Attentive Visual Semantic Specialized Network for Video Captioning

Jesus Perez-Martin, Benjamin Bustos and Jorge Pérez

Millenium Institute for Foundational Research on Data (IMFD), Chile
Department of Computer Science (DCC), University of Chile
Beauchef 851
Email: jeperez@dcc.uchile.cl

***Abstract*—As an essential high-level task of video understanding topic, automatically describing a video with natural language has recently gained attention as a fundamental challenge in computer vision. Previous models for video captioning have several limitations, such as the existence of gaps in current semantic representations and the inexpressibility of the generated captions. To deal with these limitations, in this paper, we present a new architecture that we call *Attentive Visual Semantic Specialized Network* (AVSSN), which is an encoder-decoder model based on our Adaptive Attention Gate and Specialized LSTM layers. This architecture can selectively decide when to use visual or semantic information into the text generation process. The adaptive gate makes the decoder to automatically select the relevant information for providing a better temporal state representation than the existing decoders. Besides, the model is capable of learning to improve the expressiveness of generated captions attending to their length, using a sentence-length-related loss function. We evaluate the effectiveness of the proposed approach on the Microsoft Video Description (MSVD) and the Microsoft Research Video-to-Text (MSR-VTT) datasets, achieving state-of-the-art performance with several popular evaluation metrics: BLEU-4, METEOR, CIDEr, and ROUGE$_L$.**

***Index Terms*—Specialized long-short term memory (S-LSTM), adaptive attention, teacher forcing, video captioning**

I. Introduction

Automatically generating natural language descriptions of videos represents a fundamental challenge for computer vision and multimedia information retrieval communities. Understanding the contents of a video and generating descriptive captions can be useful for tasks like video indexing and retrieval, and robotics (answer questions about the environment). For example, an automatic video description would greatly help users filter what is attractive to them among the sheer number of videos on YouTube.

Predicting a single sentence from an image has been a fundamental problem for several years [15], [29], [34]. However, more recently, that problem was extended to generate descriptions from a video with only one event [12], [16], [18], [29], [32], [33], or with multiple events [30], [36], [43]. However, video captioning is more challenging than image captioning because the videos are more diverse and complex, regarding the visual content and the associated description.

This work was partially supported by the Department of Computer Science of the University of Chile and the National Agency for Research and Development (ANID)/Millennium Science Initiative Program/ICN17_002. Jesus Perez-Martin is funded by ANID/Doctorado Nacional/2018-21180648.

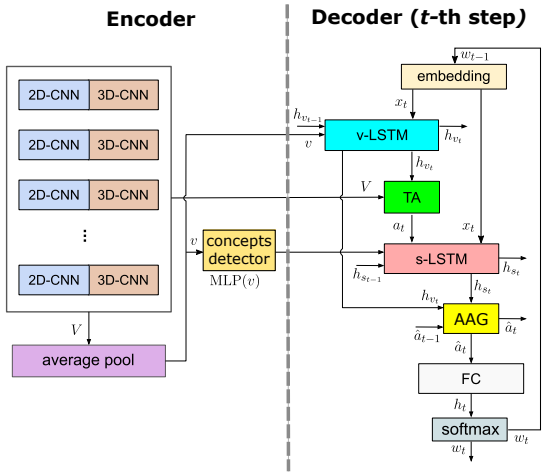

Fig. 1. Proposed Adaptive Visual Semantic Specialized Network (AVSSN). Firstly, the encoder computes V, v and MLP(v) representations, which are the set of all the concatenations of the 2D-CNN and 3D-CNN visual features extracted from the video, their average pool, and a high-level semantic representation, respectively. Then, the recurrent decoder at t-th step composes these three representations by two novel specialized LSTM layers (v-LSTM and s-LSTM), a Temporal Attention mechanism (TA) and a novel Adaptive Attention Gate (AAG).

For video captioning, due to the success of Deep Learning, researchers have proposed many combinations of Convolutional and Recurrent Neural Networks (CNN and RNN) in their frameworks. These proposals consist of a first stage with a neural network for visual recognition (the encoder) based on CNN, and a second stage with a neural network for text generation (the decoder) based on RNN. Almost all these combinations are end-to-end trainable deep network models, in which the two stages are trained simultaneously. Besides, recent works have proposed to improve the effectiveness of the encoding by learning high-level representations from the basic CNN encoder stage. These representations have more semantic information about the video contents, and the authors include this information in the decoder phase, obtaining the state-of-the-art results.

