Automatic Video Description Generation via LSTM with Joint Two-stream Encoding

Chenyang Zhang and Yingli Tian

Department of Electrical Engineering
The City College of New York
New York, New York 10031
Email: {czhang10@citymail, ytian@ccny}.cuny.edu

***Abstract*—In this paper, we propose a novel two-stream framework based on combinational deep neural networks. The framework is mainly composed of two components: one is a parallel two-stream encoding component which learns video encoding from multiple sources using 3D convolutional neural networks and the other is a long-short-term-memory (LSTM)-based decoding language model which transfers the input encoded video representations to text descriptions. The merits of our proposed model are: 1) It extracts both temporal and spatial features by exploring the usage of 3D convolutional networks on both raw RGB frames and motion history images. 2) Our model can dynamically tune the weights of different feature channels since the network is trained end-to-end from learning combinational encoding of multiple features to LSTM-based language model. Our model is evaluated on three public video description datasets: one YouTube clips dataset (Microsoft Video Description Corpus) and two large movie description datasets (MPII Corpus and Montreal Video Annotation Dataset) and achieves comparable or better performance than the state-of-the-art approaches in video caption generation.**

I. Introduction

With the explosive growth of online visual media sharing platforms such as YouTube, the scale and variety of visual medias have grown rapidly and have become a significant component of everyone's life. For example, the time adults spend watching online videos each day has increased about 260% from 2011 to 2015. This trend has brought a significant number of challenges to visual retrieval and organization. With such a large-scale corpus of videos, manually tagging and describing them is intractable. Therefore, how to automatically and effectively label, describe and understand the enormous visual corpus becomes a very popular research topic in recent years. Single image description has attracted many interests [7], [9], [5], [1] and has established several standard schemes such as *encoder-decoder framework* and *recurrent network-based language model*.

A standard encoder-decoder framework is composed of two steps: 1) visual features are extracted from raw images and a mapping between visual features and semantic components, such as words, phrases, sentences, *etc.*, is constructed. 2) The mapped features are fed into a language model, such as templates [21], [12], [8] or Recurrent Neural Networks [5], [19], [20]. This encoder-decoder framework is also valid in video description tasks, despite the fact that the input becomes multiple frames instead of a single image.

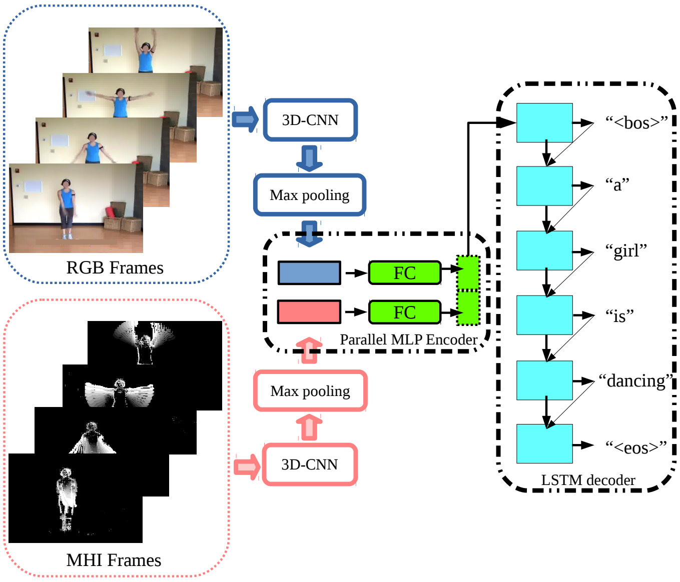

Fig. 1. Our proposed automatic video describing approach is composed of two parts: 1) parallel fully connected layer (FC) encoder which jointly learns video representation from two streams of video sequences (RGB frames and motion history images (MHI).) 2) An LSTM-based decoder which outputs a sentence word by word from the learned video representation. Pre-trained 3D-CNNs are employed for feature extraction. The rest of the networks are trained end-to-end. "<bos>" and "<eos>" are padded special tokens indicating "beginning of a sentence" and "end of a sentence", respectively.

