Translating Videos to Natural Language Using Deep Recurrent Neural Networks

Subhashini Venugopalan
UT Austin
Austin, TX
vsub@cs.utexas.edu

Huijuan Xu
UMass Lowell
Lowell, MA
Huijuan_Xu@student.uml.edu

Jeff Donahue
UC Berkeley, ICSI
Berkeley, CA
jdonahue@eecs.berkeley.edu

Marcus Rohrbach
UC Berkeley, ICSI
Berkeley, CA
rohrbach@eecs.berkeley.edu

Raymond Mooney
UT Austin
Austin, TX
mooney@cs.utexas.edu

Kate Saenko
UMass Lowell
Lowell, MA
saenko@cs.uml.edu

Abstract

Solving the visual symbol grounding problem has long been a goal of artificial intelligence. The field appears to be advancing closer to this goal with recent breakthroughs in deep learning for natural language grounding in static images. In this paper, we propose to translate videos directly to sentences using a unified deep neural network with both convolutional and recurrent structure. Described video datasets are scarce, and most existing methods have been applied to toy domains with a small vocabulary of possible words. By transferring knowledge from 1.2M+ images with category labels and 100,000+ images with captions, our method is able to create sentence descriptions of open-domain videos with large vocabularies. We compare our approach with recent work using language generation metrics, subject, verb, and object prediction accuracy, and a human evaluation.

1 Introduction

For most people, watching a brief video and describing what happened in words is an easy task. For machines, extracting the meaning from video pixels and generating natural-sounding language is a very complex problem. Solutions have been proposed for narrow domains with a small set of known actions and objects, e.g., (Barbu et al., 2012; Rohrbach et al., 2013), but generating descriptions for "in-the-wild" videos such as the YouTube domain (Figure 1) remains an open challenge.

Progress in open-domain video description has been difficult in part due to large vocabularies and very limited training data consisting of videos with associated descriptive sentences. Another serious obstacle has been the lack of rich models that can capture the joint dependencies of a sequence of frames and a corresponding sequence of words. Previous work has simplified the problem by detecting a fixed set of semantic roles, such as subject, verb, and object (Guadarrama et al., 2013; Thomason et al., 2014), as an intermediate representation. This fixed representation is problematic for large vocabularies and also leads to oversimplified rigid sentence templates which are unable to model the complex structures of natural language.

Input video:

Our output: *A cat is playing with a toy.*
Humans: *A Ferret and cat fighting with each other. / A cat and a ferret are playing. / A kitten is playing with a ferret. / A kitten and a ferret are playfully wrestling.*

Figure 1: Our system takes a short video as input and outputs a natural language description of the main activity in the video.

In this paper, we propose to translate from video pixels to natural language with a single deep neural network. Deep NNs can learn powerful features (Donahue et al., 2013; Zeiler and Fergus, 2014), but require a lot of supervised training data. We address the problem by transferring knowledge from auxiliary tasks. Each frame of the video is modeled by a convolutional (spatially-invariant) network pre-trained on 1.2M+ images with category labels (Krizhevsky et al., 2012). The meaning state and sequence of words is modeled by a recurrent (temporally invariant) deep network pre-trained on 100K+ Flickr (Peter Young and Hockenmaier, 2014) and COCO (Lin et al., 2014) images with associated sentence captions. We show that such knowledge transfer significantly improves performance on the video task.

Our approach is inspired by recent breakthroughs reported by several research groups in image-to-text generation, profiled in the popular media,[1] [2] in particular, the work by (Donahue et al., 2014). They also applied a version of their model to video-to-text generation, but stopped short of proposing an end-to-end single network, using the intermediate role representation instead. Also, they showed results only on the narrow domain of cooking videos with a small set of pre-defined objects and actors. We also utilize a Long-Short Term Memory (LSTM) recurrent neural network (Hochreiter and Schmidhuber, 1997) to model sequence dynamics, but connect it directly to a deep convolutional neural network to process incoming video frames, avoiding supervised intermediate representations altogether. This model is similar to their image-to-text model, but we adapt it for video sequences.

Our proposed approach has several important advantages over existing video description work. The LSTM model, which has recently achieved state-of-the-art results on machine translation tasks (French and English (Sutskever et al., 2014)), effectively models the sequence generation task without requiring the use of fixed sentence templates as in previous work (Guadarrama et al., 2013). Pre-training on image and text data naturally exploits related data to supplement the limited amount of descriptive video currently available. Finally, the deep convnet, the winner of the ILSVRC2012(Russakovsky et al., 2014) image classification competition, provides a strong visual representation of objects, actions and scenes depicted in the video.

