Full-GRU Natural Language Video Description for Service Robotics Applications

Silvia Cascianelli, Gabriele Costante, Thomas A. Ciarfuglia, Paolo Valigi, and Mario L. Fravolini

***Abstract*—Enabling effective human–robot interaction is crucial for any service robotics application. In this context, a fundamental aspect is the development of a user-friendly human–robot interface, such as a natural language interface. In this letter, we investigate the robot side of the interface, in particular the ability to generate natural language descriptions for the scene it observes. We achieve this capability via a deep recurrent neural network architecture completely based on the gated recurrent unit paradigm. The robot is able to generate complete sentences describing the scene, dealing with the hierarchical nature of the temporal information contained in image sequences. The proposed approach has fewer parameters than previous state-of-the-art architectures, thus it is faster to train and smaller in memory occupancy. These benefits do not affect the prediction performance. In fact, we show that our method outperforms or is comparable to previous approaches in terms of quantitative metrics and qualitative evaluation when tested on benchmark publicly available datasets and on a new dataset we introduce in this letter.**

***Index Terms*—Cognitive human-robot interaction, visual learning.**

I. Introduction

THE ability to provide a description of the scene in a form that every user can easily understand is keystone for the success of effective and user-friendly service robotics products. In fact, a natural language description offers an interpretable manifestation of the robot's inner representation of the scene and is also a good basis for natural language question answering about what is happening in the environment. Hence, this functionality would provide a friendly interface also for non-expert people who would then be able to easily interact with their home robot in the near future.

In the sight of this, this work addresses the problem of describing a scene in natural language, which is usually referred to as Natural Language Video Description (NLVD). Here we formalize this problem as a Machine Translation (MT) one, from "visual language" to English. Basically, the information in form of a varying length video sequence is encoded in a fixed-length vector and then decoded in form of varying length English sentence (Fig. 1).

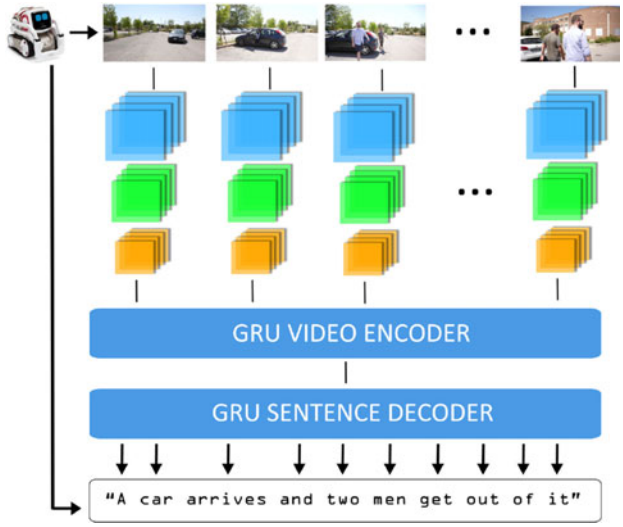

Fig. 1. Overview of the proposed NLVD system. The robot observes a generic and complex scene and represents it taking into account both the visual and temporal information, represented via ConvNet features and an encoding vector, respectively. Then, it outputs a natural language sentence describing the observed scene. The proposed encoder-decoder scheme is entirely based on GRU recurrent units.

The video translation is performed via D-RNNs, i.e., recurrent models that are able to deal with both long and short term dependencies in data sequences. Most of the previous approaches rely on the Long Short-Term Memory (LSTM) [1] architecture. Recently, Generalized Recurrent Unit (GRU) [2] were proposed as a simplification of LSTM units. Their performance is similar to LSTMs', but with fewer parameters to train. This makes the recurrent networks based on GRU faster to train and less prone to overfitting. Saving training time in any deep learning application is critical when tuning hyper parameters for field application, such as robotics. For this reason, in this letter we explore the performances of a NLVD system completely based on the GRU paradigm, comparing it to State-of-the-Art approaches that exploit LSTMs.

In this work, a full-GRU NLVD system is proposed, that is able to deal with the hierarchical nature of the temporal information typical of natural and generic video sequences and obtains comparable performance with respect to more complex State-of-the-Art (SotA) systems. The proposed system features a GRU cell modified in order to automatically change its temporal connection if a boundary, i.e., a significant modification in the

Manuscript received September 10, 2017; accepted December 29, 2017. Date of publication January 15, 2018; date of current version January 25, 2018. This letter was recommended for publication by Associate Editor S. Rossi and Editor D. Lee upon evaluation of the reviewers' comments. This work was supported by the NVIDIA Corporation with the donation of the Titan Xp GPU. *(Corresponding author: Silvia Cascianelli.)*

The authors are with the Department of Engineering, University of Perugia, Perugia 06123, Italy (e-mail: silvia.cascianelli@studenti.unipg.it; gabriele.costante@unipg.it; thomas.ciarfuglia@unipg.it; paolo.valigi@unipg.it; mario.fravolini@unipg.it).

Digital Object Identifier 10.1109/LRA.2018.2793345

scene, is detected. To the best of our knowledge, this is the first full-GRU encoder-decoder architecture applied to the problem of NLVD. In addition, a new small dataset for NLVD in typical service robotics scenarios is used, which offers a fair test bench for the specific application we target. The relevance of this dataset, is twofold. First, this is the first dataset specifically collected in typical applicative contexts of a service robot. Second, it helps to get insights on the actual performance of SotA NLVD models we are testing. Indeed, these systems are commonly trained and tested on videos from the same datasets, which may make their evaluation biased. The experiments on our dataset makes this more evident.