To extend the framework from a single image to a sequence of video frames, multiple approaches have been proposed to modify the encoder part such as using average-pooling over frames [18], applying different weights to a fixed number of sampled frames [20]. Venugopalan *et al.* [17] also propose to modify the decoder part by adapting the decoding RNN to handle a sequence of encoded frames. However, most of the previous approaches only rely on one single feature resource. While [17] attempts to merge both visual appearance features and optical-flow features, only a late-fusion is employed by assigning two pre-defined empirical weights to the two feature channels, which limits the power of the model to learn a better combination by itself.

Our work contributes as follows: 1) a novel parallel encoder integrating LSTM-based language model is proposed. The model combines multiple pre-trained C3D feature channels and automatically learns a proper representation which avoids both loosing too much learning capability or introducing too much entanglement. 2) We observe through experiments that the two feature channels (RGB and MHI) contain complementary information, although C3D has the potential to capture local temporal structure, motion-centric MHI feature practically contributes more motion information in the system.

II. Approach

The structure of our proposed framework is illustrated in Figure 1, which is composed of a two-stream encoder and a language generator model which is built on a single layer of RNNs with LSTM cells.

A. RNNs with LSTM cells

Firstly let us briefly recall the long-short-term-memory (LSTM) variant of recurrent neural network (RNN) and its connection with video captioning. A RNN is a neural network which includes an internal state which depends on both current input and output from the last time-step. Due to that feature, RNNs are capable to model complex temporal dynamics and used in temporal sequences modeling, such as speech recognition, language modeling, machine translation and image captioning. Because RNN has the ability to remember previous inputs and outputs, it is suitable to generate a sequence of words which follows the patterns of human (natural language) as well as conditional on the encoded input image or video. However, simple RNN often fails to remember long-term context [2].

To overcome the issue mentioned above, RNN is modified by replacing its internal state variables to a series of "gate" state variables as shown in the following formula:

$$\begin{aligned} i_t &= \sigma(W_i x_t + U_i h_{t-1} + b_i) \\ o_t &= \sigma(W_o x_t + U_o h_{t-1} + b_o) \\ f_t &= \sigma(W_f x_t + U_f h_{t-1} + b_f) \\ \hat{C_t} &= tanh(W_c x_t + U_c h_{t-1} + b_c) \\ C_t &= i_t \odot \hat{C_t} + f_t \odot C_{t-1} \\ h_t &= o_t \odot tanh(C_t) \end{aligned} \tag{1}$$

where $\odot$ is element-wise product and $\sigma(\cdot)$ denotes the sigmoid activation function; x_t is the input at time step t to the LSTM cell; W_i, W_f, W_c, W_o, U_i, U_f, U_c, and U_o are weight matrices assigned to different state parameters; b_i, b_f, b_c and b_o are bias vectors; i_t, o_t, f_t, C_t and h_t denote input gate, output gate, forget gate, cell state and hidden state, respectively. $\hat{C_t}$ represents the candidate cell state before combining with the previous cell state (C_{t-1}) and the forget gate.

The input gate values i_t and forget gate values h_t are computed based on the hidden state of last LSTM cell (h_t) and the current input vector x_t. The forget gate can code what memory the current cell drops and the input gate is responsible for what the encoded input is visible for computing the current cell value C_t. And finally the output gate values o_t and the current hidden state values h_t are computed based on both the cell state values C_t and o_t, which can be treated as an encoded version of the output o_t.

With enough training data, RNNs with LSTM architecture are well-suited to learn very long time correspondences between important events. In the decoder perspective, LSTM-based RNNs can generate very long sentences which look like natural language. This feature increases the capacity of LSTM-based RNNs to describe more complex relationships and events in video frames.