In this paper, we propose a novel encoder-decoder model

(see Figure 1) for video captioning. As part of this model, we include some novel and effective components that could be assessed on other video analysis tasks. Specifically, the main contributions of this paper are as follow:

1) We introduce the Specialized LSTM (S-LSTM) layers: v-LSTM and s-LSTM, which are very useful as visual-dependent and visual-semantic-dependent guiding layers, respectively. The existing decoders process the visual and semantic information by composing or concatenating them, without considering valuable temporal-specialized representations. Our layers can learn temporal representations related to a specific feature domain, *e.g.*, visual, semantic, or syntactic.
2) We proposed the novel Adaptive Attention Gate for integrating two different temporal representations into the decoder network, capable of enhancing the related information within both representations. To our knowledge, we are the first instance of successful incorporation of an explicit adaptive fusion strategy based on visual attention, fusion gates, and cross-products, at the top of the decoder model.
3) We design the so-called Attentive Visual Semantic Specialized Network (AVSSN), a new model for video captioning able to obtain more accurate high-level visual and semantic context representations. For this, encoding representations of the video are processed through two different S-LSTM layers (Section III-B) and guided by a novel Adaptive Attention Gate at each step.
4) We evaluate our proposed model on two video description benchmark datasets: MSVD [19] and MSR-VTT [39]. Our quantitative and qualitative analysis shows superior performance of our proposal, improving the state-of-the-art in both datasets in almost all metrics.

II. Related Work

Previous work deals with the video captioning by template-based models [16], [18], [32], which aim to generate sentences within a reduced set of templates that assure grammatical correctness. To determine the words to fill the templates, we first need to recognize the relevant visual content (Subject-Verb-Object (SVO) triplets). However, for any sufficient variability in the video set's content, the required complexity of rules and templates makes the time-consuming manual design of templates unfeasible or too expensive. Hence, the SVO approaches soon become inadequate in dealing with open-domain datasets.

More recently, deep learning models have shown high performance for many tasks of computer vision, multimedia information retrieval, and video understanding, *e.g.*, action recognition [6], [7], [17]. According to that, the rise of results of all metrics of video description of methods based on deep learning is notable.

A. Semantic Guiding

A way to get more accurate sentences on recent neural-network-based models is by incorporating a semantic concept level in the process. In this sense, Pan *et al.* [22] presented LSTM-TSA, which incorporates transferred semantic attributes learned from images and videos. Similarly, Gan *et al.* [8] proposed the Semantic Compositional Network (SCN-LSTM), which incorporates the semantic meaning via a *semantic-concept detector* into a variation of the conventional LSTM. This variation extends each matrix of weights of the LSTM to a set of matrices of tag-dependent weights subject to the probability of the *tag* is present in the video. In contrast, Yuan *et al.* [41] proposed the Semantic Guiding Long Short-Term Memory (SG-LSTM), a framework that jointly explores visual and semantic features using two semantic guiding layers. More recently, Chen *et al.* [3] modified the model SCN-LSTM [8] by including a semantic-related video feature term at each recurrent step. However, these models strongly depend on the quality of semantic concept detection models, commonly designed as multi-label classification approaches. Our model deals with this limitation by selectively deciding when to use visual information or semantic information at each step of the sentence generation process. We consider some words can be easily predicted using visual information without considering semantic meaning, *e.g.*, colors. Our model manages this automatic selection of the relevant information by maintaining two specialized channels and incorporating an adaptive attention mechanism able to decide when to use the output of each channel.

B. Attention Mechanisms

The frames in videos have different relevance in each step of the decoder model. To determine each frame's relevance and decide where to look at during word prediction, later video captioning models incorporate *visual attention* mechanisms. For example, Yao *et al.* [40] incorporated a temporal attention mechanism adapted from *soft attention* [1], allowing the decoder to weight each temporal feature vector. Intuitively, this mechanism simulates the human attention that sequentially focuses on the most important parts of the information over time to make predictions. More recently, instead of attention over the temporal outputs of the recurrent encoder, Gao *et al.* [9] proposed an attention model able to decide whether to depend on the visual information or language context model. For that, they proposed a hierarchical model of two LSTMs layers and *adaptive attention* that extend the *temporal visual attention*. The bottom LSTM layer contains visual dependent information. The top LSTM layer manipulates in-depth language context information (also using visual dependent information only). The adaptive attention decides which LSTM unit output must be used at each time step. However, these models cannot effectively mix the visual and semantic information because they do not explicitly consider the semantic information for the word prediction process. In contrast, our model incorporates a more effective adaptive gate as a fusion strategy of a visual-dependent layer and a semantic-dependent layer. Likewise, following the "deeper rather than wider" philosophy, we conditioned our adaptive gate and semantic layer by a temporal attention mechanism.

III. Proposed Approach

We propose a CNN-RNN framework (see Figure 1) to generate video descriptions under the assumption that detecting semantic concepts can improve the quality of generated sentences. This section describes an accurate way to incorporate semantic representations into an encoder-decoder video captioning model.