To summarize, the main contributions of this work are:

- We propose an improved architecture for NLVD that is based on GRU units, to save training time without impairing the performances.
- We perform an experimental evaluation of our method with other SotA approaches. The experiments show that this method obtains comparable performances with SotA methods that harness LSTM.
- We present a dataset that features a wide range of contexts that are typical for service robotics applications.

The remainder of this letter is organized as follows. In Section III the proposed approach is described. Section IV provides a detailed description of the experimental results and conclusion are drawn in Section V.

II. Related Work

In recent years, many researchers from both Computer Vision and Natural Language Processing communities are studying the problem of describing generic videos using natural language phrases (see e.g., [3], [4]).

Some popular approaches [4], [5] are based on filling-in predefined template sentences with the subject-verb-object concepts detected in the video. In particular, an object detector (e.g., a CNN as in [5]) is used to recognize the main actors in the video and a Probabilistic Graphical Model (PGM) (e.g., an Hidden Markov Model as in [4]) is used to predict the relation between them. These approaches have major limitations. First, the type and the number of the objects and the relations that can be described are limited to those that the detector and the PGM can estimate. Second, the output descriptions lack in diversity and naturalness.

Other works [6] propose to tackle the NLVD task in a multimodal retrieval fashion. In particular, given a corpus of paired videos and text, the system describes a new video using the sentence associated to the most similar video in the corpus [6]. Also this approach has some weaknesses. In particular, the system is constrained to use the same sentences in the corpus, which may be not semantically relevant for the new scene to describe.

Among the proposed strategies, treating the NLVD problem as a Machine Translation (MT) one gained popularity [7] and D-RNN demonstrated to be a very promising instrument [8]–[10]. This is particularly true when recurrent models are combined with State-of-the-Art Convolutional Neural Networks (ConvNet), even pre-trained.

Despite of the success of recent State-of-the-Art approaches, NLVD is still a particularly challenging problem, firstly due to the "object" of the description itself, i.e., the video sequence, that is typically open-domain and complex in real scenarios. In particular, the content of the videos can be highly diverse and the temporal dependencies between the depicted events can be at different granularity. Some architectures exist that produce accurate descriptions of videos, but in general these are either very short or very specific, or both, i.e., they depict simple activities of a particular domain with few "actors" in the scene [5], [9]. Those kinds of video sequences are far simpler than the typical complexity that a robot faces in real application contexts. The systems presented in [8] and [10] deal with generic and complex videos. Both of them represent the video sequence by mean-pooling the ConvNet features extracted from each frame, then decode the sentence with a LSTM-based decoder. A major drawback of those strategies is that they do not take into account the temporal structure of the video sequences due to mean-pooling.

Indeed, when considering more complex and generic video sequences it is crucial to deal with temporal dependencies at different granularity. This is done in [11], [12] and also in this work, where a hierarchical representation of the temporal information is explicitly learned. In [11] the authors draw from ConvNets the idea of convolutional operations and build a multi-level LSTM-based encoding able to capture longer time dependencies between the content of the frames. Then, a LSTM decoder produces the description exploiting an attention mechanism (that is basically a learned weighting strategy). The work of [12] is the most similar to our work. It presents a LSTM-based decoder that contains a boundary-aware LSTM cell. This cell and a second layer LSTM block build an encoding of the video sequence which is then decoded via a GRU. All of the above approaches, either consists of full-stack LSTM architectures or limit the use of the GRU to the decoding phase. In this letter, we present an encoder-decoder architecture that is completely based on GRU blocks, which have fewer parameters than LSTM, thus resulting arguably more suitable for robotics applications. This is motivated also by the study reported in [13], that compares the GRU and the LSTM cells on various tasks. Using input, state and output vectors of the same dimensionality, the GRU outperforms or is comparable to the LSTM in terms of convergence time, parameters update and generalization.

III. Encoder-Decoder Full-GRU Architecture

In this section our proposed model is presented. The video frames are described via the *ResNet50* and the *C3D* ConvNets (see Section III-A). The obtained feature vectors are then fed, one at each time-step, in the first layer of the encoder. This is our proposed BA-GRU recurrent block, that encodes the video frames until a boundary is detected. Afterwards, the first-layer encoding is fed to the second layer of the encoder, which consists of a classical GRU block (see Section III-B). The output of the encoding phase is a vector representing the entire video sequence. Finally, the GRU decoder produces the description emitting the most probable word at each time-step, conditioned to the video vector representation and the previous emitted

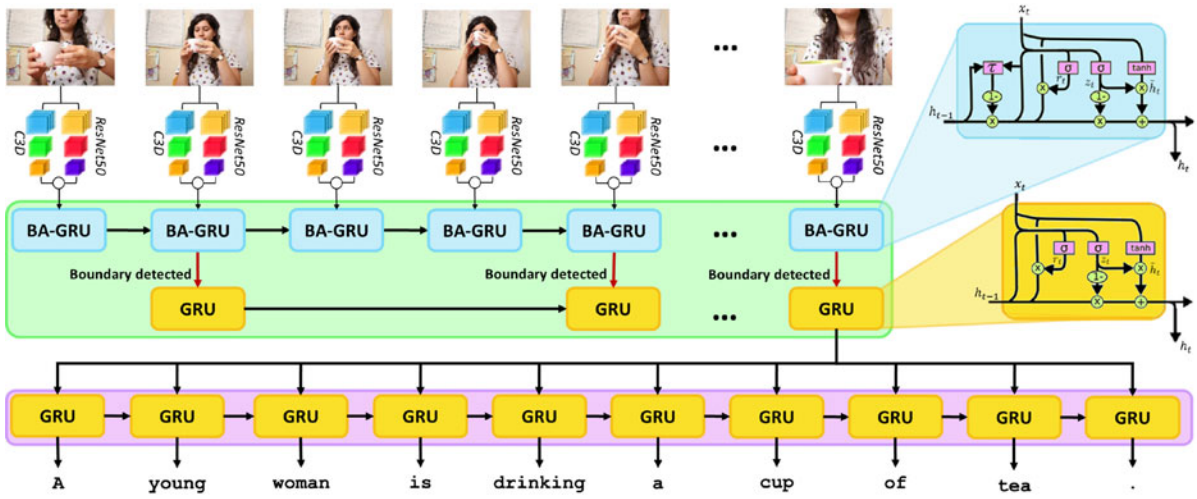

Fig. 2. Architecture of the proposed system. Recurrent layers are depicted as unfolded graphs for explanatory purpose.

words (see Section III-C). The captioning process ends when a <EOS> tag (i.e., the full-stop) is emitted. A pictorial representation of the system is shown in Fig. 2.

