Describing Videos by Exploiting Temporal Structure

Li Yao
Université de Montréal
li.yao@umontreal.ca

Atousa Torabi
Université de Montréal
atousa.torabi@umontreal.ca

Kyunghyun Cho
Université de Montréal
kyunghyun.cho@umontreal.ca

Nicolas Ballas
Université de Montréal
nicolas.ballas@umontreal.ca

Christopher Pal
École Polytechnique de Montréal
christopher.pal@polymtl.ca

Hugo Larochelle
Université de Sherbrooke
hugo.larochelle@usherbrooke.ca

Aaron Courville
Université de Montréal
aaron.courville@umontreal.ca

Abstract

Recent progress in using recurrent neural networks (RNNs) for image description has motivated the exploration of their application for video description. However, while images are static, working with videos requires modeling their dynamic temporal structure and then properly integrating that information into a natural language description. In this context, we propose an approach that successfully takes into account both the local and global temporal structure of videos to produce descriptions. First, our approach incorporates a spatial temporal 3-D convolutional neural network (3-D CNN) representation of the short temporal dynamics. The 3-D CNN representation is trained on video action recognition tasks, so as to produce a representation that is tuned to human motion and behavior. Second we propose a temporal attention mechanism that allows to go beyond local temporal modeling and learns *to automatically select the most relevant temporal segments given the text-generating RNN. Our approach exceeds the current state-of-art for both BLEU and METEOR metrics on the Youtube2Text dataset. We also present results on a new, larger and more challenging dataset of paired video and natural language descriptions.*

1. Introduction

The task of automatically describing videos containing rich and open-domain activities poses an important challenges for computer vision and machine learning research. It also has a variety of practical applications. For example, every minute, 100 hours of video are uploaded to YouTube.[1] However, if a video is poorly tagged, its utility is dramatically diminished [24]. Automatic video description generation has the potential to help improve indexing and search quality for online videos. In conjunction with speech synthesis technology, annotating video with natural language descriptions also has the potential to benefit the visually impaired.

Figure 1. High-level visualization of our approach to video description generation. We incorporate models of both the local temporal dynamic (i.e. within blocks of a few frames) of videos, as well as their global temporal structure. The local structure is modeled using the temporal feature maps of a 3-D CNN, while a temporal attention mechanism is used to combine information across the entire video. For each generated word, the model can focus on different temporal regions in the video. For simplicity, we highlight only the region having the maximum attention above.

While image description generation is already considered a very challenging task, the automatic generation of video description carries additional difficulties. Simply dealing with the sheer quantity of information contained in video data is one such challenge. Moreover, video description involves generating a sentence to characterize a video

[1] https://www.youtube.com/yt/press/statistics.html accessed on 2015-02-06.

clip lasting typically 5 to 10 seconds, or 120 to 240 frames. Often such clips contain complex interactions of actors and objects that evolve over time. All together it amounts to a vast quantity of information, and attempting to represent this information using a single, temporally collapsed feature representation is likely to be prone to clutter, with temporally distinct events and objects being potentially fused incoherently. It is therefore important that an automatic video description generator *exploit the temporal structure* underlying video.

We argue that there are two categories of temporal structure present in video: (1) local structure and (2) global structure. Local temporal structure refers to the fine-grained motion information that characterizes punctuated actions such as "answering the telephone" or "standing up". Actions such as these are relatively localized in time, evolving over only a few consecutive frames. On the other hand, when we refer to global temporal structure in video, we refer to the sequence in which objects, actions, scenes and people, etc. appear in a video. Video description may well be termed video summarization, because we typically look for a single sentence to summarize what can be a rather elaborate sequence of events. Just as good image descriptions often focus on the more salient parts of the image for description, we argue that good video description systems should selectively focus on the most salient features of a video sequence.

Recently, Venugopalan et al. [41] used a so-called encoder–decoder neural network framework [9] to automatically generate the description of a video clip. They extracted appearance features from each frame of an input video clip using a previously trained convolutional neural network [22]. The features from all the frames, or subsampled frames, were then collapsed via simple averaging to result in a single vector representation of the entire video clip. Due to this indiscriminate averaging of all the frames, this approach risks ignoring much of the temporal structure underlying the video clip. For instance, it is not possible to tell the order of the appearances of two objects from the collapsed features.

In this paper, we introduce a temporal attention mechanism to exploit *global* temporal structure. We also augment the appearance features with action features that encode *local* temporal structure. Our action features are derived from a spatio-temporal convolutional neural network (3-D CNN) [39, 19, 16]. The temporal attention mechanism is based on a recently proposed soft-alignment method [1] which was used successfully in the context of machine translation. While generating a description, the temporal attention mechanism selectively focuses on a small subset of frames, making it possible for the generator to describe only the objects and/or activities in that subset (see Fig. 1 for the graphical illustration). Our 3-D CNN, on the other hand, starts from both temporally and spatially local motion descriptors of video and hierarchically extracts more abstract action-related features. These features preserve and emphasize important local structure embedded in video for use by the description generator.

We evaluate the effectiveness of the proposed mechanisms for exploiting temporal structure on the most widely used open-domain video description dataset, called the Youtube2Text dataset [7], which consists of 1,970 video clips with multiple descriptions per video. We also test the proposed approaches on a much larger, and more recently proposed, dataset based on the descriptive video service (DVS) tracks in DVD movies [38], which contains 49,000 video clips.

Our work makes the following contributions: 1) We propose the use of a novel 3-D CNN-RNN encoder-decoder architecture which captures local spatio-temporal information. We find that despite the promising results generated by both prior work and our own here using static frame CNN-RNN video description methods, our experiments suggest that it is indeed important to exploit local temporal structure when generating a description of video. 2) We propose the use of an attention mechanism within a CNN-RNN encoder-decoder framework for video description and we demonstrate through our experiments that it allows features obtained through the global analysis of static frames throughout the video to be used more effectively for video description generation. Furthermore, 3) we observe that the improvements brought by exploiting global and local temporal information are complimentary, with the best performance achieved when both the temporal attention mechanism and the 3-D CNN are used together.

2. Video Description Generation Using an Encoder–Decoder Framework

In this section, we describe a general approach, based purely on neural networks to generate video descriptions. This approach is based on the encoder-decoder framework [9], which has been successfully used in machine translation [33, 9, 1] as well as image caption generation [20, 12, 42, 44, 18].

2.1. Encoder-Decoder Framework

The encoder-decoder framework consists of two neural networks; the encoder and the decoder. The encoder network ϕ encodes the input $\mathbf{x}$ into a continuous-space representation which may be a variable-sized set $V = \{\mathbf{v}_1, \ldots, \mathbf{v}_n\}$ of continuous vectors:

$$V = \{\mathbf{v}_1, \ldots, \mathbf{v}_n\} = \phi(\mathbf{x}).$$

The architecture choice for the encoder ϕ depends on the type of input. For example, in the case of machine translation, it is natural to use a recurrent neural network (RNN)

for the encoder, since the input is a variable-length sequence of symbols [33, 9]. With an image as input, a convolutional neural network (CNN) is another good alternative [44].

The decoder network generates the corresponding output y from the encoder representation V. As was the case with the encoder, the decoder's architecture must be chosen according to the type of the output. When the output is a natural language sentence, which is the case in automatic video description, an RNN is a method of choice.

The decoder RNN ψ runs sequentially over the output sequence. In brief, to generate an output y, at each step t the RNN updates its internal state $\mathbf{h}_t$ based on its previous internal state $\mathbf{h}_{t-1}$ as well as the previous output y_{t-1} and the encoder representation V, and then outputs a symbol y_t:

$$\begin{bmatrix} y_t \\ \mathbf{h}_t \end{bmatrix} = \psi(\mathbf{h}_{t-1}, y_{t-1}, V) \tag{1}$$

where for now we simply note as ψ the function updating the RNN's internal state and computing its output. The RNN is run recursively until the end-of-sequence symbol is generated, i.e., $y_t = \langle \text{eos} \rangle$.

In the remaining of this section, we detail choices for the encoder and decoder for a basic automatic video description system, taken from [41] and on which our work builds.

2.2. Encoder: Convolutional Neural Network

Deep convolutional neural networks (CNNs) have recently been successful at large-scale object recognition [22, 34]. Beyond the object recognition task itself, CNNs trained for object recognition have been found to be useful in a variety of other computer vision tasks such as object localization and detection (see, e.g., [29]). This has opened a door to a flood of computer vision systems that exploit representations from upper or intermediate layers of a CNN as generic high-level features for vision. For instance, the activation of the last fully-connected layer can be used as a fixed-size vector representation [20], or the feature map of the last convolutional layer can be used as a set of spatial feature vectors [44].

In the case where the input is a video clip, an image-trained CNN can be used for each frame separately, resulting in a single vector representation $\mathbf{v}_i$ of the i-th frame. This is the approach proposed by [41], which used the convolutional neural network from [22]. In our work here, we will also consider using the CNN from [34], which has demonstrated higher performance for object recognition.

2.3. Decoder: Long Short-Term Memory Network

As discussed earlier, it is natural to use a recurrent neural network (RNN) as a decoder when the output is a natural language sentence. This has been empirically confirmed in the contexts of machine translation [33, 9, 1], image caption generation [42, 44] and video description generation in open [41] and closed [12] domains. Among these recently successful applications of the RNN in natural language generation, it is noticeable that most of them [33, 9, 1, 42, 44], if not all, used long short-term memory (LSTM) units [14] or their variant, gated recurrent units (GRU) [9]. In this paper, we also use a variant of the LSTM units, introduced in [45], as the decoder.

The LSTM decoder maintains an internal memory state $\mathbf{c}_t$ in addition to the usual hidden state $\mathbf{h}_t$ of an RNN (see Eq. (1)). The hidden state $\mathbf{h}_t$ is the memory state $\mathbf{c}_t$ modulated by an output gate:

$$\mathbf{h}_t = \mathbf{o}_t \odot \mathbf{c}_t,$$

where $\odot$ is an element-wise multiplication. The output gate $\mathbf{o}_t$ is computed by

$$\mathbf{o}_t = \sigma(\mathbf{W}_o \mathbf{E}\,[y_{t-1}] + \mathbf{U}_o \mathbf{h}_{t-1} + \mathbf{A}_o \varphi_t(V) + \mathbf{b}_o),$$

where σ is the element-wise logistic sigmoid function and φ_t is a time-dependent transformation function on the encoder features. $\mathbf{W}_o$, $\mathbf{U}_o$, $\mathbf{A}_o$ and $\mathbf{b}_o$ are, in order, the weight matrices for the input, the previous hidden state, the context from the encoder and the bias. $\mathbf{E}$ is a word embedding matrix, and we denote by $\mathbf{E}\,[y_{t-1}]$ an embedding vector of word y_{t-1}.

The memory state $\mathbf{c}_t$ is computed as a weighted sum between the previous memory state $\mathbf{c}_{t-1}$ and the new memory content update $\tilde{\mathbf{c}}_t$:

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{c}}_t,$$

where the coefficients – called forget and input gates respectively – are given by

$$\begin{aligned} \mathbf{f}_t &= \sigma(\mathbf{W}_f \mathbf{E}\,[y_{t-1}] + \mathbf{U}_f \mathbf{h}_{t-1} + \mathbf{A}_f \varphi_t(V) + \mathbf{b}_f), \\ \mathbf{i}_t &= \sigma(\mathbf{W}_i \mathbf{E}\,[y_{t-1}] + \mathbf{U}_i \mathbf{h}_{t-1} + \mathbf{A}_i \varphi_t(V) + \mathbf{b}_i). \end{aligned}$$

The updated memory content $\tilde{\mathbf{c}}_t$ also depends on the current input y_{t-1}, previous hidden state $\mathbf{h}_{t-1}$ and the features from the encoder representation $\varphi_t(V)$:

$$\tilde{\mathbf{c}}_t = \tanh(\mathbf{W}_c \mathbf{E}\,[y_{t-1}] + \mathbf{U}_c \mathbf{h}_{t-1} + \mathbf{A}_c \varphi_t(V) + \mathbf{b}_c).$$

Once the new hidden state $\mathbf{h}_t$ is computed, a probability distribution over the set of possible words is obtained using a single hidden layer neural network

$$\mathbf{p}_t = \text{softmax}(\mathbf{U}_p \tanh(\mathbf{W}_p[\mathbf{h}_t, \varphi_t(V), \mathbf{E}\,[y_{t-1}]] + \mathbf{b}_p) + \mathbf{d}), \tag{2}$$

where $\mathbf{W}_p, \mathbf{U}_p, \mathbf{b}_p, \mathbf{d}$ are the parameters of this network, $[\dots]$ denotes vector concatenation. The softmax function allows us to interpret $\mathbf{p}_t$ as the probabilities of the distribution $p(y_t \mid y_{<t}, V)$ over words.

