

# The Long-Short Story of Movie Description

Anna Rohrbach<sup>1</sup> (✉), Marcus Rohrbach<sup>2</sup>, and Bernt Schiele<sup>1</sup>

<sup>1</sup> Max Planck Institute for Informatics, Saarbrücken, Germany

[arohrbach@mpi-inf.mpg.de](mailto:arohrbach@mpi-inf.mpg.de)

<sup>2</sup> UC Berkeley EECS and ICSI, Berkeley, CA, USA

**Abstract.** Generating descriptions for videos has many applications including assisting blind people and human-robot interaction. The recent advances in image captioning as well as the release of large-scale movie description datasets such as MPII-MD [28] and M-VAD [31] allow to study this task in more depth. Many of the proposed methods for image captioning rely on pre-trained object classifier CNNs and Long Short-Term Memory recurrent networks (LSTMs) for generating descriptions. While image description focuses on objects, we argue that it is important to distinguish verbs, objects, and places in the setting of movie description. In this work we show how to learn robust visual classifiers from the weak annotations of the sentence descriptions. Based on these classifiers we generate a description using an LSTM. We explore different design choices to build and train the LSTM and achieve the best performance to date on the challenging MPII-MD and M-VAD datasets. We compare and analyze our approach and prior work along various dimensions to better understand the key challenges of the movie description task.

## 1 Introduction

Automatic description of visual content has lately received a lot of interest in our community. Multiple works have successfully addressed the image captioning problem [6, 16, 17, 35]. Many of the proposed methods rely on Long Short-Term Memory networks (LSTMs) [13]. In the meanwhile, two large-scale movie description datasets have been proposed, namely MPII Movie Description (MPII-MD) [28] and Montreal Video Annotation Dataset (M-VAD) [31]. Both are based on movies with associated textual descriptions and allow studying the problem how to generate movie description for visually disabled people. Works addressing these datasets [28, 33, 38] show that they are indeed challenging in terms of visual recognition and automatic description. This results in a significantly lower performance than on simpler video datasets (e.g. MSVD [2]), but a detailed analysis of the difficulties is missing. In this work we address this by taking a closer look at the performance of existing methods on the movie description task.

This work contributes (a) an approach to build robust visual classifiers which distinguish verbs, objects, and places extracted from weak sentence annotations; (b) based on the visual classifiers we evaluate different design choices to train an LSTM for generating descriptions. This outperforms related work on the

MPII-MD and M-VAD datasets, using automatic and human evaluation (only on MPII-MD); (c) we perform a detailed analysis of prior work and our approach to understand the challenges of the movie description task.

## 2 Related Work

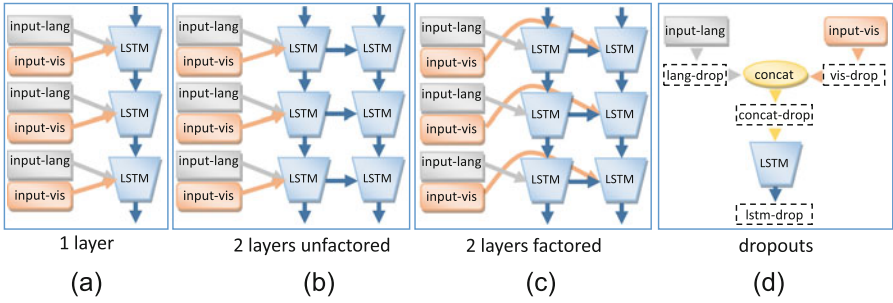
**Image captioning.** Automatic image description has been studied in the past [9, 19, 20, 23], gaining increased attention just recently [6, 8, 16, 17, 22, 35]. Many of the proposed works rely on Recurrent Neural Networks (RNNs) and in particular on Long Short-Term Memory networks (LSTMs). New datasets have been released, Flickr30k [39] and MS COCO Captions [3], where [3] also presents a standardized protocol for image captioning evaluation. There are attempts to analyze the performance of recent methods, e.g. [5] compares them with respect to the novelty of generated descriptions and additionally proposes a nearest neighbor baseline that improves over recent methods.