A. Video Frames and Caption Words Preprocessing

The video frames are preprocessed as follows. The output of the last fully connected layer of the *ResNet50* ConvNet [14] is computed every five video frames, to capture the appearance of the scene. To the same video frames is associated also the output of the *C3D* ConvNet [15] to capture the movement in the scene, based on partially overlapped sliding windows of frames. The output of the two ConvNets are concatenated (forming a 2048+4096-dimensional vector) and mapped in a learned 512-dimensional linear embedding. The entire video is then represented by a sequence of features vectors $(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n)$, where the $x.$ vectors are the feature vectors extracted from the frames of the video.

The captions are preprocessed as follows. First, the words are converted to lower-case and the punctuation characters are removed. Then, begin-of-sentence (<BOS>) and end-of-sentence (<EOS>) tags are added before and behind the sentence, respectively. Finally, the sentences are tokenized. From the tokenized sentences, we build a vocabulary (D). To prevent the formation of a large vocabulary containing many rare words, we retain only those tokens that appear at least five times in the caption corpus. To each token is associated an index in the vocabulary, based on its frequency in the vocabulary. A caption is then represented by a list of one-hot vectors $(\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_L)$, each of them corresponding to the representation of its words in the vocabulary. Similarly to what is done for the frames features, the captions are mapped in a learned 512-dimensional linear embedding.

B. Video Encoder

In this work, we build upon the boundary-aware LSTM (BA-LSTM) cell presented in [12] and devise a boundary-aware GRU (BA-GRU) cell. This cell is the first layer of a two-layers encoder. The second layer of the encoder is a simple GRU cell [2].

The BA-GRU is a modification of the classical GRU cell (see Fig. 2, top right). The GRU is a recurrent neural networks with gating strategies to model wider temporal dependencies in the input sequence. The GRU is characterized by an update gate $\mathbf{z}_t$ and a reset gate $\mathbf{r}_t$. At each timestep, a candidate activation $\tilde{\mathbf{h}}_t$ is computed based on the current input $\mathbf{x}_t$, the previous inner state $\mathbf{h}_{t-1}$ and the values of the gates. In particular, the $\mathbf{z}_t$ gate controls how much the inner state $\mathbf{h}_t$ has to be updated, the $\mathbf{r}_t$ gate controls how much the previous inner state $\mathbf{h}_{t-1}$ influences the candidate inner state value $\tilde{\mathbf{h}}_t$. More formally, the GRU is defined by the following equations:

$$\mathbf{h}_t = (1 - \mathbf{z}_t)\mathbf{h}_{t-1} + \mathbf{z}_t \tilde{\mathbf{h}}_t \tag{1}$$

$$\tilde{\mathbf{h}}_t = \tanh\left(W_{hx}\mathbf{x}_t + W_{hh}(\mathbf{r}_t \odot \mathbf{h}_{t-1}) + \mathbf{b}_h\right) \tag{2}$$

$$\mathbf{r}_t = \sigma(W_{rx}\mathbf{x}_t + W_{rh}\mathbf{h}_{t-1} + \mathbf{b}_r) \tag{3}$$

$$\mathbf{z}_t = \sigma(W_{zx}\mathbf{x}_t + W_{zh}\mathbf{h}_{t-1} + \mathbf{b}_z) \tag{4}$$

where the W_{**}s and $\mathbf{b}_*$s are learnable weight matrices and bias vectors, σ is the sigmoid function, $\tanh$ is the hyperbolic tangent function and $\odot$ is the element-wise product.

In this work, we modify the GRU by adding a boundary aware gate s_t, that modifies the inner connectivity of the unit based on the input and the inner state. In particular, when a substantial change in input sequence occurs, a boundary is estimated by a learnable function. Consequently, the inner state $\mathbf{h}_{t-1}$ is emitted as output (we denote it as $\mathbf{h}_k^{e1} \doteq \mathbf{h}_{t-1}$) and then re-initialized to zero according to:

$$\mathbf{h}_{t-1} \leftarrow \mathbf{h}_{t-1}(1 - s_t) \tag{5}$$

The boundary-aware gate is defined as follows:

$$s_t = \tau\left(\mathbf{w}_s^T(W_{sx}\mathbf{x}_t + W_{sh}\mathbf{h}_{t-1} + \mathbf{b}_s)\right) \tag{6}$$

where W_{**}s and $\mathbf{b}_s$ are learnable weights matrices and bias vectors. In this study, we set to 128 the number of their rows. The row vector $\mathbf{w}_s^T$ makes the input to the $\tau(\cdot)$ function a scalar. The $\tau(\cdot)$ function is given by:

$$\tau(\cdot) = \begin{cases} 1 & \text{if } \sigma(\cdot) < 0.5 \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

The given output $\mathbf{h}_k^{e1}$ summarizes the video substream before the boundary, which is then composed by homogeneous frames. For an input video, the BA-GRU block outputs as many vectors $\mathbf{h}_k^{e1}$, as the number of detected boundaries $(\mathbf{h}_1^{e1}, \mathbf{h}_2^{e1}, ..., \mathbf{h}_m^{e1})$, with $m \leq n$. Those vectors are given in input to the second layer of the encoder, which is a standard GRU block. This layer encodes the $\mathbf{h}_k^{e1}$ vectors in a unique vector $\mathbf{v}$, that represents the entire video. The $\mathbf{v}$ vector, that is the final output of the two-layer encoder, is fed to the decoder.