At a higher level, the LSTM decoder can be written down as

$$\begin{bmatrix} p(y_t \mid y_{<t}, V) \\ \mathbf{h}_t \\ \mathbf{c}_t \end{bmatrix} = \psi(\mathbf{h}_{t-1}, \mathbf{c}_{t-1}, y_{t-1}, V). \quad (3)$$

It is then trivial to generate a sentence from the LSTM decoder. For instance, one can recursively evaluate ψ and sample from the returned $p(y_t \mid \dots)$ until the sampled y_t is the end-of-sequence symbol. One can also approximately find the sentence with the highest probability by using a simple beam search [33].

In [41], Venugopalan et al. used this type of LSTM decoder for automatic video description generation. However, in their work the feature transformation function φ_t consisted in a simple averaging, i.e.,

$$\varphi_t(V) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{v}_i, \quad (4)$$

where the v_i's are the elements of the set V returned by the CNN encoder from Sec. 2.2. This averaging effectively collapses all the frames, indiscriminate of their temporal relationships, leading to the loss of temporal structure underlying the input video.

3. Exploiting Temporal Structure in Video Description Generation

In this section, we delve into the main contributions of this paper and propose an approach for exploiting both the local and global temporal structure in automatic video description.

3.1. Exploiting Local Structure: A Spatio-Temporal Convolutional Neural Net

We propose to model the local temporal structure of videos at the level of the temporal features $V = \{\mathbf{v}_1, \dots, \mathbf{v}_n\}$ that are extracted by the encoder. Specifically, we propose to use a spatio-temporal convolutional neural network (3-D CNN) which has recently been demonstrated to capture well the temporal dynamics in video clips [39, 19].

We use a 3-D CNN to build the higher-level representations that preserve and summarize the local motion descriptors of short frame sequences. This is done by first dividing the input video clip into a 3-D spatio-temporal grid of $16 \times 12 \times 2$ (width $\times$ height $\times$ timesteps) cuboids. Each cuboid is represented by concatenating the histograms of oriented gradients, oriented flow and motion boundary (HoG, HoF, MbH) [10, 43] with 33 bins. This transformation is done in order to make sure that local temporal structure (motion features) are well extracted and to reduce the computation of the subsequence 3-D CNN.

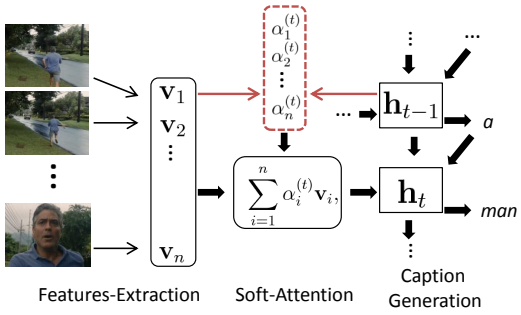

Figure 3. Illustration of the proposed temporal attention mechanism in the LSTM decoder

Our 3-D CNN architecture is composed of three 3-D convolutional layer, each followed by rectified linear activations (ReLU) and local max-pooling. From the activation of the last 3-D convolution+ReLU+pooling layer, which preserves the temporal arrangement of the input video and abstracts the local motion features, we can obtain a set of temporal feature vectors by max-pooling along the spatial dimensions (width and height) to get feature vectors that each summarize the content over short frame sequences within the video. Finally, these feature vectors are combined, by concatenation, with the image features extracted from single frames taken at similar positions across the video. Fig. 5 illustrates the complete architecture of the described 3-D CNN. Similarly to the object recognition trained CNN (see Sec. 2.2), the 3-D CNN is pre-train on activity recognition datasets.

3.2. Exploiting Global Structure: A Temporal Attention Mechanism

The 3-D CNN features of the previous section allows us to better represent short-duration actions in a subset of consecutive frames. However, representing a complete video by averaging these local temporal features as in Eq. 4 would jeopardize the model's ability to exploit the video's global temporal structure.

Our approach to exploiting such non-local temporal structure is to let the decoder selectively focus on only a small subset of frames at a time. By considering subsets of frames in sequence, the model can exploit the temporal ordering of objects and actions across the entire video clip and avoid conflating temporally disparate events. Our approach also has the potential of allowing the model to focus on key elements of the video that may have short duration. Methods that collapse the temporal structure risk overwhelming these short duration elements.

Specifically, we propose to adapt the recently proposed soft attention mechanism from [1], which allows the decoder to weight each temporal feature vector $V = \{\mathbf{v}_1, \dots, \mathbf{v}_n\}$. This approach has been used successfully by Xu et al. [44] for exploiting spatial structure underlying an

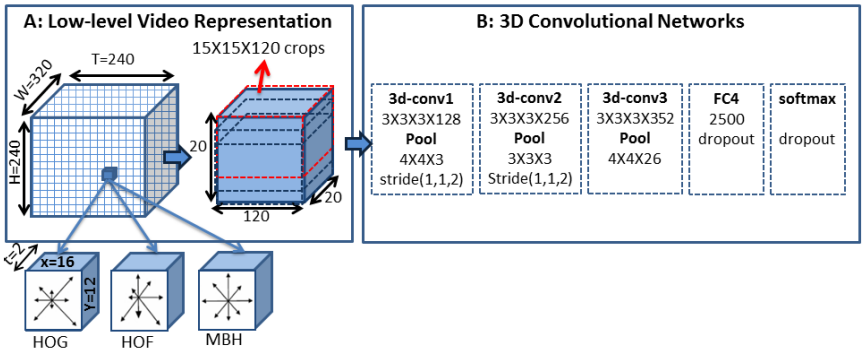

Figure 2. Illustration of the spatio-temporal convolutional neural network (3-D CNN). This network is trained for activity recognition. Then, only the convolutional layers are involved when generating video descriptions.

image. Here, we thus adapt it to exploit the temporal structure of video instead.

Instead of a simple averaging strategy (as shown in Eq. (4)), we take the *dynamic* weighted sum of the temporal feature vectors such that

$$\varphi_t(V) = \sum_{i=1}^{n} \alpha_i^{(t)} \mathbf{v}_i,$$

where $\sum_{i=1}^{n} \alpha_i^{(t)} = 1$ and $\alpha_i^{(t)}$'s are computed at each time step t inside the LSTM decoder (see Sec. 2.3). We refer to $\alpha_i^{(t)}$ as the attention weights at time t.

The attention weight $\alpha_i^{(t)}$ reflects the relevance of the i-th temporal feature in the input video given all the previously generated words, i.e., $y_1, \ldots y_{t-1}$. Hence, we design a function that takes as input the previous hidden state $\mathbf{h}_{t-1}$ of the LSTM decoder, which summarizes all the previously generated words, and the feature vector of the i-th temporal feature and returns the unnormalized relevance score $e_i^{(t)}$:

$$e_i^{(t)} = \mathbf{w}^\top \tanh\left(\mathbf{W}_a \mathbf{h}_{t-1} + \mathbf{U}_a \mathbf{v}_i + \mathbf{b}_a\right),$$

where $\mathbf{w}$, $\mathbf{W}_a$, $\mathbf{U}_a$ and $\mathbf{b}_a$ are the parameters that are estimated together with all the other parameters of the encoder and decoder networks.

Once the relevance scores $e_i^{(t)}$ for all the frames $i = 1, \ldots, n$ are computed, we normalize them to obtain the $\alpha_i^{(t)}$'s:

$$\alpha_i^{(t)} = \exp\left\{e_i^{(t)}\right\} / \sum_{j=1}^{n} \exp\left\{e_j^{(t)}\right\}.$$

We refer to the *attention mechanism* as this whole process of computing the unnormalized relevance scores and normalizing them to obtain the attention weights.

The attention mechanism allows the decoder to selectively focus on only a subset of frames by increasing the attention weights of the corresponding temporal feature. However, we do not explicitly force this type of selective attention to happen. Rather, this inclusion of the attention mechanism enables the decoder to exploit the temporal structure, *if* there is useful temporal structure in the data. Later in Sec. 5, we empirically show that this is indeed the case. See Fig. 3 for the graphical illustration of the temporal attention mechanism.

4. Related Work

Video description generation has been investigated and studied in other work, such as [21, 2, 28]. Most of these examples have, however, constrained the domain of videos as well as the activities and objects embedded in the video clips. Furthermore, they tend to rely on hand-crafted visual representations of the video, to which template-based or shallow statistical machine translation approaches were applied. In contrast, the approach we take and propose in this paper aims at open-domain video description generation with deep trainable models starting from low-level video representations, including raw pixel intensities (see Sec. 2.2) and local motion features (see Sec. 3.1).

In this sense, the approach we use here is more closely related to the recently introduced static image caption generation approaches based mainly on neural networks [20, 12, 42, 44, 18]. A neural approach to static image caption generation has recently been applied to video description generation by Venugopalan et al. [41]. However, their direct adaptation of the underlying static image caption generation mechanism to the videos is limited by the fact that the model tends to ignore the temporal structure of the underlying video. Such structure has demonstrated to be helpful in the context of event and action classification [35, 13, 6], and is explored in this paper. Other recent work [27] has explored the use of DVS annotated video for video description research and has underscored the observation that DVS descriptions are typically much more relevant and accurate descriptions of the visual content of a video compared to movie scripts. They present results using both DVS and script based annotations as well as cooking activities.

While other work has explored 3-D Deep Networks for video [36, 16, 18, 30] our particular approach differs in a number of ways from prior work in that it is based on CNNs as opposed to other 3-D deep architectures and we focus on

pre-training the model on a number of widely used action recognition datasets. In contrast to other 3-D CNN formulations, the input to our 3-D CNN consists of features derived from a number of state of the art image descriptors. Our model is also fully 3-D in that we model entire volumes across a video clip. In this paper, we use a state-of-the-art static convolutional neural network (CNN) and a novel spatio-temporal 3-D CNN to model input video clips. This way of modeling video using feedforward convolutional neural networks, has become increasingly popular recently [41, 30, 39]. However, there has also been a stream of research on using recurrent neural networks (RNN) for modeling video clips. For instance, in [32], Srivastava et al. propose to use long short-term memory units to extract video features. Ranzato et al. in [26] also models a video clip with an RNN, however, after vector-quantizing image patches of the video clip. In contrast to other approaches such as [12], which have explored CNN-RNN coupled models for video description, here we use an attention mechanism, use a 3-D CNN and focus on open-domain video description.

5. Experiments

We test the proposed approaches on two video-description corpora: Youtube2Text [7] and DVS [38]. Implementations are available at `https://github.com/yaoli/arctic-capgen-vid`.

5.1. Datasets

Youtube2Text The Youtube2Text video corpus [7] is well suited for training and evaluating an automatic video description generation model. The dataset has 1,970 video clips with multiple natural language descriptions for each video clip. In total, the dataset consists of approximately 80,000 video / description pairs, with the vocabulary of approximately 16,000 unique words. The dataset is open-domain and covers a wide range of topics including sports, animals and music. Following [41], we split the dataset into a training set of 1,200 video clips, a validation set of 100 clips and a test set consisting of the remaining clips.