The language model. In this paper, a layer of LSTM-based RNNs is employed as sentence generating decoder. The RNNs predict a probabilistic distribution over the output sentence conditional on the input video description. Suppose the input video description is denoted as X and the vocabulary dictionary is V, the output sentence is:

$$Y = \{w_0, w_1, ..., w_n\}, \quad \text{and} \quad w_i \in V, \tag{2}$$

where w_0 and w_n are padded special tokens "¡bos¿" and "¡eos¿" representing the "beginning" and the "end" of a sentence, and all other w_is are encoded words in the vocabulary. The probabilistic distribution the RNNs will predict is:

$$\begin{aligned} p(Y|X) &= p(w_0, w_1, ..., w_n|X) \\ &= p(w_1, ..., w_n|X) = \prod_{t=1}^{n} p(w_t|h_{t-1}, y_{t-1}), \end{aligned} \tag{3}$$

where h_{t-1} denotes the hidden state computed by LSTM from the previous time step. Since all sentences start with "¡bos¿", the prediction actually starts at $t = 1$. Note that in each time step, the LSTM cell computes the values of h_t by given the history pair $\{h_{t-1}, y_{t-1}\}$. And the word prediction given the current hidden state h_t is calculated by a softmax classifier which is computed based on the current LSTM hidden state at each time step.

B. Video feature extraction

Video description is different from image description because videos include temporal information. Therefore, to generate more meaningful descriptions, both visual appearance and temporal features should be jointly considered. For example, in the output sentence of Figure 1 (*"a **girl** is **dancing**."*), *"**girl**"* can be inferred from each video frame but *"**dancing**"* is more appropriate to be inferred from temporal structures.

While [17] focuses on modeling sequential inputs and [22] focuses on pooling strategy, our model tackles this problem from two aspects:

1) Video features from both RGB frames and MHI frames are extracted which ensure the model to capture information from both visual appearance and temporal motion.

2) 3D-CNN instead of 2D-CNN networks are employed to generate the features to further extract spatial-temporal dynamics from both feature channels.

We employ C3D implementation [16] as the 3D-CNN-based feature extraction layer. The network is pre-trained on a large-scale action recognition dataset (Sport1m [6]). For efficiency, the C3D networks are not fine-tuned in this work. The activations of the top fully-connected layer (4096 dimensions) are extracted as the final feature representation for a 16-frame-long video snippet. For any video longer than 16 frames, we segment it into non-overlapping 16-frame-long snippets and

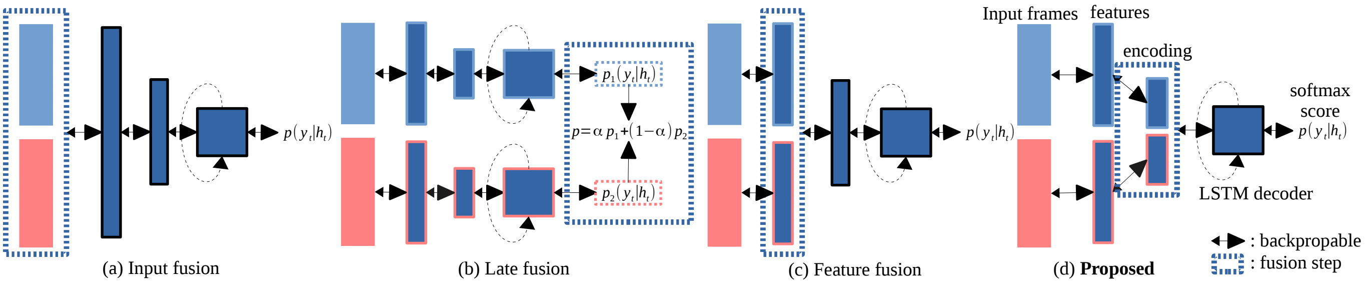

Fig. 2. Comparative illustration of four fusion paradigms discussed in this paper. (a) Input fusion: different video signals are combined at the input level, which is adopted in [22], where they combine histogram of gradients (HoG), histogram of optical-flow (HoF) and motion boundary history (MBH). (b) Late fusion: two models are trained separately and only the prediction scores are fused in the final stage using a non-learned weight (α), which is adopted in [17]. (c) is similar to (a) except the fusion takes place after the feature extraction. (d) is our proposed fusion model, where we modify the encoding layer into a parallel architecture and fusion takes place in the encoding layer. Compared to (a) and (c), our model trains different encoders for different feature channels, which avoids entanglement. Compared to (b), our model can automatically learn how to obtain a good fusion.

combine the features using max-pooling. Compared to existing work using average pooling [18], max-pooling is more desirable than average-pooling because it potentially can capture more informative events in the video sequences.