1) The Boundary-Aware Gate Training Details: The output s_t of the boundary-aware gate can be either 0 or 1, depending on the value of a sigmoid function applied to the input of the gate. Thus, following the approach of [12], in the training phase we model it as a stochastic binary neuron and learned its weights, while in test phase we use it with the learned weights as the deterministic neuron defined in (7). In particular, we re-write the activation function $\tau(\cdot)$ as:

$$\tau(\cdot) = \mathbf{1}_{\sigma(\cdot)>z}, \qquad z \sim \mathcal{U}(0,1) \tag{8}$$

where $\mathbf{1}_{\cdot}$ is the indicator function and $\mathcal{U}(0,1)$ denotes the uniform distribution between 0 and 1.

Note that $\tau(\cdot)$ in (7) is basically the composition of a step function and a sigmoid function. Thus, its derivative is equal to 0 everywhere except in 0, i.e., it is not continuous and smooth and it is also mostly flat. Hence, we cannot apply the standard back-propagation to compute the gradient in this gate. To overcome this issue, we follow the same approach of [12], that estimated the gradient by approximating the step function $\tau(\cdot)$ as the identity function [16]. The derivative of $\tau(\cdot)$ then becomes:

$$\frac{\partial \tau}{\partial(\cdot)}(\cdot) = \frac{\partial \sigma}{\partial(\cdot)}(\cdot) = \sigma(\cdot)(1-\sigma(\cdot)) \tag{9}$$

In the test phase, we use the deterministic form of $\tau(\cdot)$ (7), the parameters of which have been learned in the training phase using (8) (in the forward pass) and (9) (in the backward pass).

C. Caption Decoder

The decoder takes as input the video representation $\mathbf{v}$ and the ground truth sentence $(\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_L)$. At each timestep, it outputs a word $\mathbf{y}_l$ that is the most probable next word of the description, given the previous output words and the video representation.

To handle both the time-varying input $(\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_L)$ and the constant input $\mathbf{v}$, we modify (2)–(4) from the original GRU formulation as:

$$\tilde{\mathbf{h}}_t = \tanh\left(W_{hy} W_w \mathbf{y}_t + W_{hv}\mathbf{v} + W_{hh}(\mathbf{r}_t \odot \mathbf{h}_{t-1}) + \mathbf{b}_h\right) \tag{10}$$

$$\mathbf{r}_t = \sigma(W_{ry} W_w \mathbf{y}_t + W_{rv}\mathbf{v} + W_{rh}\mathbf{h}_{t-1} + \mathbf{b}_r) \tag{11}$$

$$\mathbf{z}_t = \sigma(W_{zy} W_w \mathbf{y}_t + W_{zv}\mathbf{v} + W_{zh}\mathbf{h}_{t-1} + \mathbf{b}_z) \tag{12}$$

where the W_{**}s and $\mathbf{b}_*$s are learnable weight matrices and bias vectors respectively, σ is the sigmoid function and $\odot$ is the element-wise product. The matrix W_w maps the input one-hot vectors representing the words $\mathbf{y}_l$ in the vocabulary space in a lower dimensional space (512-dimensional embedding). The output of the decoder (which we denote $\mathbf{h}_l^d \doteq \mathbf{h}_l$) is then mapped back in the original higher dimensional space as $\mathbf{y}_l = W_p \mathbf{h}_l^d$.

The probability of the next word in the description is modelled via the softmax function, i.e.,

$$Pr(\mathbf{y}_l|\mathbf{y}_0, \mathbf{y}_1, ..., \mathbf{y}_{l-1}, \mathbf{v}) \sim \frac{e^{\mathbf{y}_l^T W_p \mathbf{h}_l^d}}{\sum_{\mathbf{y}\in D} e^{\mathbf{y}^T W_p \mathbf{h}_l^d}} \tag{13}$$

Finally, the objective function to optimize is the log-likelihood of the correct words over the sentence i.e.,

$$\max_{\mathbf{W}} \sum_{l=1}^{L} \log Pr(\mathbf{y}_l|\mathbf{y}_0, \mathbf{y}_1, ..., \mathbf{y}_{l-1}, \mathbf{v}) \tag{14}$$

where $\mathbf{W}$ denotes all the parameters of the model.

IV. Experiments and Results

In this section, we present the experimental setup and the obtained results of our method.

A. Datasets Details

We employ two publicly available large datasets that are commonly used to study the NLVD problem. In addition, we test on a smaller dataset that we collected to be representative of daily activities that are typical of service robotics scenarios.

1) Max Plank Institute for Informatics Movie Description Dataset (MPII-MD): This dataset [17] contains over 68000 clips of average 4 s each, from a corpus of 94 HD movie of different genres. Those clips are associated with sentences taken from the movie script and the transcribed Descriptive Video Service (DVS[1]) track. As a common practice, we use the training/validation/test split provided by the authors of the dataset, resulting in 56816 training clips, 4930 validation clips and 6584 test clips. This split is the same typically used for NLVD systems [3], [7], [11], [12], [18], [19]. The vocabulary is obtained from the training corpus and consists of 7198 words.