DVS The DVS dataset was recently introduced in [38] with a much larger number of video clips and accompanying descriptions than the existing video/description corpora such as Youtube2Text. It contains video clips extracted from 92 DVD movies along with semi-automatically transcribed descriptive video service (DVS) narrations. The dataset consists of 49,000 video clips covering a wide variety of situations. We follow the standard split of the dataset into a training set of 39,000 clips, a validation set of 5,000 clips and a test set of 5,000 clips, as suggested by [38].

Description Preprocessing We preprocess the descriptions in both the Youtube2Text and DVS datasets with `wordpunct_tokenizer` from the NLTK toolbox.[2]. We did not do any other preprocessing such as lowercasing and rare word elimination. After preprocessing, the numbers of unique words were 15,903 for Youtube2Text and 17,609 for DVS Dataset.

Video Preprocessing To reduce the computational and memory requirement, we only consider the first 240 frames of each video [3] For appearance features, (trained) 2-D *GoogLeNet* [34] CNN is used to extract fixed-length representation (with the help of the popular implementation in Caffe [17]). Features are extracted from the `pool5/7x7_s1` layer. We select 26 equally-spaced frames out of the first 240 from each video and feed them into the CNN to obtain a 1024 dimensional frame-wise feature vector. We also apply the spatio-temporal 3-D CNN (trained as described in Sec. 5.2) in order to extract local motion information[4]. When using 3-D CNN without temporal attention, we simply use the 2500-dimensional activation of the last fully-connection layer. When we combine the 3-D CNN with the temporal attention mechanism, we leverage the last convolutional layer representation leading to 26 feature vectors of size 352. Those vector are contatenated with the 2D CNN features resulting in 26 feature vectors with 1376 elements.

5.2. Experimental Setup

Models We test four different model variations for video description generation based on the underlying encoder-decoder framework, with results presented in Table 1. *Enc-Dec (Basic)* denotes a baseline incorporating neither local nor global temporal structure. Is it based on an encoder using the 2-D GoogLeNet CNN [34] as discussed in Section 2.2 and the LSTM-based decoder outlined in Section 2.3. *Enc-Dec + Local* incorporates local temporal structure via the integration of our proposed 3-D CNN features (as outlined in Section 3.1) with the 2-D GoogLeNet CNN features as described above. *Enc-Dec + Global* adds the temporal attention mechanism of Section 3.2. Finally, *Enc-Dec + Local + Global* incorporates both the 3-D CNN and the temporal attention mechanism into the model. All models otherwise use the same number of temporal features $\mathbf{v}_i$. These experiments will allow us to investigate whether the contributions from the proposed approaches are complimentary and can be combined to further improve performance.

[2] `http:/s/www.nltk.org/index.html`

[3] When the video clip has less than 240 frames, we pad the video with all-zero frames to make it into 240-frame long.

[4] We perturb each video along three axes to form random crops by taking multiple $15 \times 15 \times 120$ cuboids out of the original $20 \times 20 \times 120$ cuboids, and the final representation is the average of the representations from these perturbed video clips.

Table 1. Performance of different variants of the model on the Youtube2Text and DVS datasets.

	Youtube2Text				DVS			
Model	BLEU	METEOR	CIDEr	Perplexity	BLEU	METEOR	CIDEr	Perplexity
Enc-Dec (Basic)	0.3869	0.2868	0.4478	33.09	0.003	0.044	0.044	88.28
+ Local (3-D CNN)	0.3875	0.2832	0.5087	33.42	0.004	0.051	0.050	84.41
+ Global (Temporal Attention)	0.4028	0.2900	0.4801	27.89	0.003	0.040	0.047	66.63
+ Local + Global	**0.4192**	**0.2960**	**0.5167**	**27.55**	**0.007**	**0.057**	**0.061**	**65.44**
Venugopalan *et al.* [41]	0.3119	0.2687	-	-	-	-	-	-
+ Extra Data (Flickr30k, COCO)	0.3329	0.2907	-	-	-	-	-	-
Thomason *et al.* [37]	0.1368	0.2390	-	-	-	-	-	-

Training For all video description generation models, we estimated the parameters by maximizing the log-likelihood:

$$\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{N}\sum_{n=1}^{N}\sum_{i=1}^{t_n}\log p(y_i^n \mid y_{<i}^n, \mathbf{x}^n, \boldsymbol{\theta}),$$

where there are N training video-description pairs $(\mathbf{x}^n, y^n)$, and each description y^n is t_n words long.

We used Adadelta [46] with the gradient computed by the backpropagation algorithm. We optimized the hyperparameters (e.g. number of LSTM units and the word embedding dimensionality) using random search to maximize the log-probability of the validation set. [5] Training continued until the validation log-probability stopped increasing for 5,000 updates. As mentioned earlier in Sec. 3.1, the 3-D CNN was trained on activity recognition datasets. Due to space limitation, details regarding the training and evaluation of the 3-D CNN on activity recognition datasets are provided in the Supplementary Material.

Evaluation We report the performance of our proposed method using test set perplexity and three model-free automatic evaluation metrics. These are BLEU [25], METEOR [11] and CIDEr [40]. We use the evaluation script prepared and introduced in [8].

5.3. Quantitative Analysis

In the first block of Table 1, we present the performance of the four different variants of the model using all four metrics: BLEU, METEOR, CIDEr and perplexity. Subsequent lines in the table give comparisons with prior work. The first three rows (Enc-Dec (Basic), +Local and +Global), show that it is generally beneficial to exploit some type of temporal structure underlying the video. Although this benefit is most evident with perplexity (especially with the temporal attention mechanism exploiting global temporal structure), we observe a similar trend with the other model-free metrics and across both Youtube2Text and DVS datasets.

We observe, however, that the biggest gain can be achieved by letting the model exploit *both* local and global temporal structure (the fourth row in Table 1). We observed this gain consistently across both datasets as well as using all four automatic evaluation metrics.

[5] Refer to the Supplementary Material for the selected hyperparameters.

5.4. Qualitative Analysis

Although the model-free evaluation metrics such as the ones we used in this paper (BLEU, METEOR, CIDEr) were designed to reflect the agreement level between reference and generated descriptions, it is not intuitively clear how well those numbers (see Table 1) reflect the quality of the actual generated descriptions. Therefore, we present some of the video clips and their corresponding descriptions, both generated and reference, from the test set of each dataset. Unless otherwise labeled, the visualizations in this section are from the best model which exploits both global and local temporal structure (the fourth row of Table 1).

In Fig. 4, two video clips from the test set of Youtube2Text are shown. We can clearly see that the generated descriptions correspond well with the video clips. In Fig. 4, we show also two sample video clips from the DVS dataset. Clearly, the model does not perform as well on the DVS dataset as it did on Youtube2Text, which was already evident from the quantitative analysis in Sec. 5.3. However, we still observe that the model often focuses correctly on a subset of frames according to the word to be generated. For instance, in the left pane, when the model is about to generate the second "SOMEONE", it focuses mostly on the first frame. Also, on the right panel, the model correctly attends to the second frame when the word "types" is about to be generated. As for the 3-D CNN local temporal features, we see that they allowed to correctly identify the action as "frying", as opposed to simply "cooking".

More samples of the video clips and the generated/reference descriptions can be found in the Supplementary Material, including visualizations from the global temporal attention model alone (see the third row in Table 1).

6. Conclusion

In this work, we address the challenging problem of producing natural language descriptions of videos. We identify and underscore the importance of capturing both lo-

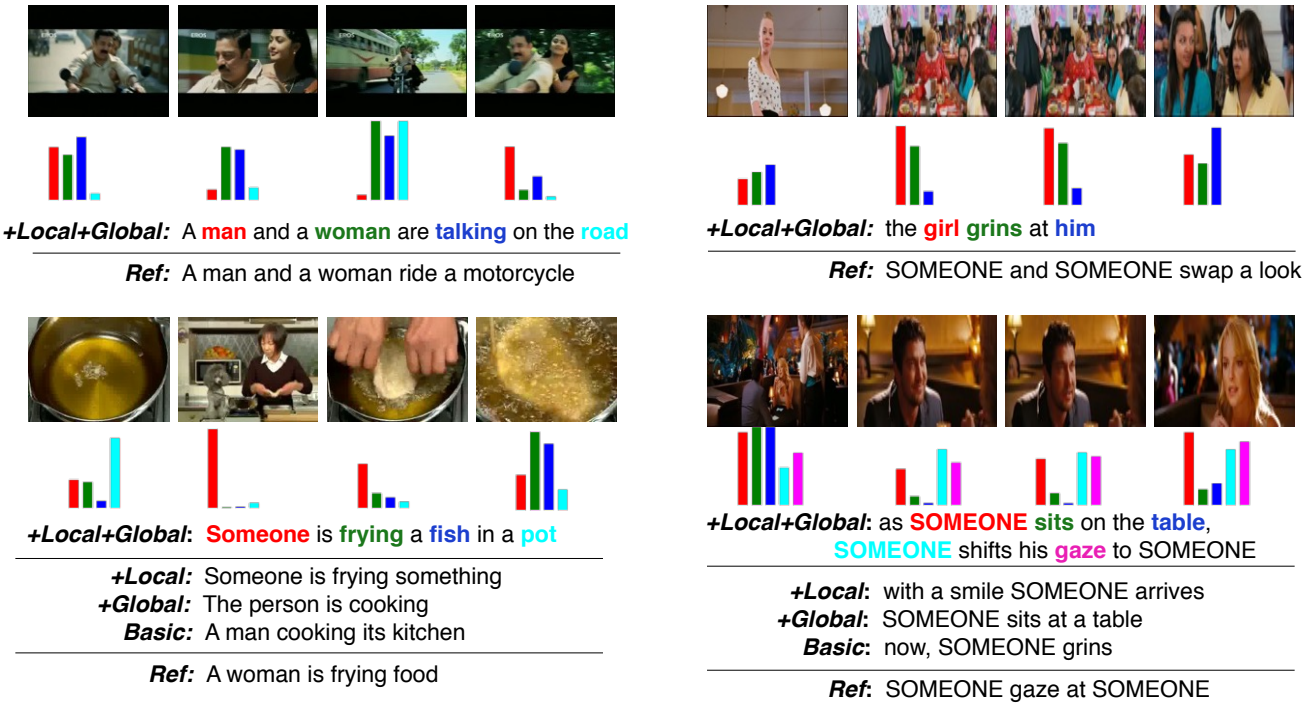

Figure 4. Four sample videos and their corresponding generated and ground-truth descriptions from Youtube2Text (Left Column) and DVS (Right Column). The bar plot under each frame corresponds to the attention weight α_i^t for the frame when the corresponding word (color-coded) was generated. From the top left panel, we can see that when the word "road" is about to be generated, the model focuses highly on the third frame where the road is clearly visible. Similarly, on the bottom left panel, we can see that the model attends to the second frame when it was about to generate the word "Someone". The bottom row includes alternate descriptions generated by the other model variations.

cal and global temporal structure in addition to frame-wise appearance information. To this end, we propose a novel 3-D convolutional neural network that is designed to capture local fine-grained motion information from consecutive frames. In order to capture global temporal structure, we propose the use of a temporal attentional mechanism that learns the ability to focus on subsets of frames. Finally, the two proposed approaches fit naturally together into an encoder-decoder neural video caption generator.

We have empirically validated each approach on both Youtube2Text and DVS datasets on four standard evaluation metrics. Experiments indicate that models using either approach improve over the baseline model. Furthermore, combining the two approaches gives the best performance. In fact, we achieved the state-of-the-art results on Youtube2Text with the combination.

Given the challenging nature of the task, we hypothesize that the performance on the DVS dataset could be significantly improved by incoporating another recently proposed dataset [27] similar to the DVS data used here. In addition, we have some preliminary experimental results that indicate that further performance gains are possible by leveraging image caption generation datasets such as MS COCO [8] and Flickr [15]. We intend to more fully explore this direction in future work.