Our main contributions are as follows:

- We present the first end-to-end deep model for video-to-text generation that simultaneously learns a latent "meaning" state, and a fluent grammatical model of the associated language.
- We leverage still image classification and caption data and transfer deep networks learned on such data to the video domain.
- We provide a detailed evaluation of our model on the popular YouTube corpus (Chen and Dolan, 2011) and demonstrate a significant improvement over the state of the art.

[1]http://www.nytimes.com/2014/11/18/science/researchers-announce-breakthrough-in-content-recognition-software.html

[2]http://www.pcworld.com/article/2849838/microsoft-five-other-groups-race-toward-automated-image-captioning.html

2 Related Work

Most of the existing research in video description has focused on narrow domains with limited vocabularies of objects and activities (Kojima et al., 2002; Lee et al., 2008; Khan and Gotoh, 2012; Barbu et al., 2012; Ding et al., 2012; Khan and Gotoh, 2012; Das et al., 2013b; Das et al., 2013a; Rohrbach et al., 2013; Yu and Siskind, 2013). For example, (Rohrbach et al., 2013; Rohrbach et al., 2014) produce descriptions for videos of several people cooking in the same kitchen. These approaches generate sentences by first predicting a semantic role representation, e.g., modeled with a CRF, of high-level concepts such as the actor, action and object. Then they use a template or statistical machine translation to translate the semantic representation to a sentence.

Most work on "in-the-wild" online video has focused on retrieval and predicting event tags rather than generating descriptive sentences; examples are tagging YouTube (Aradhye et al., 2009) and retrieving online video in the TRECVID competition (Over et al., 2012). Work on TRECVID has also included clustering both video and text features for use in video retrieval, e.g., (Wei et al., 2010; Huang et al., 2013).

The previous work on the YouTube corpus we employ (Motwani and Mooney, 2012; Krishnamoorthy et al., 2013; Guadarrama et al., 2013; Thomason et al., 2014) used a two-step semantic role approach, first detecting a fixed tuple of role words, such as subject, verb, object, and scene, and then using a template to generate a grammatical sentence. They also utilize language models learned from large text corpora to aid visual interpretation as well as sentence generation. We compare our method to the best-performing method of (Thomason et al., 2014).

A recent paper by (Xu et al., 2015) extracts deep features from video and a continuous vector from language, and projects both to a joint semantic space. They apply their joint embedding to SVO prediction and generation, but do not provide quantitative generation results. Our network also learns a joint state vector implicitly, but also models sequence dynamics of the language, and can be extended to model sequence dynamics of video frames, although we did not evaluate this due to limited training data.

Predicting a natural language desription of still images has received considerable attention, with some of the earliest works by (Kulkarni et al., 2011; Yao et al., 2010). Propelled by successes of deep learning, several groups released record breaking results in just the past year (Donahue et al., 2014; Mao et al., 2014; Karpathy et al., 2014; Fang et al., 2014; Kiros et al., 2014; Vinyals et al., 2014; Kuznetsova et al., 2014).

In this work, we use deep recurrent nets (RNNs), which have recently demonstrated strong results for machine translation tasks using Long Short Term Memory (LSTM) RNNs (Sutskever et al., 2014; Cho et al., 2014). In contrast to traditional statistical MT (Koehn, 2010), RNNs naturally combine with vector-based representations, such as those for images and video. (Donahue et al., 2014; Vinyals et al., 2014) simultaneously proposed a multimodal analog of this model, with an architecture which uses a visual convnet to encode a deep state vector, and an LSTM to decode the vector into a natural language string.

Our approach to video to text generation is inspired by the work of (Donahue et al., 2014), who also applied a variant of their model to video-to-text generation, but stopped short of training an end-to-end model. Instead they converted the video to an intermediate role representation using a CRF, then decoded that representation into the language string. In contrast, we bypass detection of high-level roles and use the output of a deep convolutional network directly as the state vector that is decoded into a sentence. This avoids the need for labeling semantic roles, which can be difficult to detect in the case of very large vocabularies. It also allows us to first pre-train the model on a large image and caption database, and transfer the knowledge to the video domain where the corpus size is smaller.

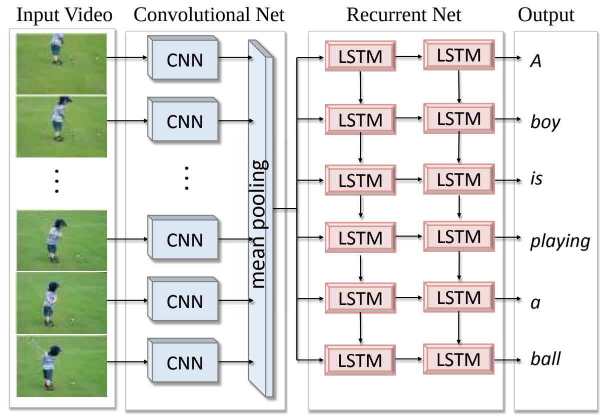

Figure 2: The structure of our video description network.