A. The Encoder

Our encoder is composed of two parts. The first part is a standard visual feature extractor that encodes the input x into a real-valued representation that we denote by v. We construct it by first sampling p frames from x and then computing 2D-CNN feature vectors $\{f_1, f_2, ..., f_p\}$ and 3D-CNN feature vectors $\{g_1, g_2, ..., g_p\}$ intuitively capturing appearance and motion features, respectively. These features are then concatenated and averaged to produce v, that is, $v = \frac{1}{p}\sum_{i=1}^{p}[f_i, g_i]$.

The second part, which we call *concept detector*, is borrowed from the work by Chen *et al*. [3] and Gan *et al*. [8]. We will briefly explain the main idea and refer to [3], [8] for details. Essentially, a neural network classifies the v values, determining the probability of each tag appears in the input video x. Since most video description datasets do not include annotations according to a set of tags, we annotate the features and define the multi-class classification approach as follows.

Let L the set of associated captions of x, and $y = [y_1, ..., y_K] \in \{0, 1\}^K$ the associated tag vector, where $y_j = 1$ if the j-th tag appears in, at least, one of the captions in L. This definition assumes all used words in the video captions are equally likely to appear in x. The cost function to be minimized is a component-wise binary cross-entropy loss as it is customary for multi-class classifiers:

$$L_{\Theta_1} = \sum_{i=1}^{K}(y_i \log s_i + (1 - y_i)\log(1 - s_i))$$

where

$$MLP_{\Theta_1}(v) = [s_1, ..., s_K]$$

and Θ_1 represents the parameters to be learned into a standard multi-layer perceptron (MLP) neural network that has v as input, relu activations in hidden layers, and a sigmoid activation at the output, producing a vector MLP$(v) \in [0, 1]^K$.

B. The Decoder

For our decoder, we proposed a new deep compositional neural network based on the existing SCN-LSTM and Chen *et al*. models [3], [8], which is capable of learning to decide when to use visual-related information or visual-semantic-related information in the sentence generation process. The SCN-LSTM and Chen *et al*. models suffer a strong dependence on the quality of concepts-detection models.This high dependence produces the decoder fails in generating some words that could be easy to predict without using explicit semantic information, *e.g*., the word "be" and the colors.

In contrast, our decoder can selectively decide when to use the visual-dependent information or visual-semantic-dependent information at each step of the sentence generation process. Our model manages this automatic selection of the relevant information maintaining two specialized channels, conditioning the semantic channel by a temporal attention mechanism, and incorporating an adaptive attention mechanism able to decide when to use each channel's output dynamically. On the one hand, the temporal attention makes the model selectively focus on the video's relevant temporal fragments, guiding what frames to look. On the other hand, the adaptive attention works as a fusion gate that decides when to use the visual information from the bottom LSTM layer and when to use the semantic context information from the top LSTM layer.

These two channels are performed by two recurrent layers, which are specialized in composing a different kind of global information of the video, *i.e*., visual (v) and semantic (MLP(v)). An advantage of using these specialized layers is to capture temporal states related to each feature representation separately. Moreover, our attention mechanism allows the model to dynamically ensemble these two kinds of feature information. Specifically, at the bottom layer, we effectively decode visual feature information. While, at the top layer, the model incorporates the semantic information and focuses on the meaning of the visual and language context information. We define the hidden states of each specialized layer S-LSTM$(h_{t-1}, c_{t-1}, x_t, q, r)$ as follows.

We first define a general operator $F(\cdot, \cdot, \cdot, \cdot, \cdot)$ over matrices and vectors as:

$$F(U, V, W, x, y) = W \cdot ((U \cdot x) \odot (V \cdot y)).$$

Now let $*$ represents i (input), f (forget), o (output) and c (cell) gates. Then we define the intermediate vectors $\hat{m}_*$, $\hat{x}_{*,t}$ and $\hat{h}_{*,t-1}$ as

$$\hat{m}_* = F(C_{*,1}, C_{*,2}, C_{*,3}, q, r) \quad (1)$$

$$\hat{x}_{*,t} = F(W_{*,1}, W_{*,2}, W_{*,3}, x_t, r) \quad (2)$$

$$\hat{h}_{*,t-1} = F(U_{*,1}, U_{*,2}, U_{*,3}, h_{t-1}, r) \quad (3)$$

where $W_{*,j}$, $U_{*,j}$ and $C_{*,j}$ with $j \in \{1, 2, 3\}$ are weight matrices to be learned for each gate. Given these intermediate vectors, we can define the (hat) gates $\hat{i}_t$, $\hat{f}_t$, $\hat{o}_t$ and $\hat{c}_t$ as