Acknowledgments

The authors would like to thank the developers of Theano [5, 3]. We acknowledge the support of the following organizations for research funding and computing support: NSERC, FQRNT, Samsung, Calcul Quebec, Compute Canada, the Canada Research Chairs and CIFAR.

References

[1] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *ICLR*, 2015.

[2] A. Barbu, A. Bridge, Z. Burchill, D. Coroian, S. Dickinson, S. Fidler, A. Michaux, S. Mussman, S. Narayanaswamy, D. Salvi, et al. Video in sentences out. *UAI*, 2012.

[3] F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. J. Goodfellow, A. Bergeron, N. Bouchard, and Y. Bengio. Theano: new features and speed improvements. Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop, 2012.

[4] J. Bergstra and Y. Bengio. Random search for hyper-parameter optimization. *JMLR*, 2012.

[5] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio. Theano: a CPU and GPU math expression compiler. In *Proceedings of the Python for Scientific Computing Conference (SciPy)*, 2010.

[6] P. Bojanowski, R. Lajugie, F. Bach, I. Laptev, J. Ponce, C. Schmid, and J. Sivic. Weakly supervised action labeling in videos under ordering constraints. In *ECCV*. 2014.

[7] D. L. Chen and W. B. Dolan. Collecting highly parallel data for paraphrase evaluation. In *ACL*, 2011.

[8] X. Chen, H. Fang, T.-Y. Lin, R. Vedantam, S. Gupta, P. Dollar, and C. L. Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv 1504.00325*, 2015.

[9] K. Cho, B. van Merrienboer, C. Gulcehre, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *EMNLP*, Oct. 2014.

[10] N. Dalal, B. Triggs, and C. Schmid. Human detection using oriented histograms of flow and appearance. In *ECCV*. 2006.

[11] M. Denkowski and A. Lavie. Meteor universal: Language specific translation evaluation for any target language. In *EACL Workshop*, 2014.

[12] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. *CVPR*, 2015.

[13] A. Gaidon, Z. Harchaoui, and C. Schmid. Temporal localization of actions with actoms. *PAMI*, 2013.

[14] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 1997.

[15] M. Hodosh, P. Young, and J. Hockenmaier. Framing image description as a ranking task: Data, models and evaluation metrics. *Journal of Artificial Intelligence Research*, 2013.

[16] S. Ji, W. Xu, M. Yang, and K. Yu. 3d convolutional neural networks for human action recognition. *PAMI*, 2013.

[17] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv:1408.5093*, 2014.

[18] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *CVPR*, 2014.

[19] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*. IEEE, 2014.

[20] R. Kiros, R. Salakhutdinov, and R. S. Zemel. Unifying visual-semantic embeddings with multimodal neural language models. *ACL*, 2014.

[21] A. Kojima, T. Tamura, and K. Fukunaga. Natural language description of human activities from video images based on concept hierarchy of actions. *IJCV*, 2002.

[22] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[23] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: a large video database for human motion recognition. In *ICCV*, 2011.

[24] N. Morsillo, G. Mann, and C. Pal. Youtube scale, large vocabulary video annotation. In *Video Search and Mining*. 2010.

[25] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, 2002.

[26] M. Ranzato, A. Szlam, J. Bruna, M. Mathieu, R. Collobert, and S. Chopra. Video (language) modeling: a baseline for generative models of natural videos. *arXiv: 1412.6604*, 2014.

[27] A. Rohrbach, M. Rohrbach, N. Tandon, and B. Schiele. A dataset for movie description. *CVPR*, 2015.

[28] M. Rohrbach, W. Qiu, I. Titov, S. Thater, M. Pinkal, and B. Schiele. Translating video content to natural language descriptions. In *ICCV*, 2013.

[29] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *ICLR*, 2014.

[30] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. *NIPS*, 2014.

[31] K. Soomro, A. R. Zamir, and M. Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *CoRR*, abs/1212.0402, 2012.

[32] N. Srivastava, E. Mansimov, and R. Salakhutdinov. Unsupervised learning of video representations using lstms. *arXiv: 1502.04681*, 2015.

[33] I. Sutskever, O. Vinyals, and Q. V. V. Le. Sequence to sequence learning with neural networks. In *NIPS*. 2014.

[34] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *CVPR*, 2015.

[35] K. Tang, L. Fei-Fei, and D. Koller. Learning latent temporal structure for complex event detection. In *CVPR*. IEEE, 2012.

[36] G. W. Taylor, R. Fergus, Y. LeCun, and C. Bregler. Convolutional learning of spatio-temporal features. In *Computer Vision–ECCV 2010*, pages 140–153. Springer, 2010.

[37] J. Thomason, S. Venugopalan, S. Guadarrama, K. Saenko, and R. Mooney. Integrating language and vision to generate natural language descriptions of videos in the wild. In *COLING*, 2014.

[38] A. Torabi, C. Pal, H. Larochelle, and A. Courville. Using descriptive video services to create a large data source for video annotation research. *arXiv: 1503.01070*, 2015.

[39] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. C3D: Generic features for video analysis. *arXiv:1412.0767*, 2014.

[40] R. Vedantam, C. L. Zitnick, and D. Parikh. CIDEr: Consensus-based image description evaluation. *CVPR*, 2015.

[41] S. Venugopalan, H. Xu, J. Donahue, M. Rohrbach, R. Mooney, and K. Saenko. Translating videos to natural language using deep recurrent neural networks. *NAACL*, 2015.

[42] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. *CVPR*, 2015.

[43] H. Wang, M. M. Ullah, A. Klser, I. Laptev, and C. Schmid. Evaluation of local spatio-temporal features for action recognition. In *University of Central Florida, U.S.A*, 2009.

[44] K. Xu, J. Ba, R. Kiros, , K. Cho, A. Courville, R. Salakhutdinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. *ICML*, 2015.

[45] W. Zaremba, I. Sutskever, and O. Vinyals. Recurrent neural network regularization. *arXiv: 1409.2329*, 2014.

[46] M. D. Zeiler. ADADELTA: an adaptive learning rate method. Technical report, 2012.

7. Details of experiments

7.1. 3-D CNN

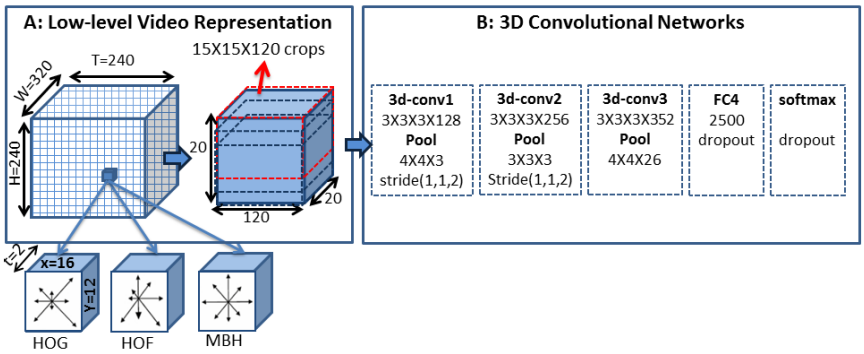

Figure 5. Illustration of the spatio-temporal convolutional neural network (3-D CNN). This network is trained for activity recognition. Then, only the convolutional layers are involved when generating video descriptions.

The 3-D CNN architecture is specified in Figure 5. Our model is composed of three 3-D convolutional layers, using $3\times3\times3$ kernels. The number of output features after the different convolutions is given in Figure 5. Each convolutional layer is followed by a rectified linear activations (ReLU) and local max-pooling. After the convolutions, a fully-connected layer (dimension 2500, with ReLU activation) is applied, followed by a Softmax layer. A dropout of 0.5 is applied on those last two layers.

Multitask learning is used to train the model on three human activity recognition datasets: UCF101 [31] with 13320 Youtube videos and 101 various human activity classes, HMDB51 [23] with 3700 videos and 51 various human activity classes, and a random subset of Sports-1M dataset [18] using 50,000 videos that have 487 sports labels.

We trained the 3-D CNN using stochastic gradient descent with a momentum of 0.7. Learning rate is initially set to 0.1 and then is decreased, using the following scheme 0.05, 0.02, 0.01, each time the validation cost stagnate. At each iteration a minibatch of size 48 is constructed by sampling uniformly all 3 human activity datasets. We perturb each video along three axes to encourage the model to learn invariant feature representation, we take random crops of size $15\times15\times120$ cuboids out of the original $20\times20\times120$ cuboids. Video are also randomly flipped.

Despite our interest in video-description, we validate that our model obtains reasonable performances on the activity recognition task using HMDB51 (split 1) and UCF101. On HMDB (split 1) our model achieves an accuracy of 52.3%, our result is 3% lower than the best motion-based single model (temporal-based CNN of [30]). On UCF-101, our 3-D CNN obtain an accuracy of 76.49%. While the temporal-based CNN [30] achieves 83.7%, our model outperforms other single 3-D convolution based approaches such as C3D (72.29%) [39] and slow-fusion convnet (65.4%) [19].

7.2. Encoder-Decoder Model Training

Hyperparameters reflects the learning capacity of the models. We have made sure each type of models have been sufficient explored in their hyperparameters. The model selections on both Youtube2Text and DVS are performed by random search [4]. There are four types of models being trained:

- Basic Enc-Dec
- Basic Enc-Dec + Local (3-D ConvNet)
- Basic + Global (Temporal Attention)
- Basic + Local + Global

For each of four types of models, we performed 50 experiments with random search on the critical hyperparameters. Each of the 50 experiments is associated with a specific hyperparameter setup. And the 50 setups are shared cross all four types of models. The critial hyperparameters experimented are:

- the dimensionality of word embedding, in the range of [100, 1000]

- the dimensionality of LSTM hidden/memory states, in the range of [100,3000]
- dropout, either use or not used, decided at random.

The same procedure is used on both Youtube2Text and DVS datasets.

Table 2. Hyperparameters of best models on Youtube2Text.

model	emb	lstm	dropout
Basic Enc-Dec	211	1096	True
Basic Enc-Dec + Local (3-D ConvNet)	161	1292	True
Basic + Global (Temporal Attention)	476	2231	True
Basic + Local + Global	454	1714	True

Table 3. Hyperparameters of best models on DVS.

model	emb	lstm	dropout
Basic Enc-Dec	345	1014	True
Basic Enc-Dec + Local (3-D ConvNet)	512	2560	True
Basic + Global (Temporal Attention)	656	1635	True
Basic + Local + Global	454	1714	True

8. Inspecting the learned soft-attention coefficients α

We illustrate the caption generation process of the proposed soft-attentional models trained with Basic+Global v.s. trained with Basic+Local+Global, with a dynamic α on frames for each word in the generated caption.

The bar chart shows the magnitude of α. The generated caption is shown on the left. Each generated word corresponds to an α vector, show in the same row. Each bar corresponds to a particular frame on the very top of the figure, organized sequentially. Within the same row, the height of the bar shows the importance of its corresponding frame in generating that word. 20 frames are shown for better visibility.