While (Donahue et al., 2014) only showed results on a narrow domain of cooking videos with a small set of pre-defined objects and actors, we generate sentences for open-domain YouTube videos with a vocabulary of thousands of words.

3 Approach

Figure 2 depicts our model for sentence generation from videos. Our framework is based on recent deep image description models in (Donahue et al., 2014; Vinyals et al., 2014) and extends them to generate sentences describing events in videos. These models work by first applying a feature transformation on an image to generate a fixed dimensional vector representation. They then use a sequence model, specifically a Recurrent Neural Network (RNN), to "decode" the vector into a sentence (i.e. a sequence of words). In this work, we apply the same principle of "translating" a visual vector into an English sentence and show that it works well for describing dynamic videos as well as static images.

We identify the most likely description for a given video by training a model to maximize the log likelihood of the sentence S, given the corresponding video V and the model parameters θ,

$$\theta^* = \underset{\theta}{\operatorname{argmax}} \sum_{(V,S)} \log p(S|V;\theta) \quad (1)$$

Assuming a generative model of S that produces each word in the sequence in order, the log probability of the sentence is given by the sum of the log probabilities over the words and can be expressed as:

$$\log p(S|V) = \sum_{t=0}^{N} \log p(S_{w_t}|V, S_{w_1}, \ldots, S_{w_{t-1}})$$

where S_{w_i} represents the i^{th} word in the sentence and N is the total number of words. Note that we have dropped θ for convenience.

A sequence model, such as an RNN is a natural choice to model $p(S_{w_t}|V, S_{w_1}, \ldots, S_{w_{t-1}})$. An RNN, parameterized by θ, maps an input x_t, and the previously seen words expressed as a hidden state or memory, h_{t-1} to an output z_t and an updated state h_t using a non-linear function f:

$$h_t = f_\theta(x_t, h_{t-1})$$

where $(h_0 = 0)$. In our work we use the highly successful Long Short-Term Memory (LSTM) net as the sequence model, since it has shown superior performance on tasks such as speech recognition (Graves and Jaitly, 2014), machine translation (Vinyals et al., 2014; Cho et al., 2014) and the more related task of generating sentence descriptions of images (Donahue et al., 2014; Vinyals et al., 2014). To be specific, we use two layers of LSTMs (one LSTM stacked atop another) as shown in Figure 2. We present details of the network in Section 3.1.

To convert videos to a fixed length representation (input x_t), we use a Convolutional Neural Network (CNN). In particular, we use the publicly available *Caffe* (Jia et al., 2014) reference model, a minor variant of *AlexNet* (Krizhevsky et al., 2012). The net is pre-trained on the 1.2M image ILSVRC-2012 object classification subset of the ImageNet dataset (Russakovsky et al., 2014) and hence provides a robust initialization for recognizing objects and thereby expedites training. We present details of how we apply the CNN model to videos in Section 3.2

3.1 LSTMs for sequence generation

A Recurrent Neural Network (RNN) is a generalization of feed forward neural networks to sequences. Standard RNNs learn to map a sequence of inputs $(x_1, \ldots, x_t)$ to a sequence of hidden states $(h_1, \ldots, h_t)$, and from the hidden states to a sequence of outputs $(z_1, \ldots, z_t)$ based on the following recurrences:

$$h_t = f(W_{xh}x_t + W_{hh}h_{t-1})$$
$$z_t = g(W_{zh}h_t)$$

where f and g are element-wise non-linear functions such as a sigmoid or hyperbolic tangent, x_t is a fixed length vector representation of the input, $h_t \in \mathbb{R}^N$ is the hidden state with N units, W_{ij} are the weights connecting the layers of neurons, and z_t the output vector.

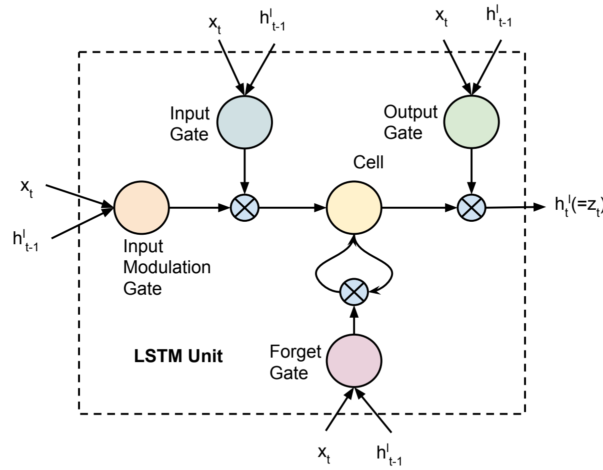

Figure 3: The LSTM unit replicated from (Donahue et al., 2014). The memory cell is at the core of the LSTM unit and it is modulated by the input, output and forget gates controlling how much knowledge is transferred at each time step.