2) The Microsoft Research Video Description Corpus (MSVD): This dataset [20] contains home-made 10–20 s long videos from YouTube. The topics of the videos include sports, animals and music. We retain the 1970 clips that have English captions associated. The captions are on average 43 for each video and have been collected by the Amazon Mechanical Turk service. As the common practice [7], [8], [10]–[12], [19], we use the first 1200 videos for training, the next 100 video for validation and the last 670 video for testing. Note that each video-caption pair is considered as a unique sample, so the actual number of samples in each split is average 43 times the number of videos. Again, we construct the vocabulary from the training set and obtain a vocabulary of 4215 words.

3) Intelligent Systems, Automation and Robotics Laboratory Video Description Dataset (ISARLab-VD): For this work, we collect a relatively small dataset. Despite that, our dataset is still generic in terms of depicted actions, environment and involved actors. Note that, none of the above datasets have been conceived

[1]Descriptive Video Service is an audio track associated to a movie to allow the visually impaired people to enjoy also the visual content of the movie.

for service robotics applications. This was a major motivation for us to produce the dataset. It contains 100 videos which length varies from 5 s to 30 s. Each video is paired with 5 manually obtained independent captions, for a total of 500 samples. The dataset features both high resolution and low resolution videos. In particular, the latter are obtained using the built-in camera of the COZMO toy robot by Anki[2] during the experimental phase of this study. In this work, we use the entire ISARLab-VD dataset for test only.

B. Evaluation Metrics Overview

In this work, we adopt classical natural language processing metrics for the evaluation of our method, which is a common practice in the NLVD research. These metrics are briefly described here for clarity and we refer to [21]–[24] for further details. First note that a *n*-gram is a sequence of *n* consecutive words. When comparing a candidate sequence X and a reference sequence Y, the *n*-gram recall is the proportion of *n*-grams in Y that appear also in X, while the *n*-gram precision is the proportion of *n*-grams in X that appear also in Y.

The first metric we use is BLEU [21], in its *4*-gram variant. It is a precision-oriented metric designed for MT evaluation. Basically, it combines the *n*-gram precision for each *n*-gram up to length 4 and penalizes the difference in length between the candidate and the reference sentences. BLEU correlates well with human judgement on the quality of the translation if evaluated on the entire test corpus, but its correlation at sentence level is poor.

We also adopt another MT evaluation metric, namely METEOR [22]. It combines unigram precision and recall based on matching unigrams in the candidate and reference sentences. Unigrams can be matched in their exact form, stemmed form, and meaning. METEOR correlates well with human judgement also at sentence level.

The third metric we use is ROUGE [23] in its variant ROUGE_L, that considers the Longest Common Subsequence (LCS) of the candidate and the reference sentence. ROUGE is a recall-oriented metric designed for summarization evaluation following the idea that a good candidate summary overlaps a reference summary. Note that all ROUGE variants correlate well with human judgement.

Finally, we adopt a recently developed metric for assessing image description quality capturing human consensus on it, namely CIDEr [24]. It is based on the average cosine similarity between *n*-grams of different order (up to *4*-grams) and rewards length similarity between candidate and reference sentences. Cosine similarity allows taking into account both precision and recall. This metric correlates well with human judgement by design, thus is particularly suitable for the task of NLVD.

C. Baseline Methods Overview

We quantitatively compare our system to some of the State-of-the-Art techniques presented in Section II, namely SA-GoogleNet+3D-ConvNet [19], S2VT [7], LSTM-YT [8], LSTM-E [10], HRNE [11] and BA-LSTM [12]. In addition, we compare to Venugopalan *et al.* [18] and to Rohrbach *et al.* [3]. SA-GoogleNet+3D-CNN applies an attention mechanism to select the most relevant video frames based on GoogLeNet [25] and 3D-CNN [26] extracted features, and an LSTM to generate the description sentence. S2VT uses a stacked LSTM encoder-decoder on the basis of ConvNet features extracted from each frame via VGG-16 [27]. LSTM-YT mean-pools each frame's AlexNet [28] ConvNet features and decodes this representation via a LSTM. LSTM-E learns an embedding based on the frame-level extracted mean-pooled VGG-19 [27] and C3D [15] ConvNet features and the video description, then generates a sentence via a LSTM. HRNE represents each video frame via GoogLeNet features and applies a hierarchical multi-layer LSTM encoder and a LSTM with soft-attention decoder. BA-LSTM is the most similar to our approach, but it uses LSTM blocks in the encoding phase. Venugopalan *et al.* [18] improves S2VT using a neural language model and distributional semantics learned from a large text corpus. Rohrbach *et al.* [3] uses CRFs to obtain tuples of verbs, objects and places on the basis of ConvNet features extracted from the video via pre-trained ConvNets, then translated the tuple into a sentence via a LSTM.

Differently from SA-GoogleNet+3D-CNN and HRNE, we do not apply any attention mechanism to deal with different-granularity time dependencies in the videos. As opposed to LSTM-YT and LSTM-E, we explicitly model the temporal dimension of the video sequence via the recurrent encoder. Finally, another major difference between our approach and the baselines is that we use a full-GRU architecture.

Note that, since BA-LSTM is the closest to our method, we used the same settings as the authors of [12] to better compare the two architectures. In particular, we set to 1024 the size of the inner state vectors and use the same size for input vectors, embeddings, weight matrices and bias vectors. Embedding matrices and weight matrices applied to inputs are initialized via the Glorot normal initializer, those applied to inner states are initialized via the orthogonal initializer and the bias vectors are initialized to zero. We perform the training until the validation loss stops improving (or up to 100 epochs), with mini-batch size of 128. As optimizer, we apply Adadelta with learning rate $l_r = 1.0$, decay constant $\rho = 0.95$ and parameter $\epsilon = 10^{-8}$. The input and the output of the BA-GRU and the GRU in the encoding phase are regularized via Dropout with retain probability $p = 0.5$.