**Video description.** In the past video description has been addressed in controlled settings [1, 18], on a small scale [4, 11, 30] or in single domains like cooking [26, 29]. Donahue *et al.* [6] first proposed to describe videos using an LSTM, relying on precomputed CRF scores from [26]. Later [34] extended this work to extract CNN features from frames which are max-pooled over time. Pan *et al.* [24] propose a framework with a visual-semantic embedding to ensure better coherence between video and text. Xu *et al.* [37] jointly address the language generation and video/language retrieval tasks by learning a joint embedding for a deep video model and compositional semantic language model.

**Movie description.** Recently two large-scale movie description datasets have been proposed, MPII Movie Description [28] and Montreal Video Annotation Dataset [31]. Compared to previous video description datasets, they have broader domain and are more varied and challenging with respect to the visual content and the associated descriptions. They also do not have any additional annotations, as e.g. TACoS Multi-Level [26], thus one has to rely on the weak sentence annotations. To handle this challenging scenario [38] proposes an attention based model which selects the most relevant temporal segments in a video, incorporates 3-D CNN and generates a sentence using an LSTM. Venugopalan *et al.* [33] propose an encoder-decoder framework, where a single LSTM encodes the input video frame by frame and decodes it into a sentence, outperforming [38]. Our approach for sentence generation is most similar to [6] and we rely on their LSTM implementation based on Caffe [15].

## 3 Approach

In this section we present our two-step approach. The first step performs visual recognition using the visual classifiers which we train according to labels’ semantics and “visuality”. The second step generates textual descriptions using an LSTM. We explore various design choices for building and training the LSTM.



**Fig. 1.** (a–c) LSTM architectures. (d) Variants of placing the dropout layer.

### 3.1 Visual Labels for Robust Visual Classifiers

For training we rely on a parallel corpus of videos and weak sentence annotations. As in [28] we parse the sentences to obtain a set of labels (single words or short phrases, e.g. *look up*) to train visual classifiers. However, in contrast to [28], we do not want to keep all of these initial labels as they are noisy, but select only visual ones which actually can be robustly recognized.

**Avoiding parser failure.** Not all sentences can be parsed successfully, as e.g. some sentences are incomplete or grammatically incorrect. To avoid losing the potential labels in these sentences, we match our set of initial labels to the sentences which the parser failed to process.

**Semantic groups.** Our labels correspond to different semantic groups. In this work we consider three most important groups: verbs, objects and places. We propose to treat each label group independently. First, we rely on a different representation for each semantic group, which is targeted to the specific group. Namely we use the activity recognition features Improved Dense Trajectories (DT) [36] for verbs, large scale object detector responses (LSDA) [14] for objects and scene classification scores (PLACES) [40] for places. Second, we train one-vs-all SVM classifiers for each group separately. The intuition behind this is to avoid “wrong negatives” (e.g. using *object* “bed” as negative for *place* “bedroom”).

**Visual labels.** Now, how do we select *visual* labels for our semantic groups? In order to find the verbs among the labels we rely on the semantic parser of [28]. Next, we look up the list of “places” used in [40] and search for corresponding words among our labels. We look up the object classes used in [14] and search for these “objects”, as well as their base forms (e.g. “domestic cat” and “cat”). We discard all the labels that do not belong to any of our three groups of interest as we assume that they are likely not visual and thus are difficult to recognize. Finally, we discard labels which the classifiers could not learn, as these are likely noisy or not visual. For this we require the classifiers to have certain minimum area under the ROC-curve (Receiver Operating Characteristic).

### 3.2 LSTM for Sentence Generation

We rely on the basic LSTM architecture proposed in [6] for video description. At each time step an LSTM generates a word and receives the visual classifiers (*input-vis*) as well as the previous generated word (*input-lang*) as input (see Fig. 1(a)). We encode each word with a one-hot-vector according to its index in a dictionary and project it in a lower dimensional embedding. The embedding is jointly learned during training of the LSTM. We feed in the classifier scores as input to the LSTM which is equivalent to the best variant proposed in [6]. We analyze the following aspects for this architecture:

**Layer structure.** We compare a 1-layer architecture with a 2-layer architecture. In the 2-layer architecture, the output of the first layer is used as input for the second layer (Fig. 1b) and was used by [6] for video description. Additionally we also compare to a 2-layer factored architecture of [6], where the first layer only gets the language as input and the second layer gets the output of the first as well as the visual input.

