

Video Description using Bidirectional Recurrent Neural Networks

Álvaro Peris¹, Marc Bolaños^{2,3}, Petia Radeva^{2,3}, and Francisco Casacuberta¹

¹ PRHLT Research Center, Universitat Politècnica de València, Valencia (Spain)

² Universitat de Barcelona, Barcelona (Spain)

³ Computer Vision Center, Bellaterra (Spain)

{lvapeab, fcn}@prhlt.upv.es

{marc.bolanos, petia.ivanova}@ub.edu

Abstract. Although traditionally used in the machine translation field, the encoder-decoder framework has been recently applied for the generation of video and image descriptions. The combination of Convolutional and Recurrent Neural Networks in these models has proven to outperform the previous state of the art, obtaining more accurate video descriptions. In this work we propose pushing further this model by introducing two contributions into the encoding stage. First, producing richer image representations by combining object and location information from Convolutional Neural Networks and second, introducing Bidirectional Recurrent Neural Networks for capturing both forward and backward temporal relationships in the input frames.

Keywords: Video Description, Neural Machine Translation, Birectional Recurrent Neural Networks, LSTM, Convolutional Neural Networks.

1 Introduction

Automatic generation of image descriptions is a recent trend in Computer Vision that represents an interesting, but difficult task. This has been possible due to the dramatic advances in Convolutional Neural Network (CNN) models that allowed to outperform the state-of-the-art algorithms in many computer vision problems: object recognition, object detection, activity recognition, etc. Generating descriptions of videos represents an even more challenging task that could lead to multiple applications (e.g. video indexing and retrieval, movie description for multimedia applications or for blind people or human-robot interaction). We can imagine the amount of data generated in YouTube (every day people watch hundreds of millions of hours of video and generate billions of views) and how video description would help to categorize them, provide an efficient retrieval mechanism and serve as a summarization tool for blind people.

However, the problem of video description generation has several properties that make it specially difficult. Besides the significant amount of image information to analyze, videos may have a variable number of images and can be described with sentences of different length. Furthermore, the descriptions

of videos use to be high-level summaries that not necessarily are expressed in terms of the objects, actions and scenes observed in the images. There are many open research questions in this field requiring deep video understanding. Some of them are how to efficiently extract important elements from the images (e.g. objects, scenes, actions), to define the local (e.g. fine-grained motion) and global spatio-temporal information, determine the salient content worth to describe, and generate the final video description. All these specific questions need the attention of computer vision, machine translation and natural language understanding communities in order to be solved.

In this work, we propose to enrich the state-of-the-art architecture using bidirectional neural networks for modeling relationships in two temporal directions. Furthermore, we test the inclusion of supplementary features, which help to detect contextual information from the scene where the video takes place.

2 Related Work

Although the problem of video captioning recently appeared thanks to the new learning capabilities offered by Deep Learning techniques, the general pipeline adopted in these works resembles the traditional encoder-decoder methodology used in Machine Translation (MT). The main difference is that, in the encoder step, instead of generating a compact representation of the source language sentence, we generate a representation of the images belonging to the video.

MT aims to automatically translate text or speech from a source to a target language. Within the last decades, the prevailing approach is the statistical one [5]. The application of connectionist models in the area has drawn much the attention of researchers in the last years. Moreover, a new approach to MT has been recently proposed: the so-called Neural Machine Translation, where the translation process is carried out by a means of a large Recurrent Neural Network (RNN) [11]. These systems rely on the encoder-decoder framework: an encoder RNN produces a compact representation of an input sentence in the source language, and the decoder RNN takes this representation and generates the corresponding target language sentence. Both RNNs usually make use of gated units, such as the popular Long Short-term Memory (LSTM) [4], in order to cope with long-term relationships.

The recent reintroduction of Deep Learning in the Computer Vision field through CNNs [6], has allowed to obtain new and richer image representations compared to the traditional hand-crafted ones. These networks have demonstrated to be a powerful tool to extract feature representations for several kinds of computer vision problems like on objects [10] or scenes [18] recognition. Thanks to the CNNs ability to serve as knowledge transfer mechanisms, they have also been usually used as feature extractors.

The majority of the works devoted to generate textual descriptions from single images also follow the encoder-decoder architecture. In the encoding stage, they apply a combination of CNN and LSTM for describing the input image. In the decoding stage, an LSTM is in charge of receiving the image information

and generating, word by word, a final description of the image [15]. The problem of video captioning is similar. Seminal works applied methodologies inspired by classical MT [9]. Nevertheless, more recent works following the encoder-decoder approach, obtained state-of-the-art performances [14,16].