C. Two-stream encoder

After feature extraction, the input video is represented by two feature vectors: C3D-RGB and C3D-MHI. Each feature vector has 4096 dimensions. To encode the two feature vectors into the encoding space of the following LSTM-based encoder, we propose a parallel encoder to jointly learn a mapping from the two feature vectors into the encoding space. There are several encoding schemes for this task, for example, canonical correlation analysis (CCA) is used in [7] to learn the linear mapping between feature space and the word-encoding space (in [7], the word-encoding space is spanned by fisher vectors of word2vec embedding [10]). Unfortunately, CCA is not applicable in our framework because: 1) linear-CCA can easily overfit and kernel-CCA is too expensive to compute for large-scale dataset. 2) It is difficult to integrate CCA to the framework and to keep the end-to-end training trait.

Our proposed framework employs two parallel fully connected layers to learn the mappings. The conceptual visual comparison of our proposed fusion scheme is compared with two other schemes used in [22] and [17] in Figure 2. In [22], multiple channels of features are computed from the input video: histogram of gradients (HoG), histogram of optical-flow (HoF) and motion boundary history (MBH). The fusion method used in [22] is an early input-level fusion (Figure 2 (a)), which concatenates all inputs together and feeds them to the CNNs. We argue that this early fusion mechanism is problematic that different feature channels have different patterns and characteristics, thus a holistic feature extraction CNN is not as appropriate as separate flows of CNNs for different feature channels. In [17], the authors use separate CNN flows for different feature channels and apply a late decision level fusion (Figure 2 (b)). However, the late fusion depends on empirical settings of weights so that it discards the capability of the framework to automatically learn the combination. Our framework employs a novel parallel encoding level fusion (Figure 2 (d)), and two separate fully connected layers for two streams to map the input feature from a high dimension to a lower dimension encoding space. ReLu nonlinearity is applied after each fully connected layer. Compared to [22], our proposed fusion scheme is more dedicated to model each feature channel, which avoids feature entanglement; compared to [17], our fusion scheme does not block the training flow so that it can automatically learn how to jointly encoding the two feature channels. In other words, our proposed scheme seeks a proper tradeoff between early input level fusion and late decision level fusion, which is demonstrated to be more effective by extensive experimental evaluation in Section III. Another mid-level fusion is naïve concatenation in the feature-level (Figure 2 (c)), which also suffers from feature entanglement problem as the early fusion at input-level (Figure 2 (a)).

III. Experiments

This section presents the evaluation results of our proposed framework and the comparison with the state-of-the-arts.

A. Datasets

Our proposed approach is evaluated on three video description datasets: 1) the Microsoft Video Description corpus (MSVD), 2) the MPII Movie Description dataset (MPII), and 3) the Montreal Video Annotation Dataset (MVAD). While the MSVD dataset is collected from YouTube and there are multiple sentences assigned for each video clip describing a simple event (such as cooking, riding a bike, *etc.*), the MPII and the MVAD datasets are more challenging and complex. The MPII and the MVAD are collected from movies, which contain a variety of scenes and events, and ground-truth sentences are collected via multiple modalities: subtitles, scripts, and descriptive video service (DVS).

1) Microsoft video description corpus: The MSVD corpus is collected from YouTube, while each video clip describes a simple event related to human action and human-object interaction, such as riding a bike, peeling a potato, dancing, *etc.* There are about 2,000 video clips in this corpus, and each video clip is assigned to multiple Mechanical Turk workers to describe using a single sentence. There are 120,000 sentences in this dataset however only a portion of them are in English. This paper only focuses on English descriptions. Note that the dataset is released using YouTube links and a small portion of them become unavailable. In this paper, we obtain 1658 available video clips and 300 of them are used for testing.