$$\begin{aligned}
\hat{i}_t &= \sigma(\hat{x}_{i,t} + \hat{m}_i + \hat{h}_{i,t-1} + b_i),\\
\hat{f}_t &= \sigma(\hat{x}_{f,t} + \hat{m}_f + \hat{h}_{f,t-1} + b_f),\\
\hat{o}_t &= \sigma(\hat{x}_{o,t} + \hat{m}_o + \hat{h}_{o,t-1} + b_o),\\
\hat{c}_t &= \tanh(\hat{x}_{c,t} + \hat{m}_c + \hat{h}_{c,t-1} + b_c),
\end{aligned}$$

where b_* with $* \in \{i, f, o, c\}$ is a different bias vector to be learned for each gate. Finally, we have

$$\text{S-LSTM}(h_{t-1}, c_{t-1}, x_t, q, r) = [h_t, c_t],$$

such that

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

1) Visual-dependent Layer: To define this layer (v-LSTM in Figure 1), we only incorporate, at each step, the visual information v of the video. To do this, we re-utilize our definition of S-LSTM unit, but not taking into account the intermediate representation $\hat{m}_*$ (Eq. 1) as part of the $\hat{*}_t$ gates computation. Then, the (hat) gates of this layer could be efficiently computed by

$$\hat{i}_t = \sigma(\hat{x}_{i,t} + \hat{h}_{i,t-1} + b_i),$$
$$\hat{f}_t = \sigma(\hat{x}_{f,t} + \hat{h}_{f,t-1} + b_f),$$
$$\hat{o}_t = \sigma(\hat{x}_{o,t} + \hat{h}_{o,t-1} + b_o),$$
$$\hat{c}_t = \tanh(\hat{x}_{c,t} + \hat{h}_{c,t-1} + b_c),$$

and the hidden and cell states can be defined by passing the visual representation v through the input parameter r of the S-LSTM as follows:

$$h_{v_t}, c_{v_t} = \text{S-LSTM}(h_{v_{t-1}}, c_{v_{t-1}}, x_t, 0, v),$$

where $h_{v_{t-1}}$ and $c_{v_{t-1}}$ are the previous hidden and cell states of the layer, respectively.

2) Temporal Attention: To obtain accurate visual-semantic-related information, we incorporate a component for temporal attention (TA in Figure 1) after the v-LSTM unit, and before the s-LSTM unit. Using the output of temporal attention as an input of the top semantic-dependent layer allows the model to effectively relate the semantic concepts to video's relevant fragments at each step. We adapted the soft attention [1] mechanism by learning to dynamically weight each temporal feature vector, taking the weighted sum of them.

3) Semantic-dependent Layer: Another specialized S-LSTM unit implements the channel to obtain the temporal semantic-dependent information. In contrast to v-LSTM, this layer (s-LSTM in Figure 1) incorporates the global semantic information, focusing on learning the visual and language context information. Specifically, this layer's procedure can be defined from our S-LSTM, introducing the global semantic representation of the video MLP(v) by the input parameter r and the output of TA a_t through the parameter q as follows:

$$h_{s_t}, c_{s_t} = \text{S-LSTM}(h_{s_{t-1}}, c_{s_{t-1}}, x_t, a_t, s),$$

where $h_{s_{t-1}}$ and $c_{s_{t-1}}$ are the previous hidden and cell states, respectively.

4) Adaptive Attention Gate: After the model computes the recurrent step of s-LSTM layer, the adaptive fusion of h_{v_t} and h_{s_t} determines the most accurate information to generate the word at step t. We divide the computation of our Adaptive Attention Gate (AAG in Figure 1) into two parts. Firstly, the unit computes the weights $\beta \in [0,1]^H$ from the concatenation of $\hat{a}_{t-1}$ and a_t, a fully-connected layer, and the sigmoid activation function (see Figure 2). These

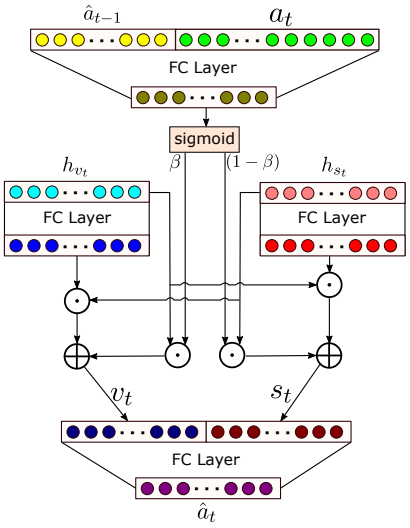

Fig. 2. Adaptive Attention Gate proposed to fuse the outputs of both S-LSTM units. Firstly, the concatenation of attention vectors $\hat{a}_{t-1}$ and a_t is passed through a fully-connected layer with the sigmoid activation function for computing the weights $\beta \in [0,1]^H$. Then, β is used in a cross-activation strategy that strengthens and merges the related information within the specialized hidden states h_{v_t} and h_{s_t}. We built this strategy by element-wise multiplications $\bigodot$ and additions $\bigoplus$.