RNNs can learn to map sequences for which the alignment between the inputs and outputs is known ahead of time (Sutskever et al., 2014) however it's unclear if they can be applied to problems where the inputs (x_i) and outputs (z_i) are of varying lengths. This problem is solved by learning to map sequences of inputs to a fixed length vector using one RNN, and then map the vector to an output sequence using another RNN. Another known problem with RNNs is that, it can be difficult to train them to learn long-range dependencies (Hochreiter et al., 2001). However, LSTMs (Hochreiter and Schmidhuber, 1997), which incorporate explicitly controllable memory units, are known to be able to learn long-range temporal dependencies. In our work we use the LSTM unit described in (Zaremba and Sutskever, 2014; Donahue et al., 2014) as shown in Figure 3.

At the core of the LSTM model is a memory cell c which encodes, at every time step, the knowledge of the inputs that have been observed up to that step. The cell is modulated by gates which are all sigmoidal, having range $[0, 1]$, and are applied multiplicatively. The gates determine whether the LSTM keeps the value from the gate (if the layer evaluates to 1) or discards it (if it evaluates to 0). The three gates – input gate (i) controlling whether the LSTM considers it's current input (x_t), the forget gate (f) allowing the LSTM to forget it's previous memory (c_{t-1}), and the output gate (o) deciding how much of the memory to transfer to the hidden state (h_t), all enable the LSTM to learn complex long-term dependencies. The recurrences for the LSTM are then defined as:

$$
\begin{aligned}
i_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1}) \\
f_t &= \sigma(W_{xf}x_t + W_{hf}h_{t-1}) \\
o_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1}) \\
c_t &= f_t \odot c_{t-1} + i_t \odot \phi(W_{xc}x_t + W_{hc}h_{t-1}) \\
h_t &= o_t \odot \phi(c_t)
\end{aligned}
$$

where σ is the sigmoidal non-linearity, ϕ is the hyperbolic tangent non-linearity, and $\odot$ represents the product with the gate value, and the weight matrices denoted by W_{ij}, the trained parameters.

3.2 CNN-LSTMs for video description

The framework for our two layer LSTM model for generating descriptions for videos is very similar to that used for generating image descriptions in (Donahue et al., 2014). We employ the LSTM to "decode" a visual feature vector representing the video to generate textual output. The first step in this process is to generate a fixed-length visual input that effectively summarizes a short video. For this we use a CNN, specifically a hybrid of the publicly available *Caffe* (Jia et al., 2014) reference model. The net is pre-trained on the 1.2M image ILSVRC-2012 classification subset of the ImageNet dataset. We extract the output of the fc_7 layer for a set of sample frames in the video (1 in every 10 frames) and perform a mean pooling over the frames to generate a single 4,096 dimension vector for each video. The resulting visual feature vector forms the input to the first LSTM layer. We stack another LSTM layer on top as in Figure 2, and the hidden state of the LSTM in the first layer is the input to the LSTM unit in the second layer. A word from the sentence forms the target of the output LSTM unit. In this work, we represent words using "one-hot" vectors (i.e 1-of-N coding).

Training and Inference: The two-layer LSTM model is trained to predict the next word S_{w_t} in the sentence given the visual features and the previous $t-1$ words, $p(S_{w_t}|V, S_{w_1}, \ldots, S_{w_{t-1}})$. During training the visual feature, sentence pair (V, S) is provided to the model, which then optimizes the log-likelihood (Equation 1) over the entire training dataset using stochastic gradient descent. At each time step, the visual feature vector x_t is input to the LSTM along with the previous time step's hidden state h_{t-1} and the LSTM emits the next hidden state vector h_t (and a word). Accordingly, inference must also be performed sequentially in the order $h_1 = f_W(x_1, 0)$, $h_2 = f_W(x_2, h_1)$, until the model emits the end-of-sentence (EOS) token at the final step T. In our model the output ($h_t = z_t$) of the second layer LSTM unit is used to obtain the emitted word. We apply the Softmax function, to get a probability distribution over the words in the vocabulary.

$$
p_t(w_t) = \text{Softmax } (z_t) = \frac{\exp(W_{zc}z_{t,c})}{\sum_{c' \in C} \exp(W_{zc}z_{t,c'})}
$$

where C in our case is the finite and discrete space determined by the vocabulary. Thus, at each step we sample a word according to p_t until we obtain the EOS token.