2) Max Planck Institut Informatic (MPII) movie description dataset: The MPII dataset [14] is a large movie and text corpus, which is dedicated in aligning High-definition movie snippets to movie scripts and DVS. The MPII dataset contains over $68,000$ sentences and video snippets from 94 HD movies. Each movie snippet is aligned with one sentence from either movie scripts or DVS, which is an additional parallel audio track which can help visually impaired person to understand and follow a movie. The MPII dataset is very challenging due to several aspects: 1) it is extracted from movies, with more complex scenes and varied backgrounds. 2) The text annotations are sourced from a combined corpus (movie scripts + DVS), therefore the linguistic complexity is much higher than well-structured sentences as in the MSVD dataset.

3) Montreal video annotation dataset (MVAD): The MVAD is similar to the MPII dataset, which is also a large movie description corpus. The MVAD dataset text annotations are also sourced from DVS. It includes $49,000$ (summing up to 84.6 hours long) video/sentence pairs from 92 movies. In fact, the MVAD and the MPII belong to the recent Large Scale Movie Description Challenge (LSMDC). We report results on the public test dataset which contains 6518 samples for the MVAD dataset and 3535 samples for the MPII dataset.

4) Experimental setup: **Evaluation metrics.** Our proposed framework is evaluated on all three datasets using the METEOR evaluation metric [4]. METEOR was proposed to quantitatively evaluating the performance of automatic machine translation. Compared to the BLEU metric [11], which is based on the match of n-grams between target sentence and the references, METEOR is more reliable by considering the quality of alignments. In addition, METEOR utilizes more linguistic and semantic information than other metrics. Therefore, METEOR is more suitable to our task. We employ the tool provided by Microsoft COCO Evaluation Server [3] as used in [17].

Training and optimization. For each feature channel, C3D networks pre-trained on Sport1m [6] are employed without fine-tuning for efficiency. The activation values from the last fully connected layer of C3D (fc-7) are extracted as inputs. Our RNN is composed of one layer of 1024 LSTM cells therefore the encoding space is of 1024 dimensions. The feature encoder is composed of two parallel fully connected layers and followed with ReLu nonlinearity, each of which maps the feature from 4096 to 512 dimensions. The encodings of two feature channels are then concatenated as a 1024-dimensional vector for LSTMs. The network is trained end-to-end from feature encoding to sentence generation by maximizing the log-likelihood computed based on Equation (3). ADAM updating is applied for LSTMs and SGD is applied for feature encoders. Learning rates are $1e^{-4}$ and $1e^{-5}$ for LSTMs and feature encoders, respectively. The training process terminates after $200,000$ mini-batches with batch-size 32 and the performance on testing set is reported. Our models are trained on one GTX Titan X GPU and it takes about 1-2 days for training, depending on the datasets.

B. Comparative results

In this section, our proposed model is compared with the state-of-the-art approaches on the three datasets. In addition, to manifest the difference made by our joint encoder, we also compare the propose model with three variants: 1) C3D(RGB): single-stream model based on C3D features on raw RGB frames. 2) C3D(MHI): similar to 1) but based on MHI frames. 3) C3D(RGB+MHI) is the feature-level fusion as illustrated in Figure 2 (c). These three models are provided as baselines. The approaches compared are 1) factor graph model (FGM) proposed in [15], which applies a quadruple template model to combine with action/object detection scores to generate sentences. 2) Mean-pooling model [18] which is fully based on a consensus of image features over all video frames. 3) The temporal attention (TA) model [22] in which the temporal structure is modeled by learning to assign different weights for sampled video frames. 4) Sequence encoding model (S2VT) which is proposed in [17] to apply a sequence-to-sequence modeling LSTM for video captioning. In additional to the four comparisons, we also compare to several variants based on different network implementations (VGG, AlexNet, GoogleNet, *etc*).

1) MSVD dataset: In Table I, the results in METEOR scores of our model and other approaches are listed. The top part shows scores for related approaches and the lower part shows the baselines and our model. Our proposed framework achieves the best METEOR score (31.1%) which is 7.2% higher than FGM [15], 2% higher than Mean-pooling [18], 1.5% higher than temporal attention model [22] and 1.3% higher than the previous state-of-the-art S2VT model [17]. Compared with S2VT and TA, despite the fact that we do not explicitly encode the temporal structure, our method achieves better results due to our joint feature encoding framework captures complementary information.