weights are used to control how much information from each specialized channel should be preserved. A cross-activation strategy with residual connections reaffirms and merges the related information within the specialized hidden states h_{v_t} and h_{s_t}. Formally, we defined the unit by:

$$\beta = \sigma(W_{a,3} \cdot [\hat{a}_{t-1}, a_t] + b_{a,2}),$$
$$s_t = (h_{v_t} \cdot W_{a,4}) \odot h_{s_t} + \beta \odot h_{v_t},$$
$$v_t = (h_{s_t} \cdot W_{a,5}) \odot h_{v_t} + (1-\beta) \odot h_{s_t},$$
$$\hat{a}_t = W_{a,6} \cdot [s_t, v_t] + b_{a,3},$$

where $W_{a,3}$, $W_{a,4}$, $W_{a,5}$, $W_{a,6}$, $b_{a,2}$ and $b_{a,3}$ are parameters to be learned, β is the result of logistic sigmoid function $\sigma(\cdot)$, and $\odot$ is the element-wise multiplication.

Intuitively, this component learns which data is essential to keep (or disregard) generating the word in each step. Besides, this definition can be easily extended for mixing a higher number of S-LSTM layers by connecting some of these components into a cascade way.

5) Word Embedding: Finally, the model determines a probability distribution over vocabulary and choose one word in each step by a fully-connected layer and the softmax activation function. We mapped this word to a vector representation using a pre-trained word embedding (embedding in Figure 1), which is used as input of the two specialized layers at the next step.

C. Loss Function

As a language generation task, video captioning models are usually trained by minimizing the Cross-Entropy loss (CELoss) [11] function. With CELoss, the model is trained to maximize the probability of generating the next correct

word of the reference caption. However, at test time, the model has only access to its predictions, which may not be correct. Additionally, the popular evaluation metrics used for video captioning are based on n-gram overlapping between the generation and reference captions, *e.g.*, BLEU [23]. Some authors, like Ranzato *et al.* [28], study this discrepancy and board some techniques like *Beam Search* [1], [31], to overcome it.

In this work, we deal with this limitation by using a loss function that operates with explicit supervision at the sequence level. We adopt the loss function proposed by Chen *et al.* [3], which weights the CELoss according to the length of the reference captions. Given a ground-truth caption $y = \{w_1, w_2, \ldots, w_L\}$ of the input video, we minimize

$$\mathcal{L}_{\Theta_2} = -\frac{1}{L^\beta} \sum_{t=1}^{L} \log p_{\Theta_2}(w_t | w_{z<t}),$$

where Θ_2 represents all the parameters of the decoder to be learned. The hyperparameter $\beta \in [0, 1]$ regulates the length of the generated sentences. The higher the value of β, the lower the loss produced by the function, and vice versa. Thus, a value of β near 0 implies that the model rapidly adapts to generate concise sentences that could affect their syntactic correctness.

IV. Experimental Evaluation

In this section, we evaluate our video captioning method on two popular datasets. For comparing the new method with existing works, we computed the evaluation metrics with the codes released on Microsoft COCO evaluation server [4].

A. Datasets

In our experiments, we considered the standard splits of two widely used benchmarks that are publicly available: MSVD and MSR-VTT (see Table I for details).

B. Training Setup

First of all, we determine the vocabulary and the maximum length L of the descriptions in the each dataset's training set. For representing each word, we use the 300-dimensional GloVe [27] word embedding. This embedding was pre-trained on a 6 billion word corpus from Wikipedia 2014 + Gigaword 5[1]. For MSVD, 1,595 words of the training set were not part of the 400,000 vocabulary entries of GloVe dictionary, resulting in a vocabulary of 8,034 words. While for MSR-VTT, 4,532 words were missing, resulting in a vocabulary of 18,995 words. Then, we represent each description as a sequence (of length L) of indices in the vocabulary, putting the end-of-sentence token (EOS) from the end of each sentence to the length L. We initialize the weight matrices of the model by Gaussian initialization [10] strategy and the biases vectors by zero.

To encode each video, we sample a $p \leq 30$ frames from MSR-VTT's videos, and $p \leq 20$ frames from MSVD's videos. For extracting the 2D-CNN visual features from each sampled frame, we extract 2048-dimension vectors with ResNet-152 model [13], pre-trained on ImageNet[2] dataset. For 3D-CNN visual features, we process segments of 16 frames (adjacent to the sampled frame) with the ECO model [44], pre-trained on Kinetics-400 dataset. ECO produces 1536-dimension vectors.