8.1. Caption generation and α visualization on Youtube2Text testset

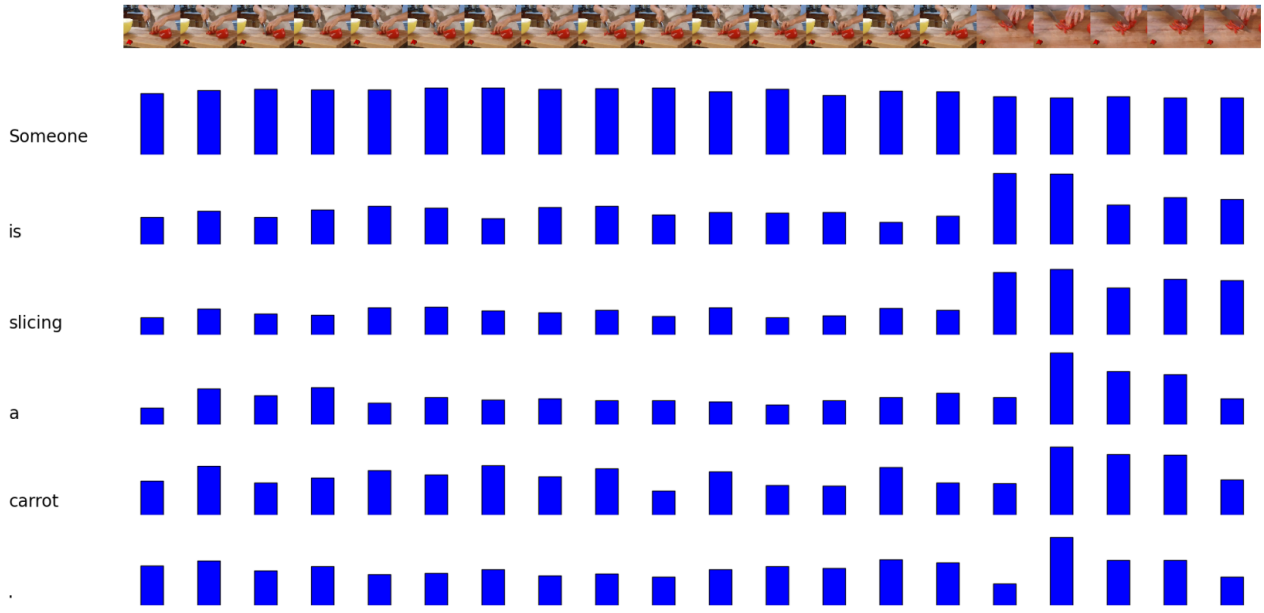

Figure 6. Model type: Basic + Global. Model shifts its attention across frames to generate a caption. The bar char shows the magitude of α, sum to 1 row-wise, the higher the bar, the bigger the magnitude.

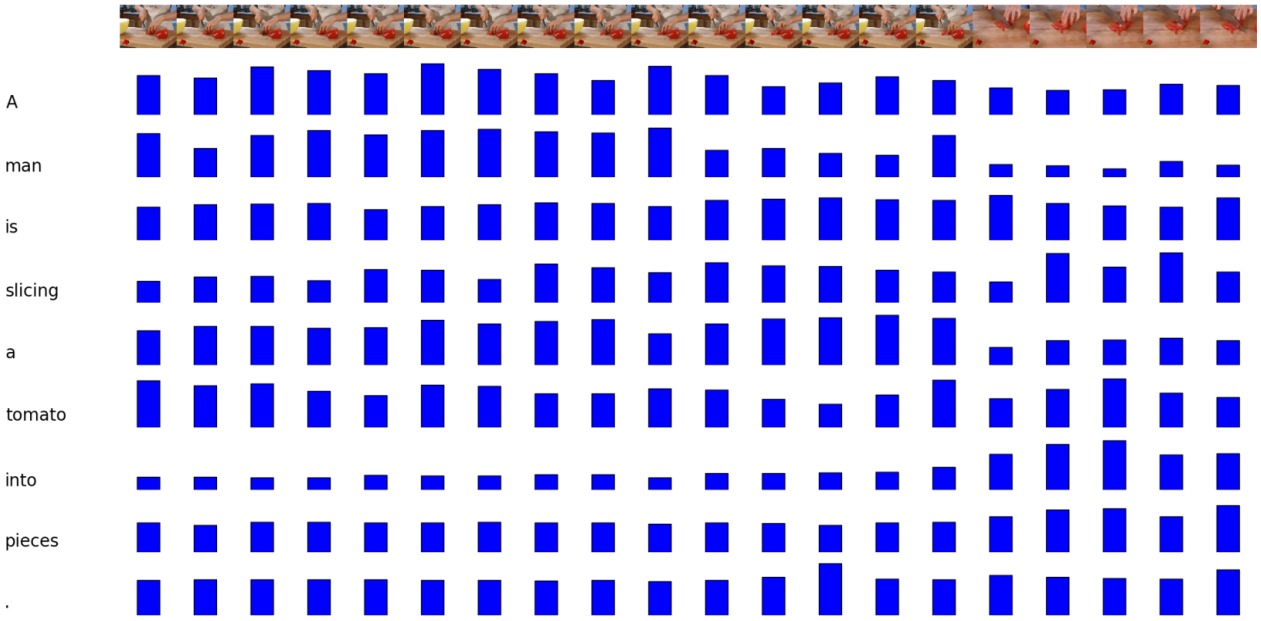

Figure 7. Model type: Basic + Local + Global. Model shifts its attention across frames when generating the caption. The bar char shows the magitude of α, sum to 1 row-wise, the higher the bar, the bigger the magnitude. It is doing a better job at guessing the object being chopped , compared with Figure 6.

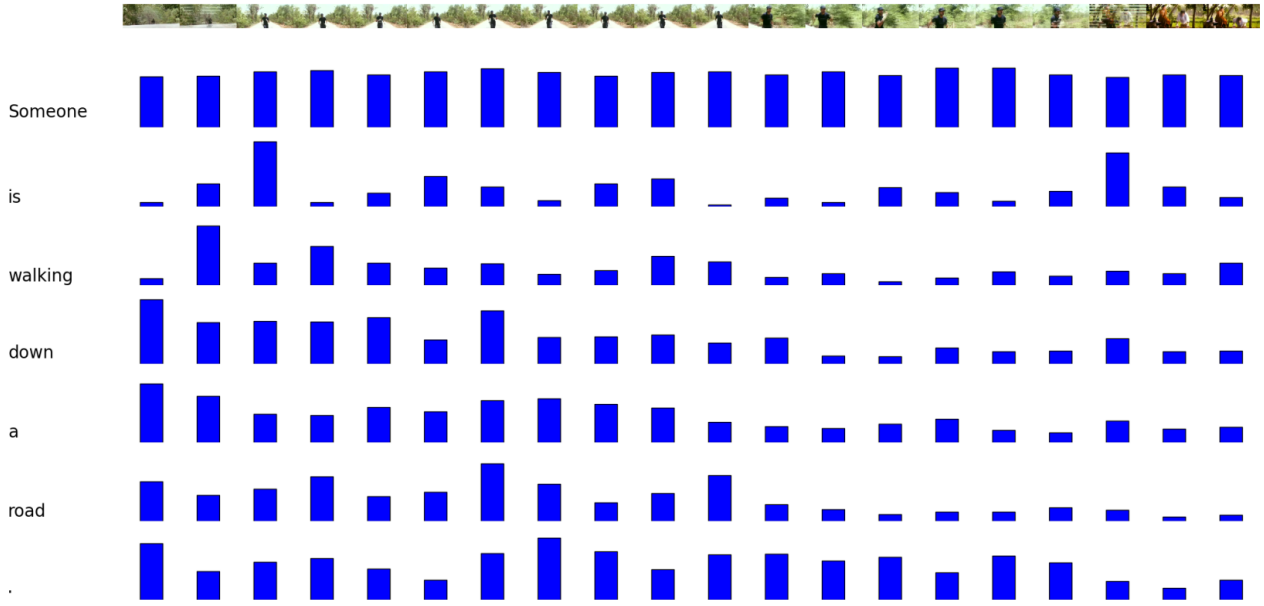

Figure 8. Model type: Basic + Global. Model shifts its attention across frames to generate a caption. The bar char shows the magitude of α, sum to 1 row-wise, the higher the bar, the bigger the magnitude.

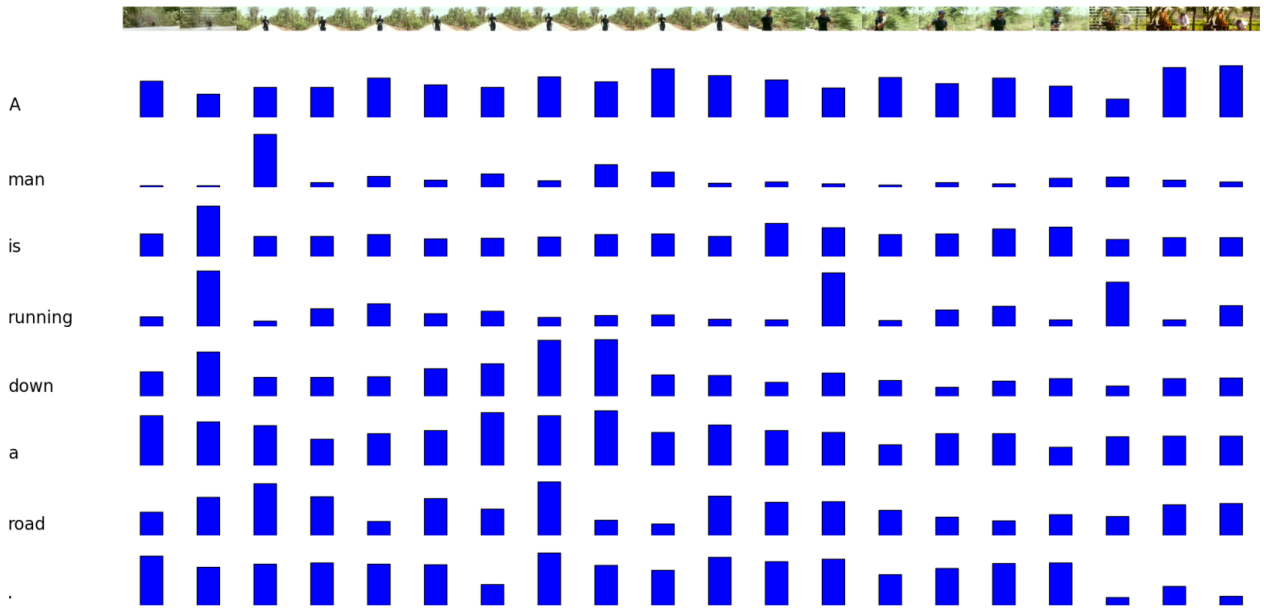

Figure 9. Model type: Basic + Local + Global. Model shifts its attention across frames when generating the caption. The bar char shows the magitude of α, sum to 1 row-wise, the higher the bar, the bigger the magnitude. The use of additional motion features offers more faithful description of the action than the one without ("running" v.s. "walking" in Figure 8).

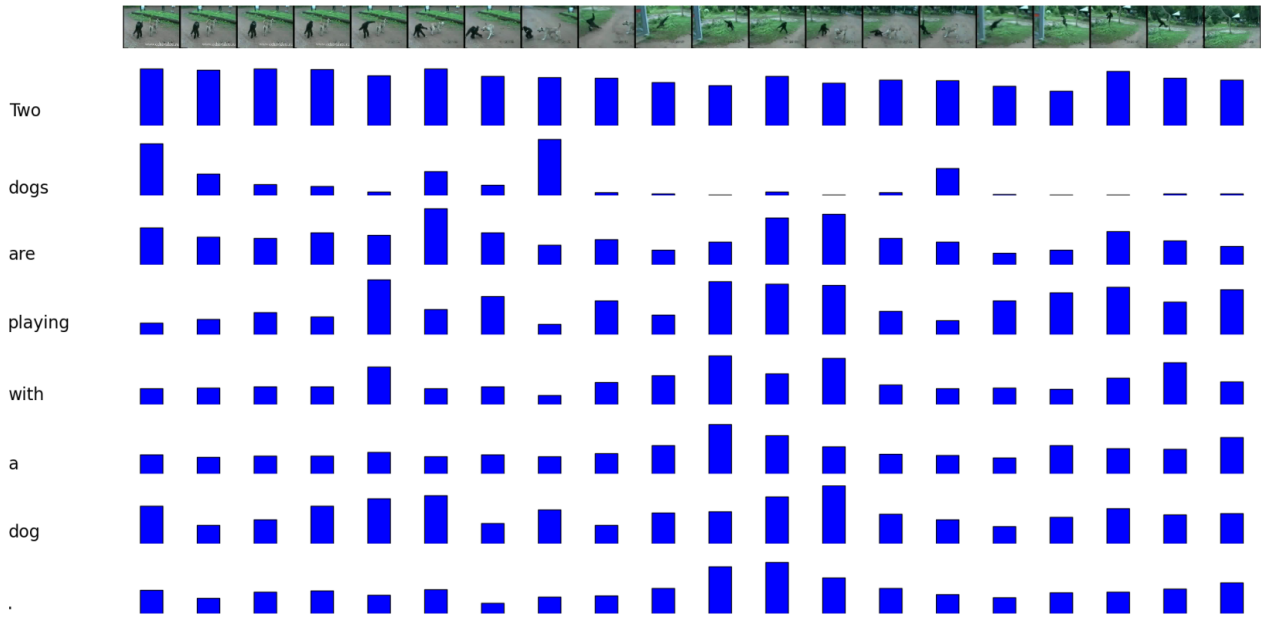

Figure 10. Model type: Basic + Global. Model shifts its attention across frames to generate a caption. The bar char shows the magitude of α, sum to 1 row-wise, the higher the bar, the bigger the magnitude.