The benefits of our new model can also be demonstrated by comparing the scores in the lower part. Using RGB (C3D(RGB)) slightly outperforms mean-pooling on AlexNet (27.1% vs. 26.9%) but MHI only achieves significantly inferior score (24.3%) because video description is highly depended on what content inside the video and less depended on the motion, such as detected *human* and *gun* in general generates a description of "a man is shooting a gun." However, combining both feature channels significantly improves the results ($30.3\% - 31.1\%$ *vs.* $24.3\% - 27.1\%$). In addition, the parallel encoder performs better than feature-level fusion, which is also demonstrated on the other two datasets.

2) MPII and MVAD datasets: Compared to the MSVD dataset, the MPII dataset contains similar number of sentences but it is much more challenging that it contains 30 times more video snippets than the MSVD datasets. Besides the number of sentences for each video snippet is much less, the significant variety for both video scenes and language structure also makes this dataset much more challenging. Table II shows the METEOR scores. Our result (7.0%) outperforms the approach

TABLE I
METEOR SCORES ON THE MSVD DATASET.

Method	METEOR (%)
FGM [15]	23.9
AlexNet[18]	26.9
VGG [18]	27.7
AlexNet-COCO [18]	29.1
GoogleNet [22]	28.7
GoogleNet + TA [22]	29.0
GoogleNet + 3D-CNN + TA [22]	29.6
AlexNet(Flow) + S2VT [17]	24.3
AlexNet + S2VT [17]	27.9
VGG + S2VT [17]	29.2
VGG + AlexNet(Flow) + S2VT [17]	29.8
C3D(RGB)	27.1
C3D(MHI)	24.3
C3D(MHI+RGB)	30.3
C3D(MHI+RGB)-Joint (Ours)	**31.1**

proposed in [14] (SMT) by 1.4% and [18] by 0.3% which is very close to the S2VT model and the Visual-Labels proposed along with the dataset [13]. Without explicitly encoding the temporal structure as constructed in [18] for such a challenging dataset, our model still achieves comparable results, which demonstrate the effectiveness of our joint encoding framework.

TABLE II
METEOR SCORES ON THE MPII DATASET.

Method	METEOR (%)
SMT [14]	5.6
Visual-Labels [13]	7.0
VGG [18]	6.7
S2VT [17]	7.1
C3D(RGB)	6.5
C3D(MHI)	6.4
C3D(MHI+RGB)	6.7
C3D(MHI+RGB)-Joint (Ours)	**7.0**

Table III shows the results on the MVAD dataset. Our performance significantly outperforms temporal attention by 2.4% and achieves comparable results with S2VT. The observations are similar to the MPII dataset except that in this dataset, the simple feature concatenation of MHI and RGB does not improve from MHI only, even slightly worse than RGB only, showing that simple concatenation is not a good strategy for multi-stream fusion. However, our proposed joint encoding fusion significantly improves over each of the two channels by 0.5% – 0.6%.

TABLE III
METEOR SCORES ON THE MVAD DATASET.

Method	METEOR (%)
Visual-Labels [13]	6.3
Temporal Attention [22]	4.3
VGG [18]	6.1
S2VT [17]	6.7
C3D(RGB)	6.2
C3D(MHI)	6.1
C3D(MHI+RGB)	6.1
C3D(MHI+RGB)-Joint (Ours)	**6.7**

Qualitative results. Figure 3 shows some sampled descriptions generated by our proposed model on three datasets. The top row shows correct descriptions and the bottom row shows incorrect but plausible results. By observing the "plausible" panel, we can infer some clues about how the model performs the text generation. Firstly, generally speaking, auto captioning algorithm tends to predict simple sentence like "someone is doing something with something", because predicting long and complex sentences is likely to make more mistakes. Another observation is that although temporal structures are encoded in the framework, the object detections are still dominating the text results. For example, in the second plausible result for the MVAD dataset, "someone sits on a bed" is easily be confused with "someone lies on a bed" because there is not enough samples to train the model to learn "sits" from "lies", especially for the complex movie scenes. For the third plausible example in the MSVD dataset, the animal is mis-classified as "cat" because the appearance similarities (like ears, eyes, fur, *etc*). Therefore we believe that further investigating how to model motion features as well as fine-grained object/action detection will produce more accurate text descriptions.

