exploiting Negative Sentential Information. In *2015 Conference on Artificial Intelligence (AAAI)*, pages 3855–3863, Austin, Texas, 2015. AAAI Press.

[59] Jin Yuan, Chunna Tian, Xiangnan Zhang, Yuxuan Ding, and Wei Wei. Video Captioning with Semantic Guiding. In *2018 IEEE Fourth International Conference on Multimedia Big Data (BigMM)*, pages 1–5. IEEE, 9 2018.

[60] Xishan Zhang, Yongdong Zhang, Dongming Zhang, Jintao Li, and And Qi Tian. Task-Driven Dynamic Fusion: Reducing Ambiguity in Video Description. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6250–6258. IEEE, 2017.

[61] Ziqi Zhang, Yaya Shi, Chunfeng Yuan, Bing Li, Peijin Wang, Weiming Hu, and Zhengjun Zha. Object Relational Graph with Teacher-Recommended Learning for Video Captioning. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13278–13288, 2020.

[62] Mohammadreza Zolfaghari, Kamaljeet Singh, and Thomas Brox. ECO: Efficient Convolutional Network for Online Video Understanding. In *Computer Vision – ECCV 2018*, pages 713–730. Springer International Publishing, 2018.