For our semantic concepts detector, we semi-manually built a set of $K = 300$ key-words (tags). We extract these tags from the training sets, sorting all words according to the number of appearances and selecting the most frequent adjectives, nouns, and verbs, *e.g.*, *young*, *man*, *boy*, *playing* and *run*.

We set the hidden size of the v-LSTM and s-LSTM specialized layers to 1024. We used a batch size of 64, and the Adam optimizer with an initial learning rate of 4×10^{-5} for the MSR-VTT dataset and 2×10^{-5} for MSVD (decaying it every 20 epochs). We train for at least 70 epochs using early-stopping criteria of 10 epochs with any improvement for METEOR score on the validation set. Besides, similar to the loss used by Xiao *et al.* [38], we set $\beta = 0.7$ for the weighted-loss function parameter. We conduct experiments drawing a word at random from the output distribution and choosing the word with the highest probability For word generation at each step.

Additionally, we train the decoder with a scheduled sampling strategy for quickly and efficiently training the recurrent neural network. This strategy prevents the slow convergence, the model instability, and the inadequate skill [20], including a teacher-forcing-ratio parameter that determines the probability of using ground-truth words of the previous step as input in the current step. We decay this parameter from 0.96 until 0.6, using the next equation:

$$p(e) = \max\left(0.6, \frac{s_f}{s_f + \exp(e/s_f)}\right),$$

where e is the epoch index, and s_f is the convergence speed factor. In our experiments $s_f = 24$ shows the best results.

We fine-tuned all the hyperparameters on each dataset's validation sets and selected the best checkpoint for testing according to METEOR score. During testing, we generate words until the decoder generates the EOS token. At each step, the model selects w_t by determining the index $argmax_\theta P(w_t | w_{<t}, V)$ in the vocabulary.

We implemented our model and training method in PyTorch [26] and publicly available[3].

C. Ablation Study

To validate the effectiveness of our model's different parts, in Table II, we compare the performance of different modifications of our proposal. The modifications consist of removing one component of the model, *i.e.*, v-LSTM layer, s-LSTM layer, TA, and AAG, or changing the training strategy, *i.e.*, the word sampling strategy or loss function. We ran six experiments:

- **without v-LSTM layer**. This experiment evaluates the v-LSTM layer's contribution to our proposal's performance.
- **without s-LSTM layer**. We evaluate the performance of the model removing the vital s-LSTM layer.

[1]Gigaword 5 corpus website: https://catalog.ldc.upenn.edu/LDC2011T07

[2]ImageNet dataset website: http://www.image-net.org/

[3]https://github.com/jssprz/attentive-specialized-network-video-captioning

TABLE I

SPLITS OF MSVD AND MSR-VTT DATASETS. THE AVG. LENGTH COLUMN SHOWS THE AVERAGE DURATION OF VIDEOS IN EACH DATASET

Dataset	Training set			Validation set		Testing set		Avg. length
	clips	sentences	unique words	clips	sentences	clips	sentences	
MSVD [19]	1,200	48,779	9,629	100	4,291	670	27,768	10.2s
MSR-VTT [39]	6,512	130,260	23,527	498	9,940	2,990	59,800	14.8s

TABLE II

ABLATION STUDY ON THE TESTING SET OF MSVD AND MSR-VTT DATASETS. IN EACH ROW A COMPONENT OF OUR DECODER WAS REMOVED.

Method	MSVD				MSR-VTT			
	BLEU-4	METEOR	CIDEr	ROUGE$_L$	BLEU-4	METEOR	CIDEr	ROUGE$_L$
AVSSN (ours)	**62.3**	**39.2**	**107.7**	78.3	**45.5**	**31.4**	**50.6**	**64.3**
-wo. v-LSTM layer	60.4	37.3	95.7	74.7	43.9	21.8	46.6	62.7
-wo. s-LSTM layer	52.6	36.0	88.2	73.0	42.8	27.3	42.5	60.4
-wo. S-LSTM layers	54.3	33.9	92.8	73.8	43.0	27.0	43.4	61.0
-wo. TA mechanism	58.7	38.3	104.9	75.7	44.4	28.4	44.6	61.1
-wo. AAG component	55.0	37.7	99.4	74.4	44.3	30.2	48.6	63.0
-wo. weighted-loss	55.9	38.0	98.5	74.8	43.8	30.1	47.9	63.2
-wo. argmax	61.5	38.9	107.1	**79.2**	44.9	29.6	50.0	63.8

TABLE III

PERFORMANCE COMPARISON WITH THE STATE-OF-THE-ART APPROACHES ON THE TESTING SET OF THE MSVD DATASET. THE APPROACHES ARE SORTED BY YEAR.