**Dropout placement.** To learn a more robust network which is less likely to overfit we rely on a dropout [12], i.e. a ratio  $r$  of randomly selected units is set to 0 during training (while all others are multiplied with  $1/r$ ). We explore different ways to place dropout in the network, i.e. either for language input (*lang-drop*) or visual (*vis-drop*) input only, for both inputs (*concat-drop*) or for the LSTM output (*lstm-drop*), see Fig. 1(d). While the default dropout ratio is  $r = 0.5$ , we evaluate the effect of other ratios.

**Learning strategy.** By default we use a step-based learning strategy, where a learning rate is halved after a certain number of steps. We find the best learning rate and step size on the validation set. Additionally we compare this to a polynomial learning strategy, where the learning rate is continuously decreased. This learning strategy has been shown to give good results faster without tweaking the step size for GoogleNet implemented by Sergio Guadarrama in Caffe [15].

## 4 Evaluation

In this section we first analyze our approach on the MPII-MD [28] dataset and explore different design choices. Then, we compare our best system to prior work.

### 4.1 Analysis of Our Approach

**Experimental setup.** We build on the labels discovered by the semantic parser of [28]. To be able to learn classifiers we select the labels that appear at least 30 times, resulting in 1,263 labels. The parser additionally tells us whether the label is a verb. We use the visual features (DT, LSDA, PLACES) provided with the MPII-MD dataset [28]. The LSTM output/hidden unit as well as memory cell have each 500 dimensions. We train our method on the training set (56,861 clips) and evaluate on the validation set (4,930 clips) using the METEOR [21] score.

According to [7, 32], METEOR supersedes previously used measures such as BLEU [25] in terms of agreement with human judgments. METEOR also outperforms CIDEr [32] when the number of references is small and in the case of MPII-MD we have only a single reference.

**Table 1.** Comparison of different choices of labels and visual classifiers. All results reported on the validation set of MPII-MD.

Approach	Labels	Classifiers	
		Retrieved	Trained
<b>Baseline: all labels treated the same way</b>			
(1) DT	1263	-	6.73
(2) LSDA	1263	-	7.07
(3) PLACES	1263	-	7.10
(4) DT+LSDA+PLACES	1263	-	7.24
<b>Visual labels</b>			
(5) Verbs(DT), Others(LSDA)	1328	7.08	7.27
(6) Verbs(DT), Places(PLACES), Others(LSDA)	1328	7.09	7.39
(7) Verbs(DT), Places(PLACES), Objects(LSDA)	913	7.10	7.48
(8) + restriction to labels with $ROC \geq 0.7$	263	7.41	<b>7.54</b>
<b>Baseline: all labels treated the same way, labels from (8)</b>			
(9) DT+LSDA+PLACES	263	7.16	7.20

**Robust visual classifiers.** In a first set of experiments we analyze our proposal to consider groups of labels to learn different classifiers and also to use different visual representations for these groups (see Sect. 3.1). In Table 1 we evaluate our generated sentences using different input features to the LSTM. In our baseline, in the top part of Table 1, we treat all labels equally, i.e. we use the same visual descriptors for all labels. The PLACES feature is best with 7.10 METEOR. Combination by stacking all features (DT + LSDA + PLACES) improves further to 7.24 METEOR.

The second part of the table demonstrates the effect of introducing different semantic label groups. We first split the labels into “Verbs” and all remaining. Given that some labels appear in both roles, the total number of labels increases to 1328 (line 5). We analyze two settings of training the classifiers. In the case of “Retrieved” we retrieve the classifier scores from the general classifiers trained in the previous step. “Trained” corresponds to training the SVMs specifically for each label type (e.g. for “Verbs”). Next, we further divide the non-“Verb” labels into “Places” and “Others” (line 6), and finally into “Places” and “Objects” (line 7). We discard the unused labels and end up with 913 labels. Out of these labels, we select the labels where the classifier obtains a ROC higher or equal to 0.7 (threshold selected on the validation set). After this we obtain 263 labels and the best performance in the “Trained” setting (line 8). To support our intuition about the importance of the label discrimination (i.e. using different features for different semantic groups of labels), we propose another baseline (line 9). Here we use the same set of 263 labels but provide the same feature for all of them, namely the best performing combination DT + LSDA + PLACES. As we see, this results in an inferior performance.