3.3 Transfer Learning from Captioned Images

Since the training data available for video description is quite limited (described in Section 4.1) we also leverage much larger datasets available for image description to train our LSTM model and then fine tune this initial model on the video dataset. Our LSTM model for images is the same as the one described above for single video frames (in Section 3.1, and 3.2). As with videos, we extract fc_7 layer features (4096 dimensional vector) from the network (Section 3.2) for the images. This forms the visual feature that is input to the 2nd layer LSTM description model. After the model is trained on the image dataset, we use the weights of the trained model to initialize the LSTM model for the video description task. Additionally, we reduce the learning rate on our LSTM model to allow for it to tune to the video dataset. This speeds up training and allows exploiting knowledge previously learned for image description.

4 Experiments

4.1 Datasets

Video dataset. We perform all our experiments on the Microsoft Research Video Description Corpus (Chen and Dolan, 2011). This video corpus is a collection of 1970 YouTube snippets. The duration of each clip is between 10 seconds to 25 seconds, typically depicting a single activity or a short sequence. The dataset comes with several human generated descriptions in a number of languages; we use the roughly 40 available English descriptions per video. This dataset (or portions of it) have been used in several prior works (Motwani and Mooney, 2012; Krishnamoorthy et al., 2013; Guadarrama et al., 2013; Thomason et al., 2014; Xu et al., 2015) on action recognition and video description tasks. For our task we pick 1200 videos to be used as training data, 100 videos for validation and 670 videos for testing, as used by the prior works on video description (Guadarrama et al., 2013; Thomason et al., 2014; Xu et al., 2015).

Domain adaptation, image description datasets. Since the number of videos for the description task is quite small when compared to the size of the datasets used by LSTM models in other tasks such as translation (Sutskever et al., 2014) (12M sentences), we use data from the Flickr30k and COCO2014 datasets for training and learn to adapt to the video dataset by fine-tuning the image description models. The Flickr30k (Peter Young and Hockenmaier, 2014) dataset has about 30,000 images, each with 5 or more descriptions. We hold out 1000 images at random for validation and use the remaining for training. In addition to this, we use the recent COCO2014 (Lin et al., 2014) image description dataset consisting of 82,783 training images and 40,504 validation images, each with 5 or more sentence descriptions. We perform ablation experiments by training models on each dataset individually, and on the combination and report results on the YouTube video test dataset.

4.2 Models

HVC This is the Highest Vision Confidence model described in (Thomason et al., 2014). The model uses strong visual detectors to predict confidence over 45 subjects, 218 verbs and 241 objects.

FGM (Thomason et al., 2014) also propose a factor graph model (FGM) that combines knowledge mined from text corpora with visual confidence from the HVC model using a factor graph and performs probabilistic inference to determine the most likely subject, verb, object and scene tuple. They then use a simple template to generate a sentence from the tuple. In this work, we compare the output of our model to the subject, verb, object words predicted by the HVC and FGM models and the sentences generated from the SVO triple.

Our LSTM models We present four main models. LSTM-YT is our base two-layer LSTM model trained on the YouTube video dataset. LSTM-YT$_{flickr}$ is the model trained on the Flickr30k (Peter Young and Hockenmaier, 2014) dataset, and fine tuned on the YouTube dataset as descibed in Section 3.3. LSTM-YT$_{coco}$ is first trained on the COCO2014 (Lin et al., 2014) dataset and then fine-tuned on the video dataset. Our final model, LSTM-YT$_{cocoflickr}$ is trained on the combined data of both the Flickr and COCO models and is tuned on YouTube. To compare the overlap in content between the image dataset and YouTube dataset, we use the model trained on just the Flickr images (LSTM$_{flickr}$) and just the COCO images (LSTM$_{coco}$) and evaluate their performance on the videos.

4.3 Evaluation Metrics and Results

SVO accuracy. Early works (Krishnamoorthy et al., 2013; Guadarrama et al., 2013) that reported results on the YouTube dataset compared their method based on how well their model could predict the subject, verb, and object (SVO) depicted in the video. Since these models first predicted the content (SVO triples) and then generated the sentences, the S,V,O accuracy captured the quality of the content generated by the models. However, in our case our sequential LSTM directly outputs the sentence itself (without an explicit *surface realization* phase); so we extract the S,V,O from the dependency parse of the generated sentence. We present, in Table 1 and Table 2, the accuracy of S,V,O words comparing the performance of our model against any valid ground truth triple and the most frequent triple found in human description for each video. The latter evaluation was also reported by (Xu et al., 2015), so we include it here for comparison.

[3]They evaluate against a filtered set of groundtruth SVO words which provides a tiny boost to their scores.