IV. CONCLUSION

In this paper, we have introduced an automatic model that generates natural language descriptions for videos based on a two-stream video representation learning model and a LSTM-based sentence generator. Our approach features a novel parallel video representation model which combines both RGB frames and motion boundary history frames which contain complementary information from visual appearance and temporal motions. 3D convolutional neural networks are employed to further extract spatial and temporal features from both RGB and MHI streams. The proposed framework can effectively learn the simultaneous fusion of multiple streams of features and train the whole model end-to-end. The proposed model is compared with the state-of-the-art video description methods on three different datasets and outperforms them or achieves similar performances.

ACKNOWLEDGMENT

This work was supported in part by NSF grants EFRI-1137172, IIP-1343402, and IIS-1400802.

REFERENCES

[1] K. Barnard, P. Duygulu, D. Forsyth, N. De Freitas, D. M. Blei, and M. I. Jordan. Matching words and pictures. *The Journal of Machine Learning Research*, 3:1107–1135, 2003.

[2] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *Neural Networks, IEEE Transactions on*, 5(2):157–166, 1994.

[3] 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 preprint arXiv:1504.00325*, 2015.

[4] M. Denkowski and A. Lavie. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the EACL 2014 Workshop on Statistical Machine Translation*, volume 6, 2014.

[5] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. *arXiv preprint arXiv:1412.2306*, 2014.

[6] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, pages 1725–1732. IEEE, 2014.

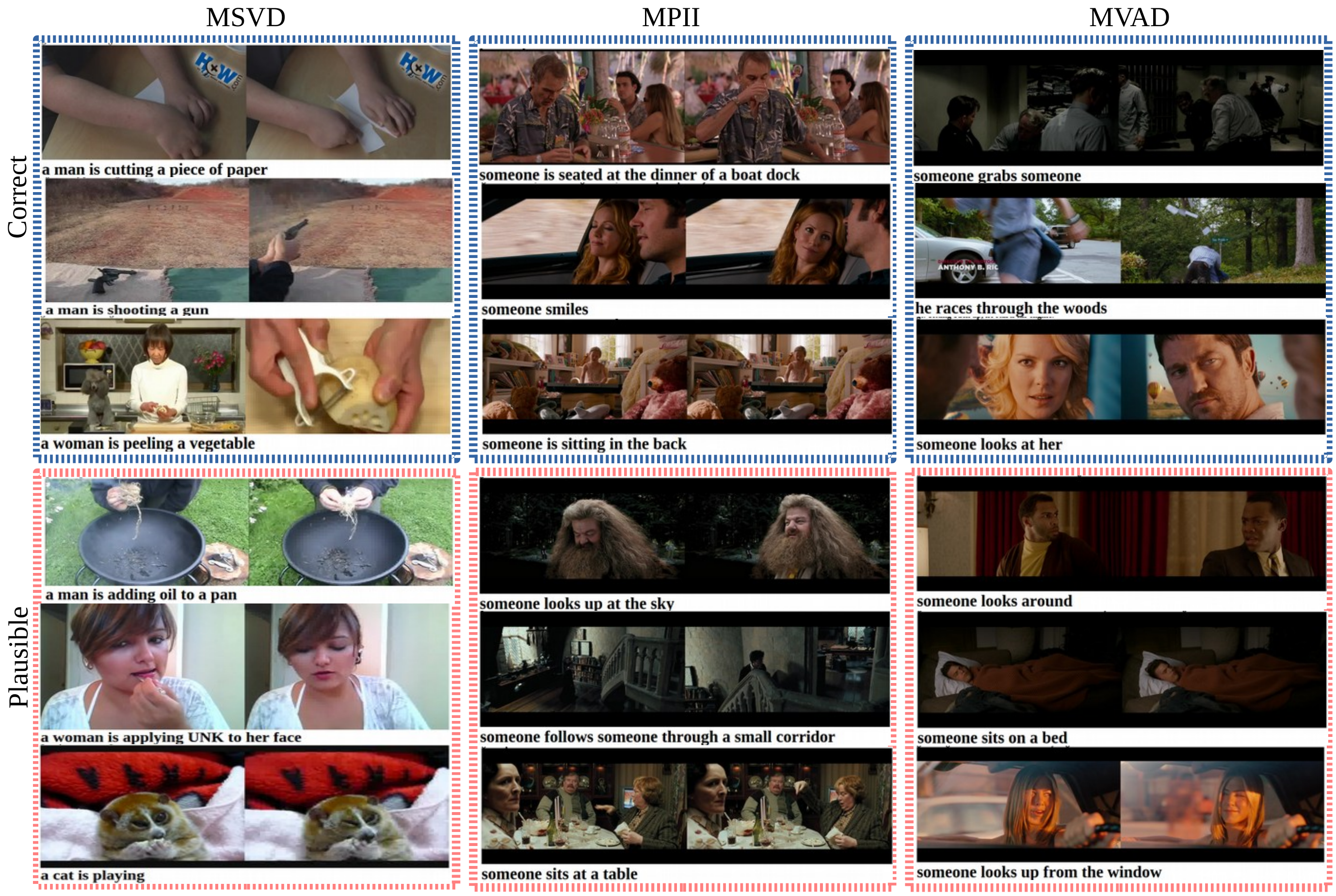
MSVD
MPII
MVAD
Correct
Plausible
a man is cutting a piece of paper
someone is seated at the dinner of a boat dock
someone grabs someone
a man is shooting a gun
someone smiles
he races through the woods
a woman is peeling a vegetable
someone is sitting in the back
someone looks at her
a man is adding oil to a pan
someone looks up at the sky
someone looks around
a woman is applying UNK to her face
someone follows someone through a small corridor
someone sits on a bed
a cat is playing
someone sits at a table
someone looks up from the window