**Table 2.** LSTM architectures, MPII-MD val set. Labels, classifiers as Table 1(8).

Architecture	<i>METEOR</i>	Dropout	<i>METEOR</i>	Dropout ratio <i>METEOR</i>	
		no dropout	7.19	r=0.1	7.22
1 layer	<b>7.54</b>	lang-drop	7.13	r=0.25	7.42
2 layers unfact.	<b>7.54</b>	vis-drop	7.34	r=0.5	<b>7.54</b>
2 layers fact.	7.41	concat-drop	7.29	r=0.75	7.46
		lstm-drop	<b>7.54</b>		

(a) LSTM architectures (lstm-dropout 0.5)      (b) Dropout strategies (1-layer, dropout 0.5)      (c) Dropout ratios (1-layer, lstm-dropout).

We make several observations from Table 1 which lead to robust visual classifiers from the weak sentence annotations. (a) It is beneficial to select features based on the label semantics. (b) Training one-vs-all SVMs for specific label groups consistently improves the performance as it avoids “wrong” negatives. (c) Focusing on more “visual” labels helps: we reduce the LSTM input dimensionality to 263 while improving the performance.

**LSTM architectures.** Now, as described in Sect. 3.2, we look at different LSTM architectures and training configurations. In the following we use the best performing “Visual Labels” approach, Table 1, line (8).

We start with examining the architecture, where we explore different configurations of LSTM and dropout layers. Table 2a shows the performance of three different networks: “1 layer”, “2 layers unfactored” and “2 layers factored” introduced in Sect. 3.2. As we see, the “1 layer” and “2 layers unfactored” perform equally well, while “2 layers factored” is inferior to them. In the following experiments we use the simpler “1 layer” network. We then compare different dropout placements as illustrated in (Fig. 2b). We obtain the best result when applying dropout after the LSTM layer (“lstm-drop”), while having no dropout or applying it only to language leads to stronger over-fitting to the visual features. Putting dropout after the LSTM (and prior to a final prediction layer) makes the entire system more robust. As for the best dropout ratio, we find that 0.5 works best with lstm-dropout (Table 2c).

Next we look at different learning rates and strategies<sup>1</sup>. We find that the best learning rate in the step-based learning is 0.01, while step size 4000 slightly improves over step size 2000 (which we used in Table 1). We explore an alternative learning strategy, namely decreasing learning rate according to a polynomial decay. We experiment with different exponents (0.5 and 0.7) and numbers of iterations (25 K and 10 K), using the base-learning rate 0.01. Our results show that the step-based learning is superior to the polynomial learning.

In most of the experiments we trained our networks for 25,000 iterations. After looking at the METEOR performance for intermediate iterations we found that for the step size 4000 at iteration 15,000 we achieve best performance overall. Additionally we train multiple LSTMs with different random orderings of the training data. In our experiments we combine three in an ensemble, averaging

<sup>1</sup> More details can be found in our corresponding arXiv version [27].

**Table 3.** Comparison to prior work. Human eval ranked 1 to 3, lower is better.

Approach	<i>METEOR</i> in %	Human evaluation: rank			Approach	<i>METEOR</i> in %
		Correct.	Grammar	Relev.		
Best of [28]	5.59	2.11	2.39	2.08	[38]	4.33
S2VT [33]	6.27	2.02	<b>1.67</b>	2.06	S2VT [33]	5.62
Visual-Labels (our)	<b>7.03</b>	<b>1.87</b>	1.94	<b>1.86</b>	Visual-Labels (our)	<b>6.36</b>
NN-upperbound	19.43	-	-	-		

(a) Test Set of MPII-MD.

(b) Test Set of M-VAD.

the resulting word predictions. In most cases the ensemble improves over the single networks in terms of METEOR score (see Footnote 1).




To summarize, the most important aspects that decrease over-fitting and lead to a better sentence generation are: (a) a correct learning rate and step size, (b) dropout after the LSTM layer, (c) choosing the training iteration based on METEOR score as opposed to only looking at the LSTM accuracy/loss which can be misleading, and (d) building ensembles of multiple networks with different random initializations. In the following section we compare our best ensemble (selected on the validation set) to related work on the test sets of MPII-MD and M-VAD.

## 4.2 Comparison to Related Work

**Experimental setup.** First we compare the best method of [28], the recently proposed method S2VT [33] and our proposed “Visual Labels”-LSTM on the test set of the MPII-MD dataset (6,578 clips). In addition to METEOR [21], we perform a human evaluation, by randomly selecting 1300 video snippets and asking AMT turkers to rank three systems with respect to correctness, grammar and relevance, similar to [28]. Next we evaluate our method on the test set of the M-VAD dataset [31] (4,951 clips) and compare it to [33] and [38]. We train our method on M-VAD and use the same LSTM architecture and parameters as for MPII-MD, but select the number of iterations on the M-VAD validation set.