D. Results on the Standard Datasets

The performance is evaluated on the MPII-MD and MSVD datasets and expressed in terms of the widely used metrics presented in Section IV-B. For consistency sake with the baselines, we use the original COCO evaluation script.[3]

The results are summarized in Table I for the MPII-MD dataset and in Table II for the MSVD dataset. It can be observed that our method is competitive with all the other approaches in terms of all the metrics. More importantly, it outperforms all the

[2]https://www.anki.com/en-us/cozmo

[3]https://github.com/tylin/coco-caption

TABLE I
EXPERIMENT RESULTS ON THE MPII-MD DATASET IN TERMS OF THE QUANTITATIVE EVALUATION METRICS BLEU IN ITS *4*-GRAM VARIANT (B_4), METEOR (M), ROUGE IN ITS LCS VARIANT (R_L) AND CIDER (C)

Model	B_4	M	R_L	C
SA-GoogleNet+3D-CNN [19]	–	5.7	–	–
S2VT-RGB [7]	0.5	6.3	15.3	9.0
Venugopalan *et al.* [18]	–	6.8	–	–
Rohrbach *et al.* [3]	**0.8**	**7.0**	16.0	10.0
BA-LSTM [12]	**0.8**	**7.0**	**16.7**	10.8
BA-GRU (ours)	**0.8**	6.8	16.5	**11.7**

Bold indicates the best performance.

TABLE II
EXPERIMENT RESULTS ON THE MSVD DATASET IN TERMS OF THE QUANTITATIVE EVALUATION METRICS BLEU IN ITS *4*-GRAM VARIANT (B_4), METEOR (M), ROUGE IN ITS LCS VARIANT (R_L) AND CIDER (C)

Model	B_4	M	R_L	C
SA-GoogleNet+3D-CNN [19]	41.9	29.6	–	–
LSTM-YT [8]	33.3	29.1	–	–
S2VT [7]	–	29.8	–	–
LSTM-E [10]	45.3	31.0	–	–
HRNE [11]	**46.7**	**33.9**	–	–
BA-LSTM [12]	41.5	31.3	68.6	*55.5*
BA-GRU (ours)	42.5	32.0	**68.8**	**59.0**

Bold indicates the best performance. Values in italic are obtained by re-running the code released by the authors of [12], which differ from those declared in their letter.

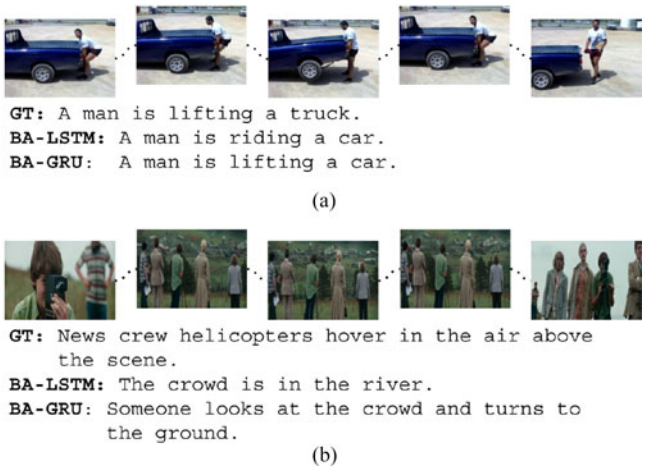

Fig. 3. Example results on a video from the MSVD test subset (a) and on a video from a movie in the MPII-MD test subset (b).

baselines in terms of the CIDEr metric, that has been reported in [24] best capturing human consensus on captions.

The lower performances of the MPII-MD dataset compared to MSVD are due to the fact that in the former the ground truth is taken from the DVS subtitle system, so it is not a real description of the scenes. Furthermore, the ground truth captions in the MSVD dataset are more precise and higher in number when compared to those of the MPII-MD dataset ($\sim$40 versus 1–2). Some examples are given in Fig. 3.

In addition, to gain some insights on the statistical significance of the presented quantitative results, we perform a *K*-fold cross-validation (with $K = 10$) of our approach and the BA-LSTM baseline on the MSVD dataset. We choose this dataset because it is smaller than the MPII-MD dataset, thus the model assessment experiment can be run in less time. The resulting values for the evaluation metrics, expressed in terms of mean and standard deviation, are reported in Table III. It is observed that our method is still comparable to the BA-LSTM baseline.

TABLE III
EXPERIMENT RESULTS OF THE *K*-FOLD CROSS-VALIDATION ON THE MSVD DATASET IN TERMS OF THE QUANTITATIVE EVALUATION METRICS BLEU IN ITS *4*-GRAM VARIANT (B_4), METEOR (M), ROUGE IN ITS LCS VARIANT (R_L) AND CIDER (C)

Model	B_4	M	R_L	C
BA-LSTM	41.5 ± 1.0	31.4 ± 0.3	68.5 ± 0.5	56.1 ± 2.0
BA-GRU	41.1 ± 1.1	31.2 ± 0.7	68.3 ± 0.5	53.5 ± 3.8

The results are expressed in terms of mean and standard deviation.

We also evaluate the training and testing time of the ten different variants of both BA-GRU and BA-LSTM. In particular, for BA-GRU the test time is on average 190.89 ± 5.28 ms, while for BA-LSTM is on average 197.78 ± 3.70 ms. In terms of training time, for BA-GRU it is on average $\sim 8\,\text{h}\,21' \pm \sim 5\,\text{h}\,34'$, while for BA-LSTM it is on average $\sim 13\,\text{h}\,40' \pm \sim 3\,\text{h}\,22'$. Despite both the BA-GRU and the BA-LSTM require much time to complete the training phase, saving 5 hours for each training helps in faster iteration when tuning hyperparameters for network deployment. For example, in the case of our 10-fold cross validation we saved on average 50 h with respect to the BA-LSTM model, and this could make the difference during the deployment of the architecture in a real robotic application.