We present a new methodology for natural language video description that makes use of deeper structures and a double-way analysis of the input video. We propose to use as a base architecture the one introduced in [16]. On the top of it, our contributions are twofold. First, we produce richer image representations by combining complementary CNNs for detecting objects and contextual information from the input images. Second, we introduce a Bidirectional LSTM (BLSTM) network in the encoding stage, which has the ability to learn forward and backward long-term relationships on the input sequence. Moreover, we make our code public¹.

3 Methodology

An overview of our proposal is depicted in Fig. 1. We propose an encoder-decoder approach consisting of four stages, using both CNNs and LSTMs for describing images and for modeling their temporal relationship, respectively.

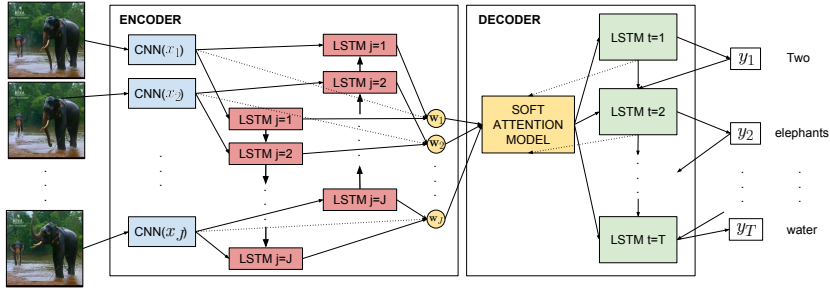


Fig. 1: General scheme of our proposed methodology.

First (blue in the scheme), we apply two state of the art CNN models for extracting complementary features on each of the raw images from the video.

Second (red in the scheme), considering we need to describe the actions performed in consecutive frames, we apply a BLSTM for capturing temporal relationships and complementary information by taking a look at the action in a forward and in a backward manner.

Third (yellow in the scheme), the two output vectors from forward and backward LSTM models of the previous step are concatenated together with the CNN output for each image and are fed to a soft attention model in the decoder. This model decides on which parts of the input video should focus for emitting the next word, considering the description generated so far.

Fourth (green in the scheme), an LSTM network generates the video caption from the representation obtained in previous stages. The variable-length caption

¹ <https://github.com/lvapeab/ABiViRNet>

is obtained word by word, using a softmax function on the top of this LSTM network.

3.1 Encoder

Given the video description problem, in the encoding stage we need to properly characterize the video for 1) understanding which kind of objects and structures appear in the images, and 2) modeling their relationships and actions along time.

For tackling the first part of the problem, several kinds of pretrained CNNs may be used for describing the images, which can be distinguished by the different architectures or by the different datasets used for training. Although an extended comparison and combinations of models could be used for applying this characterization, we propose combining object and context-related information. For this purpose we use a GoogleNet architecture [12] separately trained on two datasets, one for objects (ILSVRC dataset [10]), and the other for scenes (Places 205 [18]).

The combination of these two kinds of data can inform about the objects appearing and their surroundings, being ideal for the problem at hand. For a given video, the CNNs generate a sequence \mathbf{V}_c of J d -dimensional feature vectors, $\mathbf{x}_1, \dots, \mathbf{x}_J$ with $\mathbf{x}_j \in \mathbb{R}^d$ for $1 \leq j \leq J$, where J is the number of frames in the video.

To solve the second problem, a BLSTM processes the sequence \mathbf{V}_c , generating a new sequence $\mathbf{V}_{bi} = \mathbf{v}_1, \dots, \mathbf{v}_J$ of J vectors. BLSTM networks are composed of two independent LSTM layers namely, forward and backward. Both layers are analogue, but the latter processes the input sequence reversed in time.

LSTM networks have, in addition to the classical hidden state, a memory state. Let \mathbf{v}_j^f be the forward layer hidden state at the time-step j , and let \mathbf{c}_j^f be its memory state. The hidden state \mathbf{v}_j^f is computed as \mathbf{c}_j^f controlled by an output gate \mathbf{o}_j^f . The current memory state depends on an updated memory state, and on the previous memory state, \mathbf{c}_{j-1}^f , respectively modulated by the forget and input gates, \mathbf{f}_j^f and \mathbf{i}_j^f . The updated memory state $\tilde{\mathbf{c}}_j^f$ is obtained by applying a logistic non-linear function to the input and the previous hidden state. Each LSTM gate has associated two weight matrices, accounting for the input and