**Results on MPII-MD.** Table 3a summarizes the results on the test set of MPII-MD (see Footnote 1). While we rely on identical features and similar labels as [28], we significantly improve the performance, specifically by 1.44 METEOR points. Moreover, we improve over the recent approach of [33], which also uses LSTM to generate video descriptions. Exploring different strategies to label selection and classifier training, as well as various LSTM configurations allows to obtain best result to date on the MPII-MD dataset. Human evaluation mainly agrees with the automatic measure. We outperform both prior works in terms of Correctness and Relevance, however we lose to S2VT in terms of Grammar. This is due to the fact that S2VT produces overall shorter (7.4 versus 8.7 words per sentence) and simpler sentences, while our system generates longer sentences and therefore has higher chances to make mistakes. We also propose a retrieval upperbound. For every test sentence we retrieve the closest training sentence according to the METEOR score. The rather low METEOR score of 19.43 reflects the difficulty of the dataset.



	Approach	Sentence
	SMT [28]	Someone is a man, someone is a man.
	S2VT [33]	Someone looks at him, someone turns to someone.
	Our	Someone is standing in the crowd, a little man with a little smile.
	Reference	Someone, back in elf guise, is trying to calm the kids.
	SMT [28]	The car is a water of the water.
	S2VT [33]	On the door, opens the door opens.
	Our	The fellowship are in the courtyard.
	Reference	They cross the quadrangle below and run along the cloister.
	SMT [28]	Someone is down the door, someone is a back of the door, and someone is a door.
	S2VT [33]	Someone shakes his head and looks at someone.
	Our	Someone takes a drink and pours it into the water.
	Reference	Someone grabs a vodka bottle standing open on the counter and liberally pours some on the hand.

**Fig. 2.** Qualitative comparison of prior work and our proposed method. Examples from the test set of MPII-MD. Our approach identifies activities, objects, and places better than related work.

An interesting characteristic of the compared methods is the size of the output vocabulary, which is 94 for [28], 86 for [33] and 605 for our method, while it is 6,422 for the reference test sentences. This clearly shows a higher diversity of our output. Unlike other methods ours can generate e.g. verbs as *grab*, *drive*, *sip*, *climb*, *follow*, objects as *suit*, *chair*, *cigarette*, *mirror*, *bottle* and places as *kitchen*, *corridor*, *restaurant*. We show some qualitative results in Fig. 2. Here, the verb *pour*, object *drink* and place *courtyard* only appear in our output. We attribute this, on one hand, to our diverse and robust visual classifiers. On the other hand, the architecture and parameter choices of our LSTM allow us to learn better correspondence between the words and the visual classifiers’ scores.

**Results on M-VAD.** Table 3b shows the results on the test set of M-VAD dataset. Our method outperforms the other two in METEOR score. As we see, the results overall agree with Table 3a, but are consistently lower suggesting that M-VAD is more challenging than MPII-MD. We attribute this to more precise manual alignments of the MPII-MD dataset.

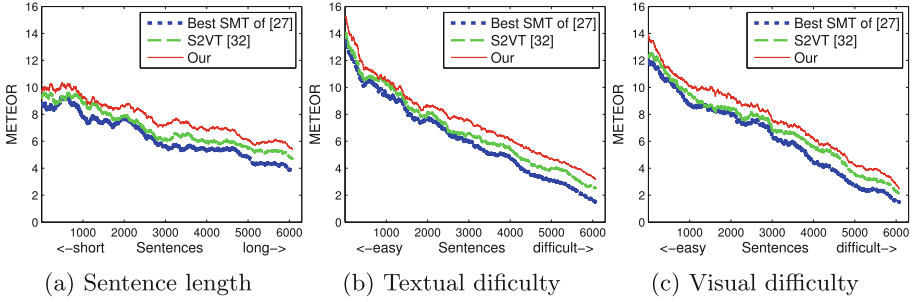
## 5 Analysis

Despite the recent advances in the video description task, the performance on the movie description datasets (MPII-MD and M-VAD) remains rather low. In this section we want to look closer at three methods, SMT of [28], S2VT [33] and ours, in order to understand where these methods succeed and where they fail. In the following we evaluate all three methods on the MPII-MD test set.