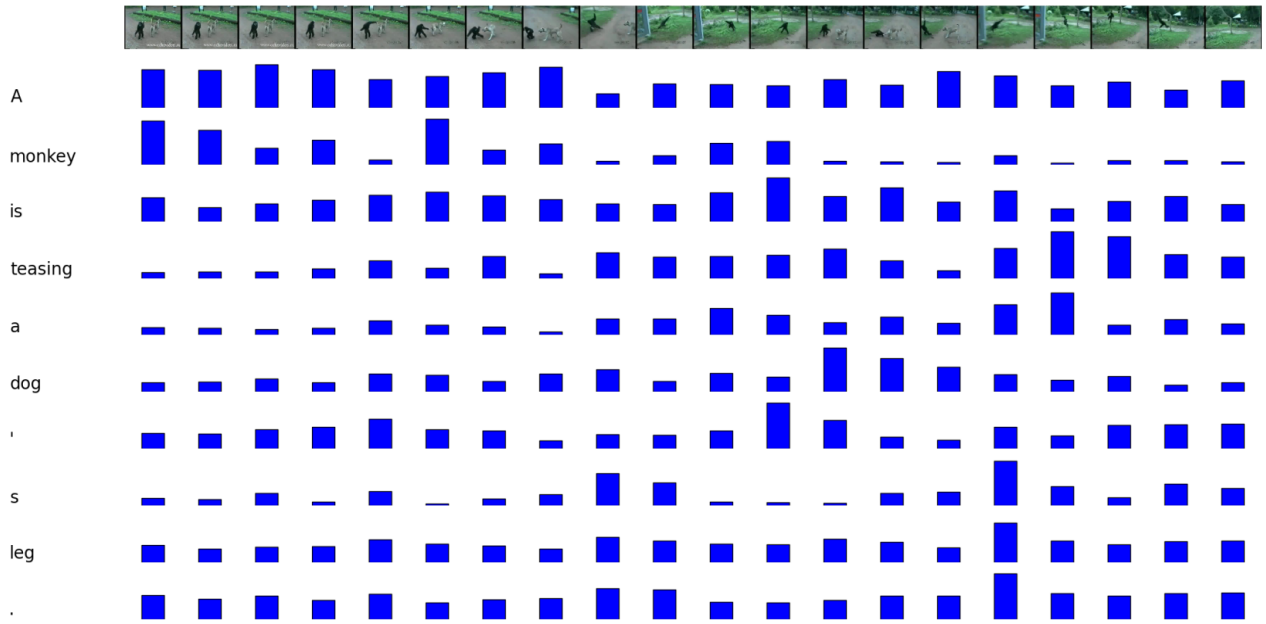

Figure 11. Model type: Basic + Local + Global. Model shifts its attention across frames to generate a caption. The bar char shows the magitude of α, sum to 1 row-wise, the higher the bar, the bigger the magnitude. $3\text{DConv}_{\text{att}}$ generates a more faithful description with a much richer content than Figure 10. It even learns to generate a rare work "teasing".

8.2. Caption generation and α visualization on DVS testset

This section illustrates on DVS, a much more challenging dataset. See the following figures for detailed explaination of soft-attention applied on videos with different properties.

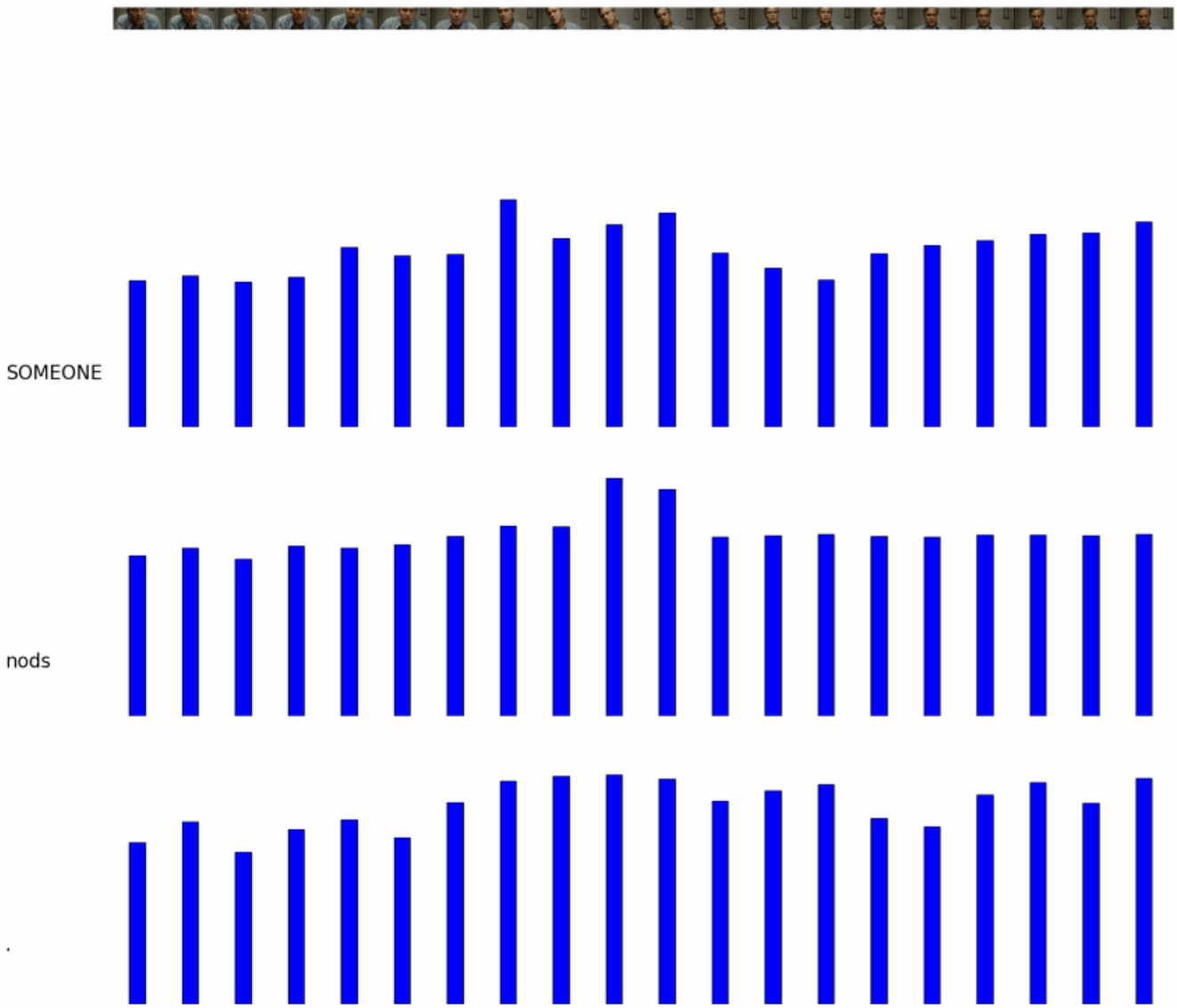

Figure 12. Model type: Basic + Global. The model tends to produce a smooth distribution in α row-wise, due to the uniformity of the scene with a slowly changing continuous shot.

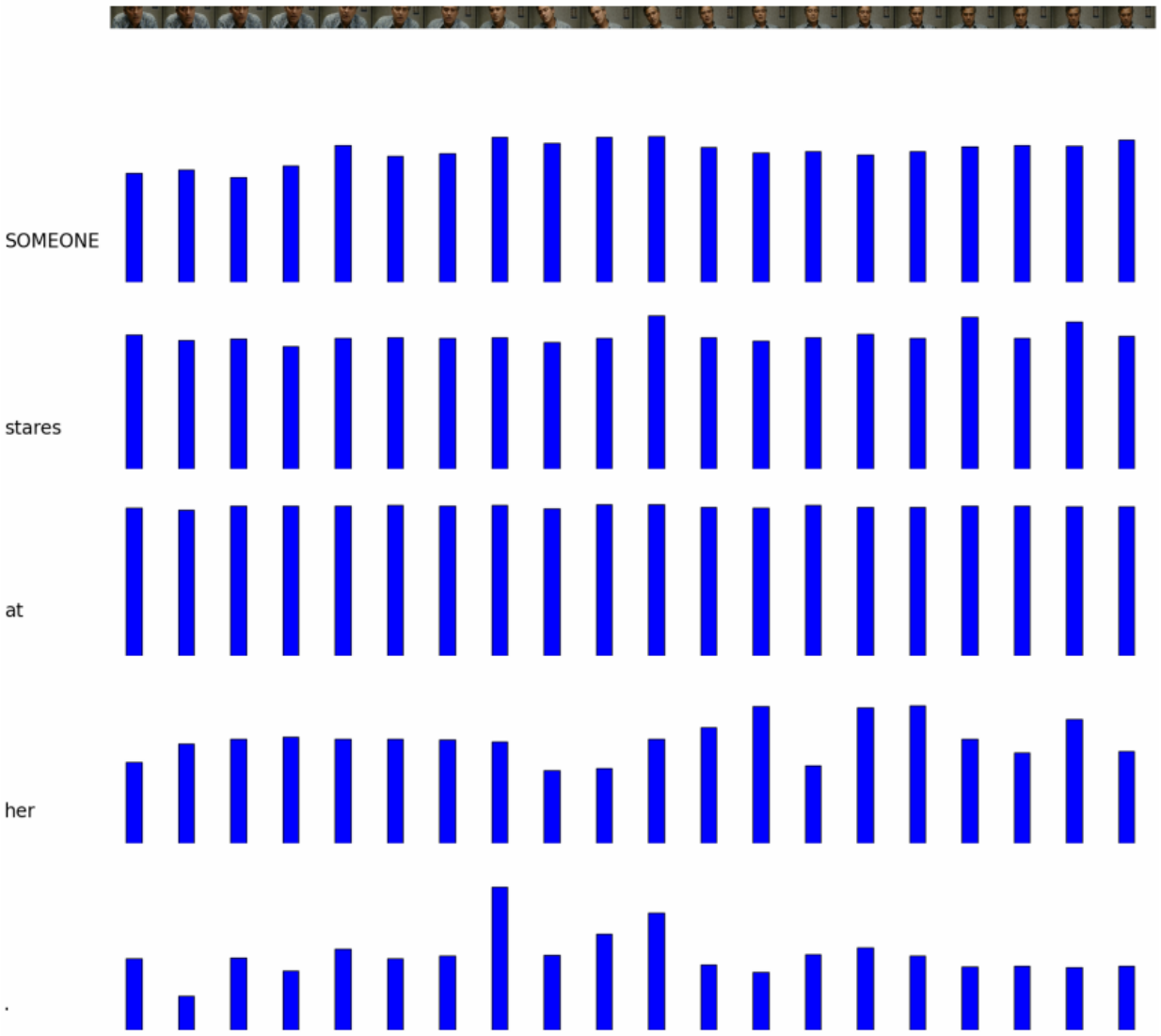

Figure 13. Model type: Basic + Local + Global. The model learns a smooth α on the slowly changing scene. It captures a different action from Basic + Global in Figure 12.