Approach	BLEU-4	METEOR	CIDEr	ROUGE$_L$
SCN-LSTM [8]	51.1	33.5	77.7	-
TDDF [42]	45.8	33.3	73.0	69.7
MTVC [24]	54.5	36.0	92.4	72.8
BAE [2]	42.5	32.4	63.5	-
ECO [44]	53.5	35.0	85.8	-
SibNet [21]	54.2	34.8	88.2	71.7
J-VisualPOS [14]	52.8	36.1	87.8	71.5
GFN-POS_RL [35]	53.9	34.9	91.0	72.1
hLSTMat [9]	54.3	33.9	73.8	-
SAVCSS [3]	61.8	37.8	103.0	76.8
AVSSN (ours)	**62.3**	**39.2**	**107.7**	**78.3**

TABLE IV

PERFORMANCE COMPARISON WITH THE STATE-OF-THE-ART APPROACHES ON THE TESTING SET OF THE MSR-VTT DATASET. THE APPROACHES ARE SORTED BY YEAR. * DENOTES RESULTS THAT WERE OBTAINED BY REINFORCEMENT LEARNING OF THAT METRIC.

Approach	BLEU-4	METEOR	CIDEr	ROUGE$_L$
TDDF [42]	37.3	27.8	43.8	59.2
MTVC [24]	40.8	28.8	47.1	60.2
CIDEnt [25]	40.5	28.4	51.7*	61.4
HRL [37]	41.3	28.7	48.8*	61.7
PickNet [5]	38.9	27.2	42.1	59.5
SibNet [21]	40.9	27.5	47.5	60.2
J-VisualPOS [14]	42.3	29.7	49.1	62.8
GFN-POS_RL [35]	41.3	28.7	53.4*	62.1
hLSTMat [9]	39.7	27.0	43.4	-
SAVCSS [3]	43.8	28.9	51.4*	62.4
AVSSN (ours)	**45.5**	**31.4**	**50.6**	**64.3**

- **without S-LSTM layers**. We evaluate the model using two standard LSTM layers instead of our S-LSTM layers. Thus, compositional operations are not used.
- **without TA mechanism**. This experiment evaluates how much the TA between the two specialized layers contributes to our model's effectiveness.
- **without AAG component**. We evaluate the model performance replacing the AAG with a fully-connected layer.
- **without weighted-loss**. This experiment shows the importance of training the model using our length-weighted-loss function. We train the model using the cross-entropy loss.
- **without argmax sampling**. This experiment evaluates sampling the word according to the output multinomial probability distribution, making higher probabilities more likely. During testing, we choose the argmax probability.

1) Comparison with the state of the art on MSVD: Table III shows the performance of several proposed models in the literature on MSVD. All these approaches are encoder-decoder frameworks based on deep learning and that have been proposed since 2017. Some of them use high-level semantic representations, *e.g.*, SCN-LSTM [8] and SAVCSS [3]. Moreover, others like hLSTMat [9] use an attention mechanism to integrate basic LSTM layers. From this table, we can infer our model outperforms the state-of-the-art results on all metrics. Specifically, we surpass the other methods by at least 3.7% relatively in terms of METEOR, and at least 4.5% relatively in terms of CIDEr. In terms of ROUGE$_L$, although our model improves the existing methods by more than 1.9% relatively, the ablation experiment *wo. argmax* gets an even better score.

2) Comparison with the state of the art on MSR-VTT: Table IV compares our AVSSN architecture with the state-of-the-art methods on the MSR-VTT dataset. This table shows that our method outperforms the best on all metrics except CIDEr. Specifically, we obtain a relative improvement of 2.7%, 5.7% and 2.3% for BLEU-4, METEOR and ROUGE$_L$, respectively. While for CIDEr, the models that obtain better results directly maximized this metric by reinforcement learning. However, compared to unreinforced methods, we improve CIDEr by at least 3.0% relatively. For instance, we improve the hierarchic LSTM-based architecture of hLSTMat [9] by 16.6%. We can also notice that some of our ablation experiments also get state-of-the-art results in terms of BLEU-4, *e.g.*, *wo. TA mechanism*.

3) Qualitative Analysis: Figure 3 presents the predictions of our model for three different video examples of the MSVD dataset. To observe the improvement in the captions generated by our model, we compared these predictions with the outputs

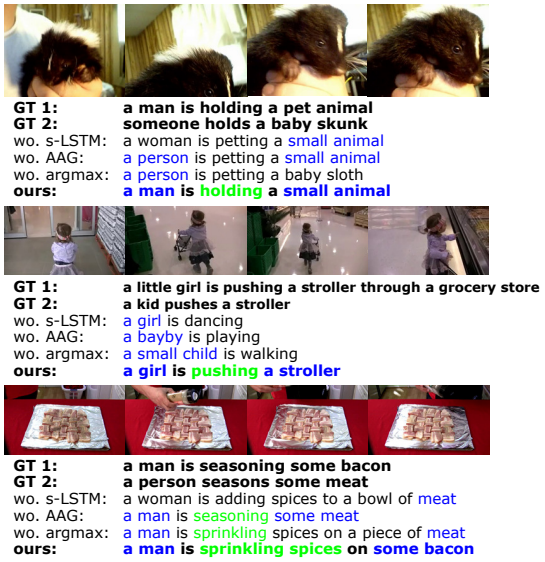

Fig. 3. Three representative samples from the test split of MSVD, which cover ground-truth captions, three of our ablation models, and our proposal. In blue and green, semantic concepts that the model predicted correctly.