### 5.1 Difficulty Versus Performance

As the first study we suggest to sort the test reference sentences by difficulty, where difficulty is defined in multiple ways.





**Fig. 3.** Y-axis: METEOR score per sentence. X-axis: test sentences 1 to 6,578 sorted by (a) length (increasing); (b) textual difficulty (increasing); (c) visual difficulty (increasing). Shown values are smoothed with a mean filter of size 500.

**Sentence length and Word frequency.** Some of the intuitive sentence difficulty measures are its length and average frequency of its words. When sorting the data by difficulty (increasing sentence length or decreasing average word frequency), we find that all three methods have the same tendency to obtain lower METEOR score as the difficulty increases. Figure 3a shows the performance of compared methods w.r.t. the sentence length. For the word frequency the correlation is even stronger (see Footnote 1). Our method consistently outperforms the other two, most notable as the difficulty increases.

**Textual and Visual difficulty.** Next, for each test reference sentence we search for the closest training sentence (in terms of the METEOR score). We use the obtained best scores to sort the reference sentences by *textual difficulty*, i.e. the “easy” sentences are more likely to be retrieved. If we consider all training sentences, we obtain a textual Nearest Neighbor. We plot the performance of three methods w.r.t. the textual difficulty in Fig. 3b. All methods “agree” and ours is best throughout the difficulty range, in particular in the most challenging part of the plot (right). We can also use visual features to find the  $k$  visual Nearest Neighbors in the Training set, select the best one (in terms of the METEOR score) and use this score to sort the reference sentences. We call this a *visual difficulty*. The intuition behind it is to consider a video clip as visually “easy” if the most similar training clips also have similar descriptions (the “difficult” clip might have no close visual neighbours). We rely on our best visual representation (8) from Table 1 and *cos* similarity measure to define the visual difficulty and sort the reference sentences according to it, using  $k = 10$  (Fig. 3c). Again, we see a clear correlation between the visual difficulty and the performance of all methods (Fig. 3c).

**Summary.** (a) All methods perform better on shorter, common sentences and our method notably wins on longer sentences. (b) Our method also wins on sentences that are more difficult to retrieve. (c) Visual difficulty, defined by *cos* similarity and representation (8) from Table 1, strongly correlates with the performance of all methods.

## 5.2 Semantic Analysis

**WordNet Verb Topics.** Next we analyze the test reference sentences w.r.t. verb semantics. We rely on WordNet Topics (high level entries in the WordNet ontology), e.g. “motion”, “perception”, defined for most synsets in WordNet [10]. Sense information comes from the semantic parser of [28], thus it might be noisy. We select sentences with a single verb, group them according to the verb’s Topic and compute an average METEOR score for each group (see Footnote 1). We find that our method is best for all Topics except “communication”, where [28] wins. The most frequent verbs there are “look up” and “nod”, which are also frequent in the dataset and in the sentences produced by [28]. The best performing Topic, “cognition”, is highly biased to “look at” verb. The most frequent Topics, “motion” and “contact”, which are also visual (e.g. “turn”, “walk”, “open”), are nevertheless quite challenging, which we attribute to their high diversity. Topics with more abstract verbs (e.g. “be”, “have”, “start”) get lower scores.

**Top 100 best and worst sentences.** We look at 100 test reference sentences, where our method obtains highest and lowest METEOR scores. Out of 100 best sentences 44 contain the verb “look” (including phrases such as “look at”). The other frequent verbs are “walk”, “turn”, “smile”, “nod”, “shake”, i.e. mainly visual verbs. Overall the sentences are simple. Among the worst 100 sentences we observe more diversity: 12 contain no verb, 10 mention unusual words (specific to the movie), 24 have no subject, 29 have a non-human subject. This leads to a lower performance, in particular, as most training sentences contain “Someone” as subject and generated sentences are biased towards it.

**Summary.** (a) The test reference sentences that mention verbs like “look” get higher scores due to their high frequency in the dataset. (b) The sentences with more “visual” verbs tend to get higher scores. (c) The sentences without verbs (e.g. describing a scene), without subjects or with non-human subjects get lower scores, which can be explained by dataset biases.

## 6 Conclusion

We propose an approach to automatic movie description which trains visual classifiers and uses their scores as input to LSTM. To handle the weak sentence annotations we rely on three ingredients. (1) We distinguish three semantic groups of labels (verbs, objects and places). (2) We train them separately, removing the noisy negatives. (3) We select only the most reliable classifiers. For sentence generation we show the benefits of exploring different LSTM architectures and learning configurations. As the result we obtain the highest performance on the MPII-MD and M-VAD datasets as shown by automatic and human evaluation.