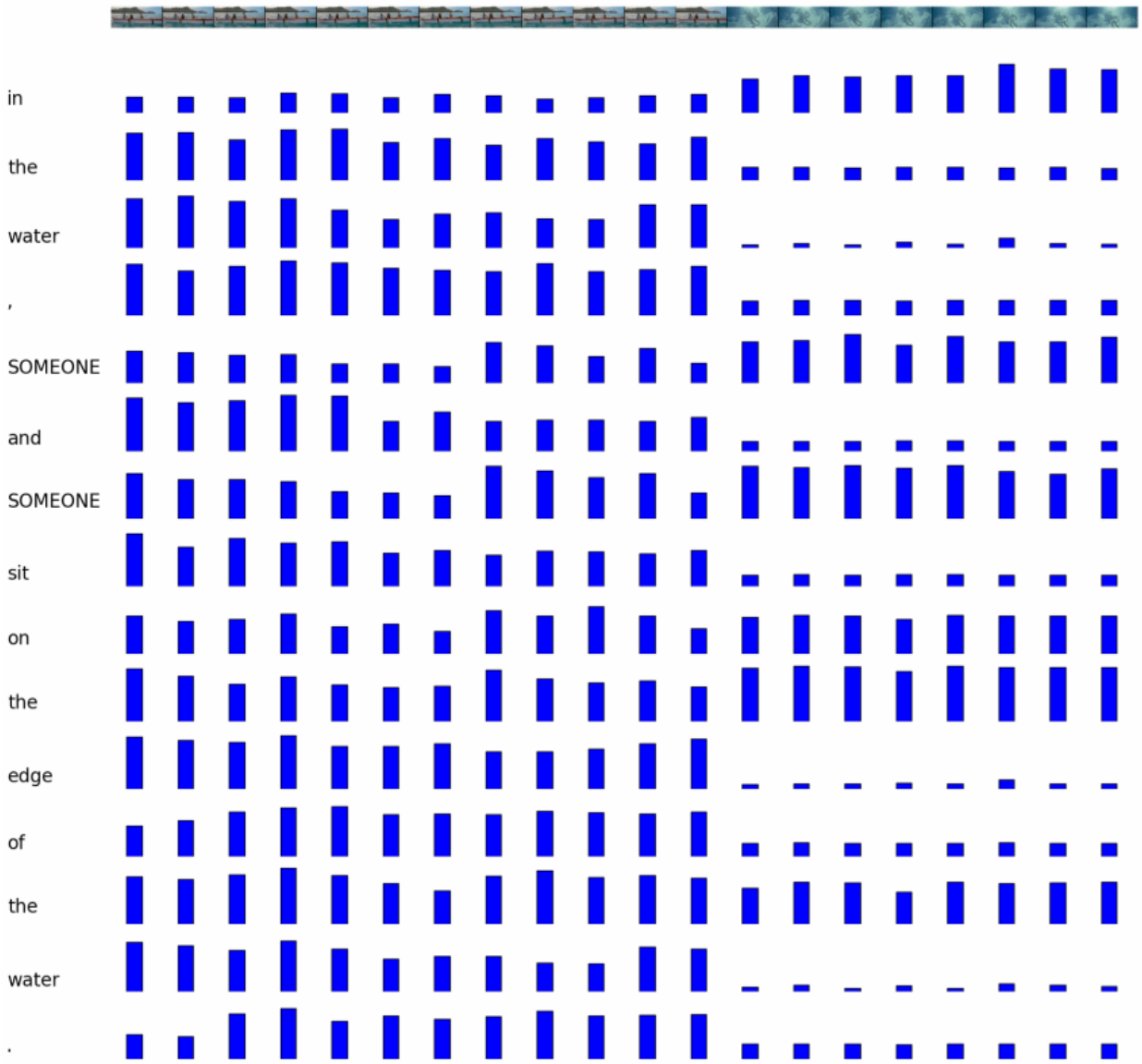

Figure 14. Model type: Basic + Global. α also reflects the sudden transition between two shots.

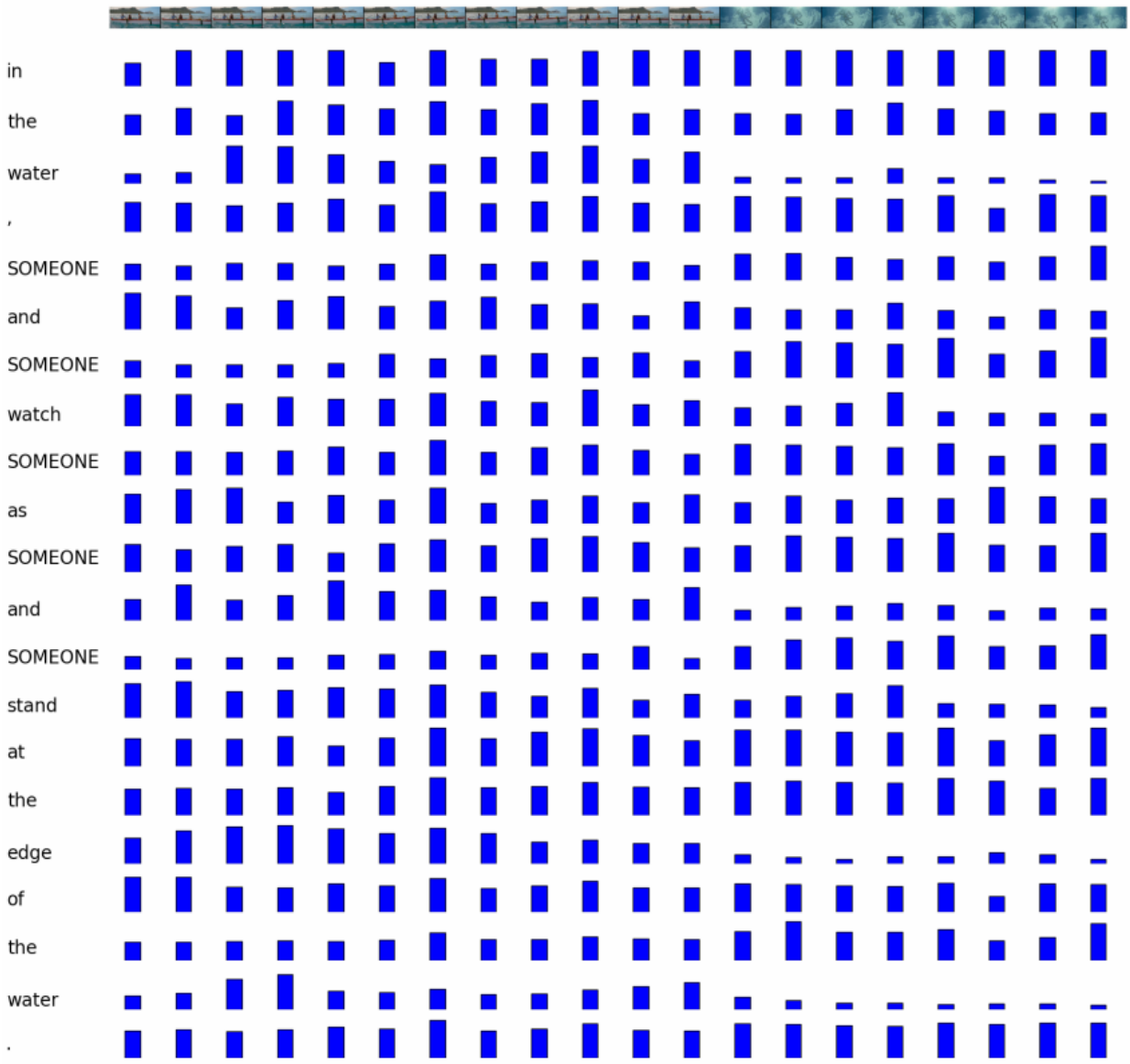

Figure 15. Model type: Basic + Local + Global. The learned model generates a more sophiscated description than Figure 14, attempting to incoporate character-level interaction inside the first part of the scene.

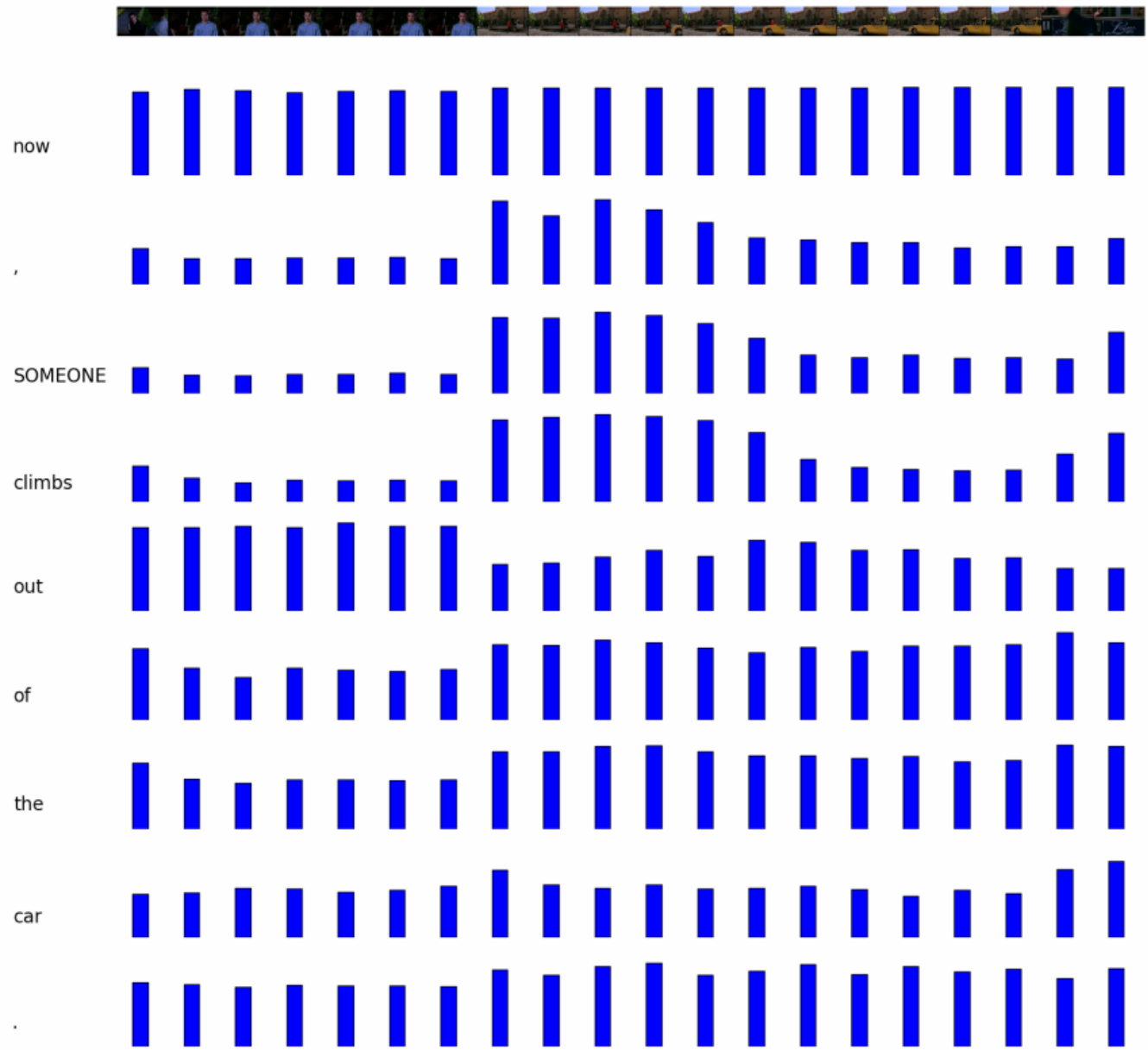

Figure 16. Model type: Basic + Global. The model seems to focus on the second shot of the scene at the beginning, yet the part of the generated caption "out of the car" distributes a decent amount of its attention on the first scene as well. This may due to the fact that the memory of decoding LSTM already contains the information of almost the entire scene (two shots).