Fig. 3. Examples of generated descriptions of our proposed method on three datasets.

[7] B. Klein, G. Lev, G. Sadeh, and L. Wolf. Fisher vectors derived from hybrid gaussian-laplacian mixture models for image annotation. In *CVPR*, 2015.

[8] C. W. Leong, R. Mihalcea, and S. Hassan. Text mining for automatic image tagging. In *ICCL*, pages 647–655. Association for Computational Linguistics, 2010.

[9] J. Mao, W. Xu, Y. Yang, J. Wang, and A. L. Yuille. Explain images with multimodal recurrent neural networks. *arXiv preprint arXiv:1410.1090*, 2014.

[10] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, pages 3111–3119, 2013.

[11] K. Papineni, S. Roukos, T. Ward, and W.-J. 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. Association for Computational Linguistics, 2002.

[12] K. Pastra, H. Saggion, and Y. Wilks. Nlp for indexing and retrieval of captioned photographs. In *Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics-Volume 2*, pages 143–146. Association for Computational Linguistics, 2003.

[13] A. Rohrbach, M. Rohrbach, and B. Schiele. The long-short story of movie description. In *German Conference on Pattern Recognition (GCPR)*, 2015.

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

[15] 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 *Proceedings of the 25th International Conference on Computational Linguistics (COLING), August*, 2014.

[16] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4489–4497, 2015.

[17] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, and K. Saenko. Sequence to sequence - video to text. In *ICCV*, 2015.

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

[19] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. *arXiv preprint arXiv:1411.4555*, 2014.

[20] K. Xu, J. Ba, R. Kiros, A. Courville, R. Salakhutdinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. *arXiv preprint arXiv:1502.03044*, 2015.

[21] B. Z. Yao, X. Yang, L. Lin, M. W. Lee, and S.-C. Zhu. I2t: Image parsing to text description. *Proceedings of the IEEE*, 98(8):1485–1508, 2010.

[22] L. Yao, A. Torabi, K. Cho, N. Ballas, C. Pal, H. Larochelle, and A. Courville. Describing videos by exploiting temporal structure. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4507–4515, 2015.