We analyze the challenges in the movie description task using our and two prior works. We find that the factors which contribute to higher performance include: presence of frequent words, sentence length and simplicity as well as presence of “visual” verbs (e.g. “nod”, “walk”, “sit”, “smile”). We observe a high bias in the data towards humans as subjects and verbs similar to “look”.

Future work has to focus on dealing with less frequent words and handle less visual descriptions. This potentially requires to consider external text corpora, modalities other than video, such as audio and dialog, and to look across multiple sentences. This would allow exploiting long- and short-range context and thus understanding and describing the story of the movie.

**Acknowledgements.** Marcus Rohrbach was supported by a fellowship within the FIT weltweit-Program of the German Academic Exchange Service (DAAD). The authors thank Niket Tandon for help with the WordNet Topics analysis.

## References

1. Barbu, A., Bridge, A., Burchill, Z., Coroian, D., Dickinson, S., Fidler, S., Michaux, A., Mussman, S., Narayanaswamy, S., Salvi, D., Schmidt, L., Shangquan, J., Siskind, J.M., Waggoner, J., Wang, S., Wei, J., Yin, Y., Zhang, Z.: Video in sentences out. In: UAI (2012)
2. Chen, D., Dolan, W.: Collecting highly parallel data for paraphrase evaluation. In: ACL (2011)
3. Chen, X., Fang, H., Lin, T., Vedantam, R., Gupta, S., Dollr, P., Zitnick, C.L.: Microsoft coco captions: data collection and evaluation server (2015). [arXiv:1504.00325](https://arxiv.org/abs/1504.00325)
4. Das, P., Xu, C., Doell, R., Corso, J.: Thousand frames in just a few words: lingual description of videos through latent topics and sparse object stitching. In: CVPR (2013)
5. Devlin, J., Cheng, H., Fang, H., Gupta, S., Deng, L., He, X., Zweig, G., Mitchell, M.: Language models for image captioning: the quirks and what works (2015). [arXiv:1505.01809](https://arxiv.org/abs/1505.01809)
6. Donahue, J., Hendricks, L.A., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., Darrell, T.: Long-term recurrent convolutional networks for visual recognition and description. In: CVPR (2015)
7. Elliott, D., Keller, F.: Image description using visual dependency representations. In: EMNLP, pp. 1292–1302 (2013)
8. Fang, H., Gupta, S., Iandola, F.N., Srivastava, R., Deng, L., Dollár, P., Gao, J., He, X., Mitchell, M., Platt, J.C., Zitnick, C.L., Zweig, G.: From captions to visual concepts and back. In: CVPR (2015)
9. Farhadi, A., Hejrati, M., Sadeghi, M.A., Young, P., Rashtchian, C., Hockenmaier, J., Forsyth, D.: Every picture tells a story: generating sentences from images. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part IV. LNCS, vol. 6314, pp. 15–29. Springer, Heidelberg (2010)
10. Fellbaum, C.: WordNet: An Electronic Lexical Database. The MIT Press, Cambridge (1998)
11. Guadarrama, S., Krishnamoorthy, N., Malkarnenkar, G., Venugopalan, S., Mooney, R., Darrell, T., Saenko, K.: Youtube2text: Recognizing and describing arbitrary activities using semantic hierarchies and zero-shoot recognition. In: ICCV (2013)
12. Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.R.: Improving neural networks by preventing co-adaptation of feature detectors (2012). [arXiv:1207.0580](https://arxiv.org/abs/1207.0580)
13. Hochreiter, S., Schmidhuber, J.: Long short-term memory. *Neural Comput.* **9**(8), 1735–1780 (1997)