Model	S%	V%	O%
HVC (Thomason et al., 2014)	86.87	38.66	22.09
FGM (Thomason et al., 2014)	**88.27**	37.16	24.63
LSTM$_{flickr}$	79.95	15.47	13.94
LSTM$_{coco}$	56.3	6.9	14.86
LSTM-YT	79.4	35.52	20.59
LSTM-YT$_{flickr}$	84.92	38.66	21.64
LSTM-YT$_{coco}$	86.58	42.23	**26.69**
LSTM-YT$_{coco+flickr}$	87.27	**42.79**	24.23

Table 1: SVO accuracy: Binary SVO accuracy compared against any valid S,V,O triple in the ground truth descriptions. We extract S,V,O values from sentences output by our model using a dependency parser.

Model	S%	V%	O%
HVC (Thomason et al., 2014)	76.57	22.24	11.94
FGM (Thomason et al., 2014)	76.42	21.34	12.39
JointEmbed[3] (Xu et al., 2015)	**78.25**	24.45	11.95
LSTM$_{flickr}$	70.8	10.02	7.84
LSTM$_{coco}$	47.44	2.85	7.05
LSTM-YT	71.19	19.4	9.7
LSTM-YT$_{flickr}$	75.37	21.94	10.74
LSTM-YT$_{coco}$	76.01	23.38	**14.03**
LSTM-YT$_{coco+flickr}$	75.61	**25.31**	12.42

Table 2: SVO accuracy: Binary SVO accuracy compared against most frequent S,V,O triple in the ground truth descriptions. We extract S,V,O values from parses of sentences output by our model using a dependency parser.

Sentence Generation. To evaluate the generated sentences we use the BLEU (Papineni et al., 2002) and METEOR (Banerjee and Lavie, 2005) scores against all ground truth sentences. BLEU is the metric that is seen more commonly in image description literature, but a more recent study (Elliott and Keller, 2014) has shown METEOR to be a better evaluation metric. Since both metrics have been shown to correlate well with human evaluations, we compare the generated sentences using both and present our results in Table 3.

Human Evaluation. We used Amazon Mechanical Turk to also collect human judgements. We created a task which employed three Turk workers to watch each video, and rank sentences generated by the different models from "Most Relevant" (1) to "Least Relevant" (5). We also evaluate sentences on grammatical correctness. We created a different task which required workers to rate sentences based on grammar. This task displayed only the sentences and did not show any video. Here, workers had to choose a rating between 1-5 for each sentence. We discard responses from workers who fail gold-standard items and report the mean ranking/rating for each of the evaluated models in Table 4.

Model	BLEU	METEOR
FGM (Thomason et al., 2014)	13.68	23.9
LSTM-YT	31.19	26.87
LSTM-YT$_{flickr}$	32.03	27.87
LSTM-YT$_{coco}$	**33.29**	**29.07**
LSTM-YT$_{coco+flickr}$	**33.29**	28.88

Table 3: Scores for BLEU at 4 (combined n-gram 1-4), and METEOR scores from automated evalutation metrics comparing the quality of the generation. All values are reported as percentage (%).

Model	Relevance	Grammar
FGM (Thomason et al., 2014)	3.20	**3.99**
LSTM-YT	2.88	3.84
LSTM-YT$_{coco}$	**2.83**	3.46
LSTM-YT$_{coco+flickr}$	-	3.64
GroundTruth	1.10	4.61

Table 4: Human evaluation mean scores. Sentences were ranked based on relevance (lower is better) and were rated for grammatical correctness (higher is better).

5 Discussion

Image only models. The models trained purely on the image description data LSTM$_{flickr}$ and LSTM$_{coco}$ achieve lower accuracy on the verbs and objects (Tables 1, 2) since the YouTube videos encompass a wider domain and a variety of actions not detectable from static images.

Base LSTM model. We note that in the SVO binary accuracy metrics (Tables 1 and 2), the base LSTM model (LSTM-YT) achieves a slightly lower accuracy compared to prior work. This is likely due to the fact that previous work explicitly optimizes to identify the best subject, verb and object for a video; whereas the LSTM model is trained on objects and actions jointly in a sentence and needs to learn to interpret these in different contexts. However, with regard to the generation metrics BLEU and METEOR, training based on the full sentence helps the LSTM model develop fluency and vocabulary similar to that seen in the training descriptions and allows it to outperform the template based generation.

Transferring helps. From our experiments, it is clear that learning from the image description data improves the performance of the model in all criteria of evaluation. We present a few examples demonstrating this in Figure 4. The model that was pre-trained on COCO2014 shows a larger performance improvement, indicating that our model can effectively leverage a large auxiliary source of training data to improve its object and verb predictions. The model pre-trained on the combined data of Flickr30k and COCO2014 shows only a marginal improvement, perhaps due to overfitting. Adding dropout as in (Vinyals et al., 2014) is likely to help prevent overfitting and improve performance.