The GRU block has fewer parameters than the LSTM block. In particular, our method BA-GRU requires approximately 114 MB of memory to store network weights, while the BA-LSTM needs 128 MB. Another benefit of using fewer parameters is that it reduces the risk of overfitting and, potentially, it allows the model to better generalize on completely new datasets.

E. Results on the ISARLab-VD Datasets

We further evaluate and compare BA-GRU with BA-LSTM on our collected dataset. Note that, in this case the algorithms are not trained on any subset of the ISARLab-VD dataset. With this experiment we want to test the generalization capabilities of the two architectures. We report the results of both the BA-GRU and BA-LSTM architectures trained on either the MPII-MD and MSVD datasets, both in quantitative and qualitative terms.

In particular, in Table IV we report the results in terms of the previously defined evaluation metrics. For the statistical significance of those results, we refer to Table V. There we also report the results of the ten variants of the BA-GRU and BA-LSTM models obtained via *K*-fold cross-validation on the MSVD dataset.

Some examples are given in Fig. 4 showing high resolution and low resolution videos. The reported ground truth description is the most representative of the multiple caption associated to the clips. We refer to the complete results corpus available

TABLE IV
EXPERIMENT RESULTS ON THE ISARLAB-VD DATASET IN TERMS OF THE QUANTITATIVE EVALUATION METRICS BLEU IN ITS *4*-GRAM VARIANT (B_4), METEOR (M), ROUGE IN ITS LCS VARIANT (R_L) AND CIDER (C)

Model	B_4	M	R_L	C
BA-LSTM on MSVD	14.0	19.5	51.6	23.3
BA-GRU on MSVD	**14.7**	**20.0**	**52.8**	**27.7**
BA-LSTM on MPII-MD	00.0	08.4	18.2	06.9
BA-GRU on MPII-MD	00.0	12.1	20.2	10.6

Bold indicates the best performance.

TABLE V
EXPERIMENT RESULTS OF THE TEN VARIANTS OF THE BA-GRU AND BA-LSTM MODELS OBTAINED VIA *K*-FOLD CROSS-VALIDATION ON THE MSVD DATASET IN TERMS OF THE QUANTITATIVE EVALUATION METRICS BLEU IN ITS *4*-GRAM VARIANT (B_4), METEOR (M), ROUGE IN ITS LCS VARIANT (R_L) AND CIDER (C)

Model	B_4	M	R_L	C
BA-LSTM	14.2 ± 0.8	19.0 ± 0.3	50.8 ± 0.7	25.2 ± 4.0
BA-GRU	15.0 ± 1.0	19.4 ± 0.5	51.2 ± 0.8	24.7 ± 2.6

The results are expressed in terms of mean and standard deviation.

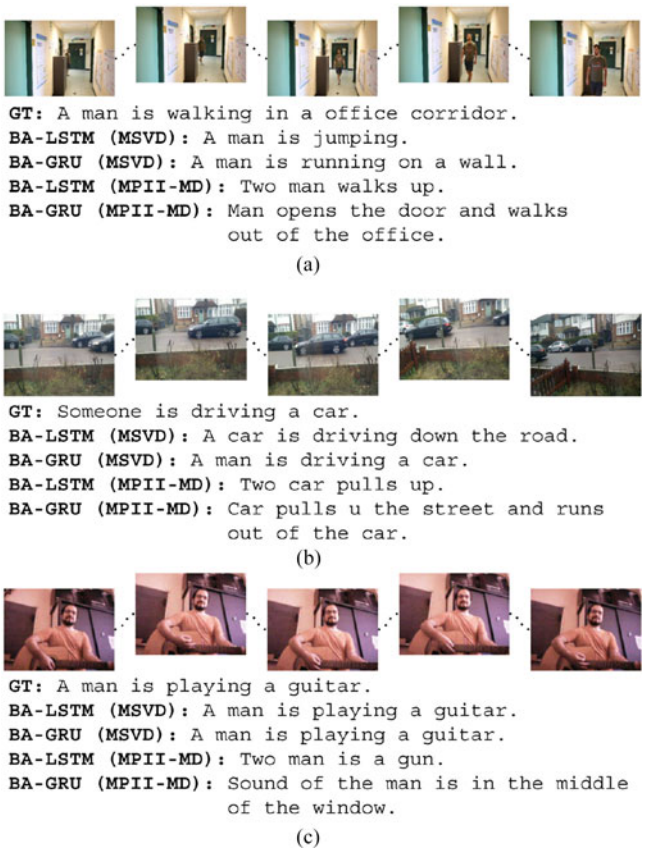

Fig. 4. Example results on videos from the ISARLab-VD dataset. In particular, (a) and (b) refer to videos that have been collected with two different high resolution cameras, while (c) refers to a low resolution video collected during the experiments with the Anki's COZMO robot.

online[4] for further examples. It can be observed that the quality of the videos does not influence the semantic and syntactic correctness of the description produced by the two methods. On the other hand, we observe that the captions for the videos of the ISARLab-VD dataset are simpler and less precise than those produced for the test subset videos of the public dataset used for the training. This suggests that these NLVD systems do not generalize well with respect to scenarios that significantly differ from those observed in training phase. Despite that, we can observe that the use of the BA-GRU gives a slight performance improvement. This suggests that the BA-GRU could be better suited to achieve architectures more robust to domain changes. The exploration of this aspect is beyond the scope of this letter, but this insights could be definitely useful for future investigations.