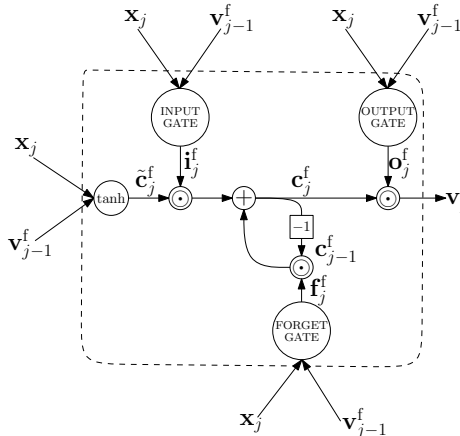


Fig. 2: Forward layer LSTM unit for the encoder. The output depends on the previous hidden state (\mathbf{v}_{j-1}^f) and the current feature vector from the video extracted by the CNN (\mathbf{x}_j). Input, output and forget gates module the amount of information that flows across the unit.

the previous hidden state. Such matrices must be estimated on a training set. Figure 2 shows an illustration of an LSTM unit. The same architecture applies to the backward layer, but dependencies flow from the next time-step to the previous one. Since forward and backward layers are independent, they have different weight matrices to estimate.

Each feature vector \mathbf{v}_j computed by the BLSTM results as the concatenation of the forward and backward hidden states: $\mathbf{v}_j = [\mathbf{v}_j^f; \mathbf{v}_j^b] \in \mathbb{R}^{2 \cdot D}$ for $1 \leq j \leq J$, being D the size of each forward and backward hidden state.

Finally, the encoder combines the sequences \mathbf{V}_c and \mathbf{V}_{bi} by concatenating the vectors from the CNN and from the BLSTM, producing a final sequence \mathbf{V} of J feature vectors $\mathbf{w}_1, \dots, \mathbf{w}_J$, $\mathbf{w}_j = [\mathbf{x}_j; \mathbf{v}_j] \in \mathbb{R}^{d+2 \cdot D}$ for $1 \leq j \leq J$.

3.2 Decoder

The decoder is an LSTM network, which acts as a language model, conditioned by the information provided by the encoder. This network is equipped with an attention mechanism [1,16]: a soft alignment model, implemented as a single-layered perceptron, that helps the decoder to know *where* to look at for generating each output word. Given the sequence \mathbf{V} generated by the encoder, at each decoding time-step t the attention mechanism weights the J feature vectors and combines them into a single context vector $\mathbf{z}_t \in \mathbb{R}^{d+2 \cdot D}$.

The decoder LSTM is defined similarly to the forward layer from the encoder, but it takes into account the previously generated word and the context vector from the attention mechanism, in addition to its previous hidden state. The last word representation is provided by a word embedding matrix $\mathbf{E} \in \mathbb{R}^{m \times V}$, being m the size of the word embedding and V the size of the vocabulary. \mathbf{E} is estimated together with the rest of the model parameters.

A probability distribution over the vocabulary of output words is defined from the hidden state \mathbf{h}_t , by means of a softmax function. This function represents the conditional probability of a word given an input video \mathbf{V} and its history (the previously generated words): $p(y_t | y_1, \dots, y_{t-1}, \mathbf{V})$. Following [11], a beam-search method is used to find the caption with highest conditional probability.

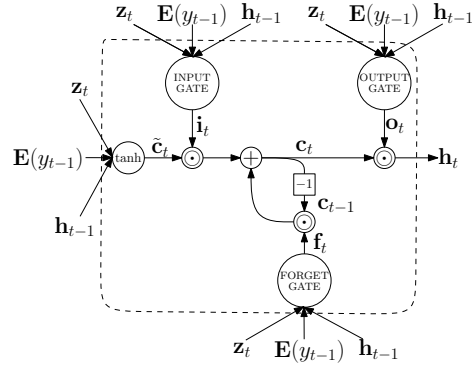


Fig. 3: Decoder LSTM unit. The output depends on the previous hidden state (\mathbf{h}_t), the word embedding of the previously generated word ($\mathbf{E}(y_{t-1})$) and the context vector provided by the attention mechanism (\mathbf{z}_t).

4 Results

In this section we describe the datasets and metrics used for evaluating and comparing our model to the video captioning state of the art.

4.1 Dataset

The **Microsoft Research Video Description Corpus** (MSVD) [2] is a dataset composed of 1970 open domain clips collected from YouTube and annotated using a crowd sourcing platform. Each video has a variable number of captions, written by different users. We used the splits made by [14,16], separating the dataset in 1200 videos for training, 100 for validation and the remaining 670 for testing. During training, the clips and each of their captions were treated separately, accounting for a total of more than 80,000 training samples.

4.2 Evaluation Metrics

In order to evaluate and compare the results of the different models we used the standardized COCO-Caption evaluation package [3], which provides several metrics for text description comparison. We used three main metrics, all of them presented from 0 (minimum quality) to 100 (maximum quality):

BLEU [8]: this metric compares the ratio of n-gram structures that are shared between the system hypotheses and the reference sentences.