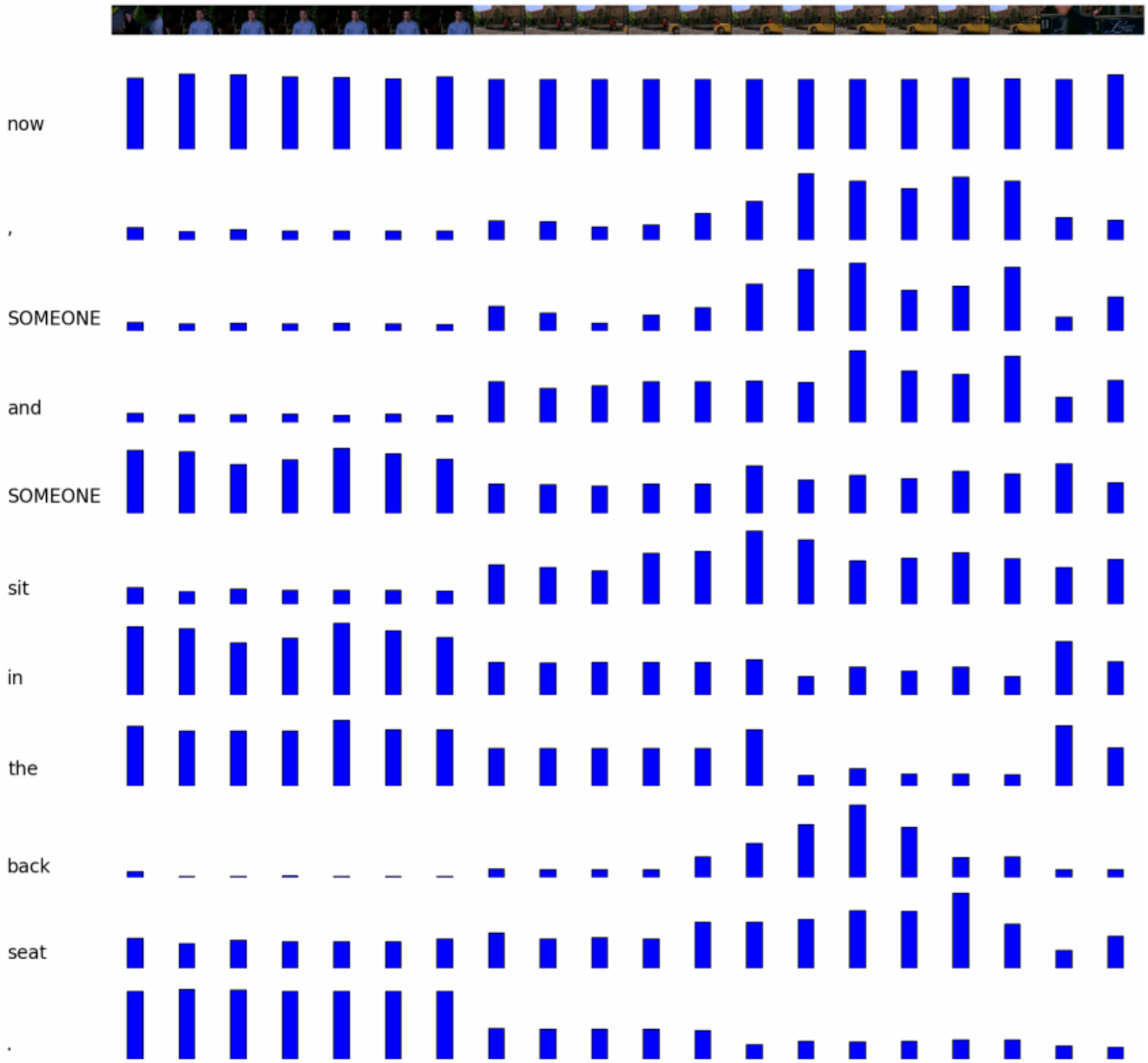

Figure 17. Model type: Basic + Local + Global. The learned model generates a more sophiscated description than Figure 16. The model focuses on the car in the second shot when generating “sit”, “back seat”. When generating two “SOMEONE”, it divides its attentio among two shots.

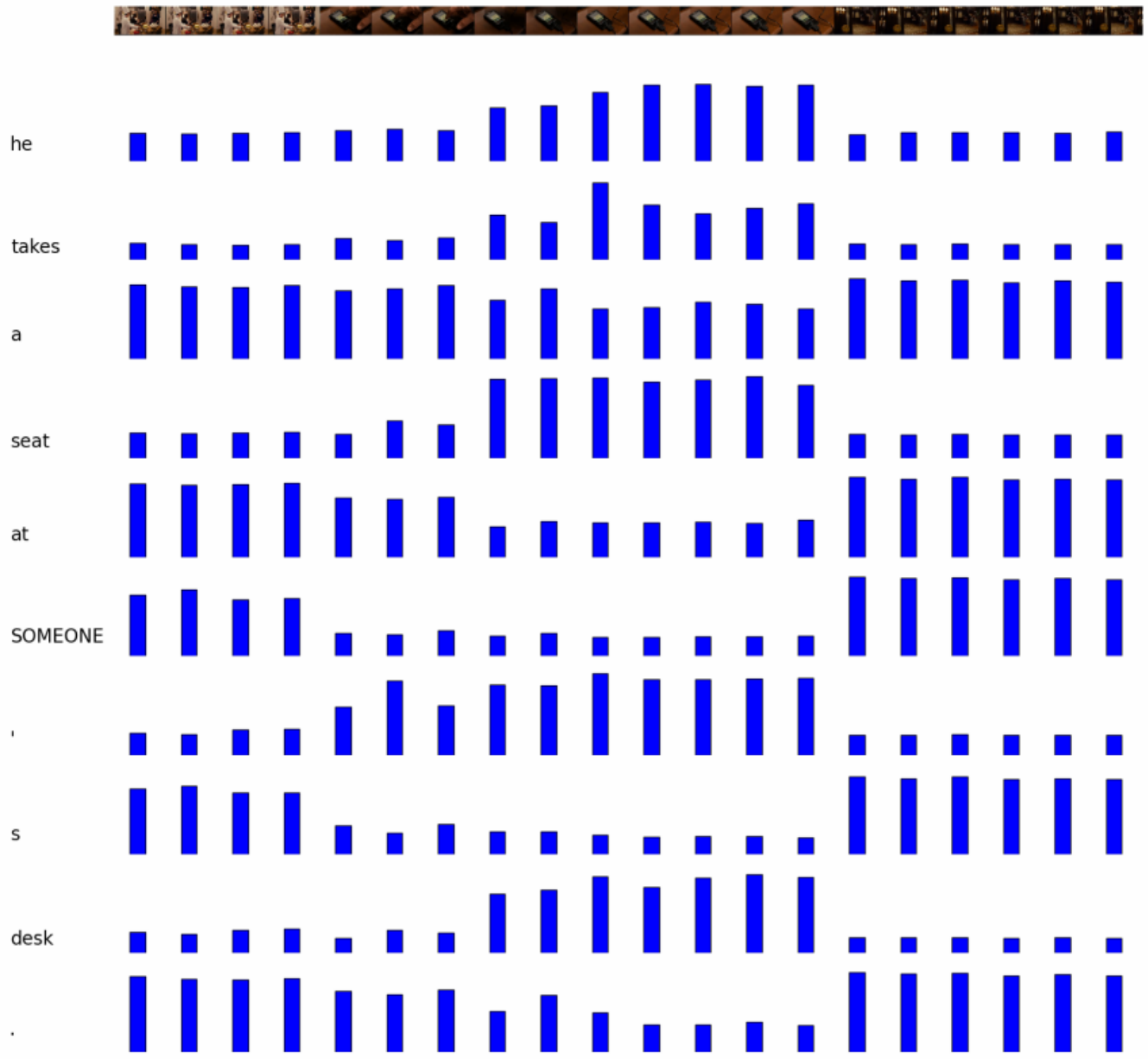

Figure 18. Model type: Basic + Global. The description is argubly not very accurate

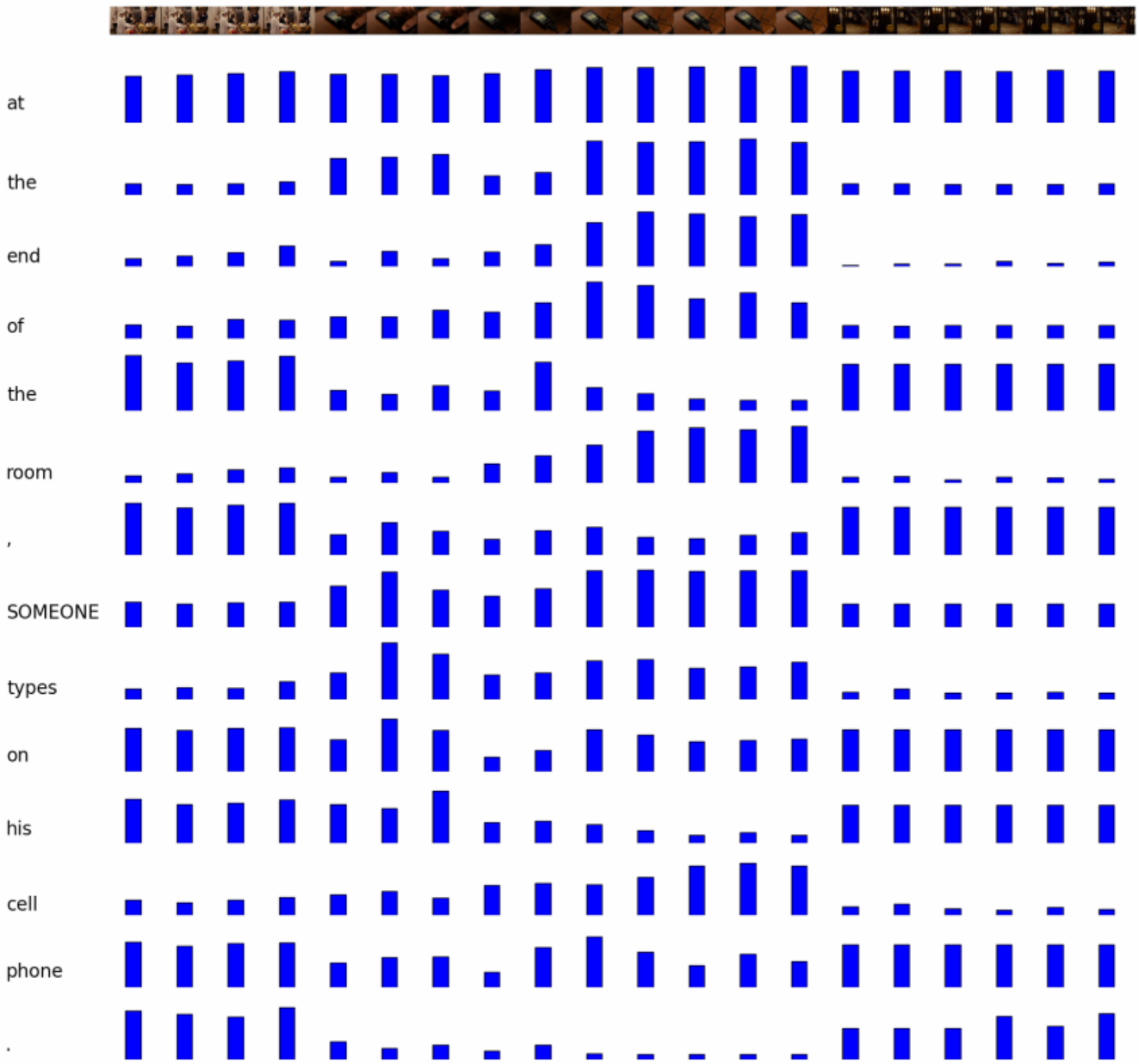

Figure 19. Model type: Basic + Local + Global. With the help of additional features, the model successfully describes the cell phone and the room, a much faithful description than Figure 18