of three of our ablation models, *i.e.*, *wo. s-LSTM layer*, *wo. AAG component* and *wo. argmax*. We highlighted in green and blue the verbs, adjectives and nouns where the model could decide when to use the semantic information correctly. In these three video examples, we can notice that the model could generate better descriptions than the other models:

- In the first example, our proposal was the only one that could predict the action "holding".
- In the second example, our model was the only one that could predict the verb "pushing" and the noun "stroller".
- In the third example, our model was the only one that could predict the noun "bacon".

However, there is still much work to do for video captioning. Current models cannot capture what happens in elaborate videos with multiples events, *e.g.*, in the second video, all models failed to predict the child was in a "grocery store".

V. Conclusions

In this paper, we presented the Attentive Visual Semantic Specialized Network (AVSSN). This model explores both visual representations and high-level semantic representations for video captioning, and can selectively decide which of them is more important for predicting each word. These representations are mainly processed simultaneously by two different specialized layers (S-LSTM) and fused by a novel Adaptive Attention Gate. This adaptive gate effectively determines the essential information to keep or disregard for generating the word in each step. In addition, the model can be easily extended to a higher number of S-LSTM layers for new video representations, *e.g.*, syntax. We plan to assess our approach to other video understanding tasks besides video captioning, such as text retrieval from videos. The quantitative and qualitative experiments we performed shown that the temporal representations offered by the S-LSTMs and the fusion strategy help the model to generate more representative descriptions. Our model achieves high results, which are superior to the state-of-the-art results on the MSVD and MSR-VTT datasets.

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] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] Xavier Glorot, Xavier Glorot, and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *International Conference on Artificial Intelligence and Statics*, 2010.

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

[12] 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.

[13] 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.

[14] 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.

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

[16] 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.

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

[18] 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.

[19] 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.

[20] Alex Lamb, Anirudh Goyal, Ying Zhang, Saizheng Zhang, Aaron Courville, and Yoshua Bengio. Professor Forcing: A New Algorithm for Training Recurrent Networks. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, page 4608–4616, Barcelona, Spain, 2016.

[21] 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.

[22] 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.

[23] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics - ACL '02*, number July in ACL '02, page 311, Morristown, NJ, USA, 2002. Association for Computational Linguistics.

[24] 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.

[25] 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.

[26] 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.

[27] Jeffrey Pennington, Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. *IN EMNLP*, 2014.

[28] 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.

[29] 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.

[30] Zhiqiang Shen, Jianguo Li, Zhou Su, Minjun Li, Yurong Chen, Yu-Gang Jiang, Xiangyang Xue, and † Shanghai. Weakly Supervised Dense Video Captioning. Technical report, 2017.

[31] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. *Advances in Neural Information Processing Systems (NIPS)*, pages 3104–3112, 2014.

[32] 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.

[33] Subhashini Venugopalan, Huijuan Xu, Jeff Donahue, Marcus Rohrbach, Raymond Mooney, and Kate Saenko. Translating Videos to Natural Language Using Deep Recurrent Neural Networks. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, number June, pages 1494–1504, Stroudsburg, PA, USA, 2015. Association for Computational Linguistics.

[34] 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.

[35] 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.

[36] Jingwen Wang, Wenhao Jiang, Lin Ma, Wei Liu, and Yong Xu. Bidirectional Attentive Fusion with Context Gating for Dense Video Captioning. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7190–7198. IEEE, 6 2018.

[37] 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.

[38] Huanhou Xiao and Jinglun Shi. A Novel Attribute Selection Mechanism for Video Captioning. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 619–623. IEEE, 9 2019.

[39] 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.

[40] Li Yao, Atousa Torabi, Kyunghyun Cho, Nicolas Ballas, Christopher Pal, Hugo Larochelle, and Aaron Courville. Describing Videos by Exploiting Temporal Structure. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 4507–4515. IEEE, 12 2015.

[41] 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.

[42] 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.

[43] Luowei Zhou, Yingbo Zhou, Jason J. Corso, Richard Socher, and Caiming Xiong. End-to-End Dense Video Captioning with Masked Transformer. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8739–8748. IEEE, 6 2018.

[44] 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.