14. Hoffman, J., Guadarrama, S., Tzeng, E., Donahue, J., Girshick, R., Darrell, T., Saenko, K.: LSDA: large scale detection through adaptation. In: NIPS (2014)
15. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: Convolutional architecture for fast feature embedding (2014). [arXiv:1408.5093](https://arxiv.org/abs/1408.5093)
16. Karpathy, A., Fei-Fei, L.: Deep visual-semantic alignments for generating image descriptions. In: CVPR (2015)
17. Kiros, R., Salakhutdinov, R., Zemel, R.S.: Unifying visual-semantic embeddings with multimodal neural language models. TACL (2015)
18. Kojima, A., Tamura, T., Fukunaga, K.: Natural language description of human activities from video images based on concept hierarchy of actions. IJCV **50**(2), 171–184 (2002)
19. Kulkarni, G., Premraj, V., Dhar, S., Li, S., Choi, Y., Berg, A.C., Berg, T.L.: Baby talk: understanding and generating simple image descriptions. In: CVPR (2011)
20. Kuznetsova, P., Ordonez, V., Berg, T.L., Hill, U.C., Choi, Y.: Treetalk: composition and compression of trees for image descriptions. In: TACL (2014)
21. Lavie, M.D.A.: Meteor universal: language specific translation evaluation for any target language. In: ACL 2014, p. 376 (2014)
22. Mao, J., Xu, W., Yang, Y., Wang, J., Huang, Z., Yuille, A.: Deep captioning with multimodal recurrent neural networks (m-RNN). In: ICLR (2015)
23. Mitchell, M., Dodge, J., Goyal, A., Yamaguchi, K., Stratos, K., Han, X., Mensch, A., Berg, A.C., Berg, T.L., Daume III, H.: Midge: generating image descriptions from computer vision detections. In: EACL (2012)
24. Pan, Y., Mei, T., Yao, T., Li, H., Rui, Y.: Jointly modeling embedding and translation to bridge video and language (2015). [arXiv:1505.01861](https://arxiv.org/abs/1505.01861)
25. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: BLEU: a method for automatic evaluation of machine translation. In: ACL (2002)
26. Rohrbach, A., Rohrbach, M., Qiu, W., Friedrich, A., Pinkal, M., Schiele, B.: Coherent multi-sentence video description with variable level of detail. In: Jiang, X., Hornegger, J., Koch, R. (eds.) GCPR 2014. LNCS, vol. 8753, pp. 184–195. Springer, Heidelberg (2014)
27. Rohrbach, A., Rohrbach, M., Schiele, B.: The long-short story of movie description (2015). [arXiv:1506.01698](https://arxiv.org/abs/1506.01698)
28. Rohrbach, A., Rohrbach, M., Tandon, N., Schiele, B.: A dataset for movie description. In: CVPR (2015)
29. Rohrbach, M., Qiu, W., Titov, I., Thater, S., Pinkal, M., Schiele, B.: Translating video content to natural language descriptions. In: ICCV (2013)
30. Thomason, J., Venugopalan, S., Guadarrama, S., Saenko, K., Mooney, R.J.: Integrating language and vision to generate natural language descriptions of videos in the wild. In: COLING (2014)
31. Torabi, A., Pal, C., Larochelle, H., Courville, A.: Using descriptive video services to create a large data source for video annotation research (2015). [arXiv:1503.01070v1](https://arxiv.org/abs/1503.01070v1)
32. Vedantam, R., Zitnick, C.L., Parikh, D.: Cider: Consensus-based image description evaluation. In: CVPR (2015)
33. Venugopalan, S., Rohrbach, M., Donahue, J., Mooney, R., Darrell, T., Saenko, K.: Sequence to sequence - video to text (2015). [arXiv:1505.00487](https://arxiv.org/abs/1505.00487)
34. Venugopalan, S., Xu, H., Donahue, J., Rohrbach, M., Mooney, R., Saenko, K.: Translating videos to natural language using deep recurrent neural networks. In: NAACL (2015)
35. Vinyals, O., Toshev, A., Bengio, S., Erhan, D.: Show and tell: A neural image caption generator. In: CVPR (2015)

36. Wang, H., Schmid, C.: Action recognition with improved trajectories. In: ICCV (2013)
37. Xu, R., Xiong, C., Chen, W., Corso, J.J.: Jointly modeling deep video and compositional text to bridge vision and language in a unified framework. In: AAAI (2015)
38. Yao, L., Torabi, A., Cho, K., Ballas, N., Pal, C., Larochelle, H., Courville, A.: Describing videos by exploiting temporal structure (2015). [arXiv:1502.08029v4](https://arxiv.org/abs/1502.08029v4)
39. Young, P., Lai, A., Hodosh, M., Hockenmaier, J.: From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *TACL* **2**, 67–78 (2014)
40. Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., Oliva, A.: Learning Deep Features for Scene Recognition using Places Database. In: NIPS (2014)