From the automated evaluation in Table 3 it is clear that the fully deep video-to-text generation models outperform previous work. As mentioned previously, training on the full sentences is probably the main reason for the improvements.

Human evaluation. We note that the sentences generated by our model have been ranked more relevant (Table 4) to the content in the video than previous models. However, there is still a significant gap between the human ground truth sentence and the ones generated by the LSTM models. Additionally, when we ask Turkers to rate sentences (no video) on grammatical correctness, the template based FGM (Thomason et al., 2014) achieves the highest ratings. This can be explained by the fact that their work use a template technique to generate sentences from content, and is hence grammatically well formed. Our model sometimes predicts prepositions and articles more frequently, resulting in duplicates and hence incorrect grammar.

6 Conclusion

In this paper we have proposed a model for video description which uses neural networks for the entire pipeline from pixels to sentences. In an extensive experimental evaluation, we showed that our approach generates better sentences than related approaches. We also showed that exploiting image description data improves performance compared to relying only on video description data. We will re-

Figure 4: Examples to demonstrate effectiveness of transferring from the image description domain. YT refer to the LSTM_YT, YT_C to the LSTM-YT$_{coco}$, and YT_CF to the LSTM-YT$_{coco+flickr}$ models. GT is a random human description in the ground truth. Bottom two examples show how transfer can overfit. Thus, while base LSTM_YT model detects water and monkey, the YT_C and YT_CF models fail to describe the event completely.

lease our *Caffe*-based implementation upon publication, as well as the model and generated sentences.

References

H. Aradhye, G. Toderici, and J. Yagnik. 2009. Video2text: Learning to annotate video content. In *IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 144–151.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72.

Andrei Barbu, Alexander Bridge, Zachary Burchill, Dan Coroian, Sven Dickinson, Sanja Fidler, Aaron Michaux, Sam Mussman, Siddharth Narayanaswamy, Dhaval Salvi, Lara Schmidt, Jiangnan Shangguan, Jeffrey Mark Siskind, Jarrell Waggoner, Song Wang, Jinlian Wei, Yifan Yin, and Zhiqi Zhang. 2012. Video in sentences out. In *Association for Uncertainty in Artificial Intelligence (UAI)*.

David L. Chen and William B. Dolan. 2011. Collecting highly parallel data for paraphrase evaluation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 190–200. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*.

P. Das, R. K. Srihari, and J. J. Corso. 2013a. Translating related words to videos and back through latent topics. In *Proceedings of Sixth ACM International Conference on Web Search and Data Mining (WSDM)*.

P. Das, C. Xu, R. F. Doell, and J. J. Corso. 2013b. A thousand frames in just a few words: Lingual description of videos through latent topics and sparse object stitching. In *CVPR*.

D. Ding, F. Metze, S. Rawat, P.F. Schulam, S. Burger, E. Younessian, L. Bao, M.G. Christel, and A. Hauptmann. 2012. Beyond audio and video retrieval: towards multimedia summarization. In *Proceedings of the 2nd ACM International Conference on Multimedia Retrieval (ICMR)*, page 2. ACM.

Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. 2013. Decaf: A deep convolutional activation feature for generic visual recognition. *arXiv preprint arXiv:1310.1531*.

Jeff Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. 2014. Long-term recurrent convolutional networks for visual recognition and description. *CoRR*, abs/1411.4389.

Desmond Elliott and Frank Keller. 2014. Comparing automatic evaluation measures for image description. In *ACL*, volume 2, pages 452–457.

Hao Fang, Saurabh Gupta, Forrest N. Iandola, Rupesh Srivastava, Li Deng, Piotr Dollár, Jianfeng Gao, Xiaodong He, Margaret Mitchell, John C. Platt, C. Lawrence Zitnick, and Geoffrey Zweig. 2014. From captions to visual concepts and back. *CoRR*, abs/1411.4952.

Alex Graves and Navdeep Jaitly. 2014. Towards end-to-end speech recognition with recurrent neural networks. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1764–1772.

Sergio Guadarrama, Niveda Krishnamoorthy, Girish Malkarnenkar, Subhashini Venugopalan, Raymond Mooney, Trevor Darrell, and Kate Saenko. 2013. Youtube2text: Recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition. In *IEEE International Conference on Computer Vision (ICCV)*, December.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Sepp Hochreiter, Yoshua Bengio, Paolo Frasconi, and Jürgen Schmidhuber. 2001. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies.

Haiqi Huang, Yueming Lu, Fangwei Zhang, and Songlin Sun. 2013. A multi-modal clustering method for web videos. In *Trustworthy Computing and Services*, pages 163–169. Springer.

Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. 2014. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*.

Andrej Karpathy, Armand Joulin, and Li Fei-Fei. 2014. Deep fragment embeddings for bidirectional image sentence mapping. *NIPS*.

Muhammad Usman Ghani Khan and Yoshihiko Gotoh. 2012. Describing video contents in natural language. *Proceedings of the Workshop on Innovative Hybrid Approaches to the Processing of Textual Data*, pages 27–35.

Ryan Kiros, Ruslan Salakhuditnov, and Richard. S Zemel. 2014. Unifying visual-semantic embeddings with multimodal neural language models. *arXiv preprint arXiv:1411.2539*.

Philipp Koehn. 2010. *Statistical Machine Translation*. Cambridge University Press.

A. Kojima, T. Tamura, and K. Fukunaga. 2002. Natural language description of human activities from video images based on concept hierarchy of actions. *International Journal of Computer Vision (IJCV)*, 50(2):171–184.

Niveda Krishnamoorthy, Girish Malkarnenkar, Raymond J. Mooney, Kate Saenko, and Sergio Guadarrama. 2013. Generating natural-language video descriptions using text-mined knowledge. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 541–547.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. ImageNet classification with deep convolutional neural networks. In *NIPS*.

Girish Kulkarni, Visruth Premraj, Sagnik Dhar, Siming Li, Yejin Choi, Alexander C Berg, and Tamara L Berg. 2011. Baby talk: Understanding and generating simple image descriptions. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1601–1608. IEEE.

Polina Kuznetsova, Vicente Ordonez, Tamara L Berg, UNC Chapel Hill, and Yejin Choi. 2014. Treetalk: Composition and compression of trees for image descriptions. *Transactions of the Association for Computational Linguistics*, 2(10):351–362.

M.W. Lee, A. Hakeem, N. Haering, and S.C. Zhu. 2008. Save: A framework for semantic annotation of visual events. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. *arXiv preprint arXiv:1405.0312*.

Junhua Mao, Wei Xu, Yi Yang, Jiang Wang, and Alan L Yuille. 2014. Explain images with multimodal recurrent neural networks. *arXiv preprint arXiv:1410.1090*.

Tanvi S. Motwani and Raymond J. Mooney. 2012. Improving video activity recognition using object recognition and text mining. In *Proceedings of the 20th European Conference on Artificial Intelligence (ECAI)*, pages 600–605.

Paul Over, George Awad, Martial Michel, Jonathan Fiscus, Greg Sanders, B Shaw, Alan F. Smeaton, and Georges Quéenot. 2012. Trecvid 2012 – an overview of the goals, tasks, data, evaluation mechanisms and metrics. In *Proceedings of TRECVID 2012*. NIST, USA.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *ACL*.

Micah Hodosh Peter Young, Alice Lai and Julia Hockenmaier. 2014. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *TACL*, 2:67–78.

Marcus Rohrbach, Wei Qiu, Ivan Titov, Stefan Thater, Manfred Pinkal, and Bernt Schiele. 2013. Translating video content to natural language descriptions. In *IEEE International Conference on Computer Vision (ICCV)*, pages 433–440.

Anna Rohrbach, Marcus Rohrbach, Wei Qiu, Annemarie Friedrich, Manfred Pinkal, and Bernt Schiele. 2014. Coherent multi-sentence video description with variable level of detail. In *German Conference on Pattern Recognition (GCPR)*, September. Oral.

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. 2014. ImageNet Large Scale Visual Recognition Challenge.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems (NIPS)*.

J. Thomason, S. Venugopalan, S. Guadarrama, K. Saenko, and R.J. Mooney. 2014. 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.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2014. Show and tell: A neural image caption generator. *CoRR*, abs/1411.4555.

Shikui Wei, Yao Zhao, Zhenfeng Zhu, and Nan Liu. 2010. Multimodal fusion for video search reranking. *IEEE Transactions on Knowledge and Data Engineering,*, 22(8):1191–1199.

R. Xu, C. Xiong, W. Chen, and J. J. Corso. 2015. Jointly modeling deep video and compositional text to bridge vision and language in a unified framework. In *Proceedings of AAAI Conference on Artificial Intelligence*.

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

Haonan Yu and Jeffrey Mark Siskind. 2013. Grounded language learning from videos described with sentences. In *ACL*.

Wojciech Zaremba and Ilya Sutskever. 2014. Learning to execute. *arXiv preprint arXiv:1410.4615*.

Matthew D Zeiler and Rob Fergus. 2014. Visualizing and understanding convolutional networks. In *Computer Vision–ECCV 2014*, pages 818–833. Springer.