METEOR [7]: it was introduced to solve the lack of the recall component when computing BLEU. It computes the F1 score of precision and recall between hypotheses and references.

CIDEr [13]: similarly to BLEU, it computes the number of matching n-grams, but penalizes any n-gram frequently found in the whole training set.

4.3 Experimental Results

On all the tests we used a batch size of 64, the learning rate was automatically set by the Adadelta [17] method and, as the authors in [16] reported, we applied a frame subsampling, picking only one image every 26 frames for reducing the computational load. The parameters of the network were randomly initialized. An evaluation on the validation set was performed every 1000 updates. The learning process was stopped when the reported error increased after 5 evaluations.

For each configuration we run 10 experiments. At each of them, we randomly set the value of the critical model hyperparameters. Such hyperparameters and their tested ranges are $m \in [300, 700]$, $|\mathbf{h}_t| \in [1000, 3000]$. When using the BLSTM encoder, we performed an additional selection on $|\mathbf{v}_j| \in [100, 2100]$.

For each configuration, the best model with respect to the BLEU measure on the validation set was selected. In Table 1 we report the results of the best models on the test set. The first row correspond to the result obtained with our system with the object features from [16]. The configurations reported below

Model	BLEU [%]	METEOR [%]	CIDEr [%]
Objects*	51.5	32.5	66.0
Objects + BLSTM	53.6	32.6	66.4
Objects + Scenes	52.6	32.5	67.0
Objects + Scenes + BLSTM	52.8	31.3	67.2

Table 1. Text generation results for each model on the MSVD dataset. The results below the horizontal line are our proposals. *Model from [16] only with *Object* features evaluated on our system.

the horizontal line are our proposals, where *Scenes* indicates we use scene-related features concatenated to *Objects* and *BLSTM* denotes the use of the additional BLSTM encoder.

5 Discussion and Conclusions

Analyzing the obtained results, a clear improvement trend can be derived when applying the BLSTM as a temporal inference mechanism. The BLSTM addition when using *Objects* features allows to improve the result on all metrics, obtaining a benefit of more than 2 BLEU points. Adding scenes-related features also slightly improves the result, although it is not as remarkable as the BLSTM improvement. The combination of *Objects+Scenes+BLSTM* offers the best CIDEr performance, nevertheless, this result is slightly below the *Objects+BLSTM* one on the other metrics. This behaviour is probably due to the significant increase on the number of parameters to learn. It should be investigated whether the reduction of the number of parameters by reducing the size of the CNN features, or the use of larger datasets could lead to further improvements.

In conclusion, we have presented a new methodology for natural language video description that takes profit from a bidirectional analysis of the input sequence. This architecture has the ability to learn forward and backward long-term relationships on the input images. Additionally, the use of complementary object and scene-related image features has proved to obtain a richer video representation. The improvements have allowed the method to outperform the state-of-the-art results in the problem at hand.

These results suggest that deep structures help to transfer the knowledge from the input sequence of frames to the output natural language caption. Hence, the next step to take must delve into the application deeper modeling structures, such as 3D CNNs and multi-layered LSTM networks.

Acknowledgments. This work was partially funded by TIN2012-38187-C03-01, SGR 1219, PrometeoII/2014/030 and by a travel grant by the MIPRCV network. P. Radeva is partially supported by an ICREA Academia2014 grant. We acknowledge NVIDIA for the donation of a GPU used in this work.

References

1. Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations (arXiv:1409.0473)*, 2015.
2. David L Chen and William B Dolan. Collecting highly parallel data for phrase evaluation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (Vol. 1)*, pages 190–200, 2011.
3. Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015.
4. Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
5. Philipp Koehn. *Statistical Machine Translation*. Cambridge University Press, 2010.
6. Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.
7. Alon Lavie and Michael J Denkowski. The meteor metric for automatic evaluation of machine translation. *Machine translation*, 23(2-3):105–115, 2009.
8. Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318, 2002.
9. Marcus Rohrbach, Wei Qiu, Ivan Titov, Stefan Thater, Manfred Pinkal, and Bernt Schiele. Translating video content to natural language descriptions. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, pages 433–440, 2013.
10. Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.
11. Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, volume 27, pages 3104–3112. 2014.
12. Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.
13. Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4566–4575, 2015.
14. Subhashini Venugopalan, Marcus Rohrbach, Jeffrey Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko. Sequence to sequence-video to text. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4534–4542, 2015.
15. Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3156–3164, 2015.
16. Li Yao, Atousa Torabi, Kyunghyun Cho, Nicolas Ballas, Christopher Pal, Hugo Larochelle, and Aaron Courville. Describing videos by exploiting temporal structure. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4507–4515, 2015.

17. Matthew D Zeiler. Adadelata: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.
18. Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. In *Advances in neural information processing systems*, pages 487–495, 2014.