V. CONCLUSIONS AND FUTURE DEVELOPMENTS

This letter focuses on the NLVD task and presents a full-GRU encoder-decoder architecture to address it. We show that the proposed approach is faster to train and less memory consuming that other State-of-the-Art algorithms. Our method is also competitive in terms of performance on the public datasets which were partially used also for training. The experimental results on the devised dataset show that all methods have serious overfitting, making the generalization capabilities of new algorithm one of the most important questions to solve in future work.

Other future work is the ability to better cope with videos of variable lengths. This issue could be tackled by cutting the continuous video sequence in shorter chunks and describing each chunk using our proposed method as it is. However, being able to deal with much longer videos is surely of great interest and the development of effective solutions to this problem will be the subject of future work.

The code and the dataset used for this study are publicly available online.

REFERENCES

[1] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[2] K. Cho *et al.*, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proc. 2014 Conf. Empirical Methods Natural Lang. Process.*, 2014, pp. 1724–1734.
[3] A. Rohrbach, M. Rohrbach, and B. Schiele, "The long-short story of movie description," in *Proc. German Conf. Pattern Recognit.*, 2015, pp. 209–221.
[4] A. Barbu *et al.*, "Video in sentences out," in *Proc. 28th Conf. Uncertainty Artif. Intell.*, 2012, pp. 102–112.
[5] J. Donahue *et al.*, "Long-term recurrent convolutional networks for visual recognition and description," in *Proc. 2015 IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 2625–2634.
[6] Y. Zhu *et al.*, "Aligning books and movies: Towards story-like visual explanations by watching movies and reading books," in *Proc. 2015 IEEE Int. Conf. Comput. Vis.*, 2015, pp. 19–27.
[7] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, and K. Saenko, "Sequence to sequence-video to text," in *Proc. 2015 IEEE Int. Conf. Comput. Vis.*, 2015, pp. 4534–4542.

[4]http://isar.unipg.it/index.php?option=com_content&view=article&id=46&catid=2&Itemid=188

[8] S. Venugopalan, H. Xu, J. Donahue, M. Rohrbach, R. Mooney, and K. Saenko, "Translating videos to natural language using deep recurrent neural networks," in *Proc. 2015 Conf. North Amer. Chapter Assoc. Comput. Linguistics, Human Lang. Technol.*, 2014, pp. 1495–1504.

[9] M. Rohrbach, W. Qiu, I. Titov, S. Thater, M. Pinkal, and B. Schiele, "Translating video content to natural language descriptions," in *Proc. 2013 IEEE Int. Conf. Comput. Vis.*, 2013, pp. 433–440.

[10] Y. Pan, T. Mei, T. Yao, H. Li, and Y. Rui, "Jointly modeling embedding and translation to bridge video and language," in *Proc. 2016 IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 4594–4602.

[11] P. Pan, Z. Xu, Y. Yang, F. Wu, and Y. Zhuang, "Hierarchical recurrent neural encoder for video representation with application to captioning," in *Proc. 2016 IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1029–1038.

[12] L. Baraldi, C. Grana, and R. Cucchiara, "Hierarchical boundary-aware neural encoder for video captioning," in *Proc. 2017 IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 3185–3194.

[13] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," arXiv:1412.3555, 2014.

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. 2016 IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[15] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks," in *Proc. 2015 IEEE Int. Conf. Comput. Vis.*, 2015, pp. 4489–4497.

[16] Y. Bengio, N. Léonard, and A. Courville, "Estimating or propagating gradients through stochastic neurons for conditional computation," arXiv:1308.3432, 2013.

[17] A. Rohrbach, M. Rohrbach, N. Tandon, and B. Schiele, "A dataset for movie description," in *Proc. 2015 IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3202–3212.

[18] S. Venugopalan, L. A. Hendricks, R. Mooney, and K. Saenko, "Improving LSTM-based video description with linguistic knowledge mined from text," in *Proc. 2016 Conf. Empirical Methods Natural Lang. Process.*, 2016, pp. 1961–1966.

[19] L. Yao *et al.*, "Describing videos by exploiting temporal structure," in *Proc. 2015 IEEE Int. Conf. Comput. Vis.*, 2015, pp. 4507–4515.

[20] D. L. Chen and W. B. Dolan, "Collecting highly parallel data for paraphrase evaluation," in *Proc. 49th Annu. Meeting Assoc. Comput. Linguistics, Human Lang. Technolog.*, 2011, vol. 1, pp. 190–200.

[21] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: A method for automatic evaluation of machine translation," in *Proc. 40th Annu. Meeting Assoc. Comput. Linguistics*, 2002, pp. 311–318.

[22] S. Banerjee and A. Lavie, "METEOR: An automatic metric for MT evaluation with improved correlation with human judgments," in *Proc. ACL Workshop Intrinsic Extrinsic Eval. Measures Mach. Transl. Summarization*, 2005, vol. 29, pp. 65–72.

[23] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in *Proc. ACL-04 Workshop Text Summarization Branches Out*, Barcelona, Spain, 2004, vol. 8, pp. 74–81.

[24] R. Vedantam, C. Lawrence Zitnick, and D. Parikh, "CIDEr: Consensus-based image description evaluation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 4566–4575.

[25] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1–9.

[26] S. Ji, W. Xu, M. Yang, and K. Yu, "3D convolutional neural networks for human action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 221–231, Jan. 2013.

[27] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv:1409.1556, 2014.

[28] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. 25th Int. Conf. Adv. Neural Inf. Process. Syst.*, 2012, vol. 1, pp. 1097–1105.