Natural Language Video Description using Deep Recurrent Neural Networks

Subhashini Venugopalan
University of Texas at Austin
vsub@cs.utexas.edu

Doctoral Dissertation Proposal

Supervising Professor: Raymond J. Mooney

Abstract

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 a sentence description is a very complex problem. The goal of my research is to develop models that can automatically generate natural language (NL) descriptions for events in videos. As a first step, this proposal presents deep recurrent neural network models for video to text generation. I build on recent "deep" machine learning approaches to develop video description models using a unified deep neural network with both convolutional and recurrent structure. This technique treats the video domain as another "language" and takes a machine translation approach using the deep network to translate videos to text. In my initial approach, I adapt a model that can learn on images and captions to transfer knowledge from this auxiliary task to generate descriptions for short video clips. Next, I present an end-to-end deep network that can jointly model a sequence of video frames and a sequence of words. The second part of the proposal outlines a set of models to significantly extend work in this area. Specifically, I propose techniques to integrate linguistic knowledge from plain text corpora; and attention methods to focus on objects and track their interactions to generate more diverse and accurate descriptions. To move beyond short video clips, I also outline models to process multi-activity movie videos, learning to jointly segment and describe coherent event sequences. I propose further extensions to take advantage of movie scripts and subtitle information to generate richer descriptions.

Contents

1 Introduction **2**

2 Background and Related Work **4**

2.1 Background: Language and Vision 4

2.2 Video Description 4

2.3 Integrating Language and Vision using Factor Graphs 5

2.4 Background: Long Short-Term Memory Networks 5

3 Deep Recurrent Neural Networks for Video Description **8**

3.1 Sequence modeling using LSTMs 8

3.2 Translating Videos to Natural Language using LSTMs 9

3.3 Sequence to Sequence – Video to Text 13

3.4 Summary of Completed Work 19

4 Proposed Work **21**

4.1 Using Statistical Linguistic Knowledge to Aid Video Description 21

4.2 Using Attention to Generate Descriptions 24

4.3 Multi-sentential Descriptions of Longer Videos 25

4.4 Exploiting Movie Scripts and Sub-titles to Enhance DVS Descriptions for Movies 27

5 Conclusion **30**

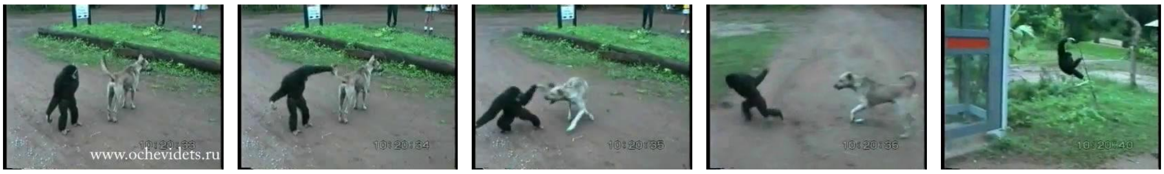
Chapter 1

Introduction

The ability to describe videos in natural language (NL) enables many important applications such as content-based video retrieval, video segmentation and segment indexing, textual summarization of video clips, video description for the visually impaired, and automated video surveillance. The past year has seen a marked increase in work on natural-language image description and a growing interest in video description. We develop the first fully deep model for video captioning. These models have achieved promising results on the task of generating natural language descriptions of short video clips. Yet there is very limited existing work on solving the problem at scale, to recognize and capture interactions between objects, particularly for large vocabulary, "in-the-wild" video collections, and long (possibly movie-length) sequences. In this proposal, I develop and outline improved methods for natural-language video description by combining the latest techniques in computer vision and natural language processing (NLP) and leveraging transformative advances in "deep" machine learning.

Most prior work on NL-description of visual data focus on static images [106, 30, 53, 58, 54, 105]. In the last year alone, several deep neural network based methods [25, 15, 44, 47, 55, 65, 80, 98] announced breakthrough results on the task of describing images with a single sentence. In contrast, video description has seen far less attention, and many of the recent deep neural network approaches to captioning do not address the problem of detecting sequences of activities and describing them in full sentences. Existing research in video description has focused on narrow domains with limited vocabularies of objects and activities [49, 56, 45, 7, 24, 45, 20, 19, 80, 108]. 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.

My completed research takes a step towards addressing some of these challenges. Describing activities depicted in video requires integrating both visual and linguistic capabilities, as seen from the example in Figure 1.1. In previous work [91], we first focus on addressing the issue of describing open-domain videos with large vocabularies by integrating linguistic knowledge with visual recognition. Using a two step approach, we build visual classifiers to recognize several hundred objects, activities and scenes in videos. Then, to determine salient objects and activities, we combine knowledge mined from text corpora with confidences from the visual classifiers using a factor graph to estimate the best



Description: A monkey pulls a dog's tail and is chased by the dog.

Figure 1.1: Describing activities depicted in videos require integration of both visual and linguistic capabilities. This example presents frames from a YouTube video clip and a description that a good model should generate.

subject-verb-object-scene (SVOP) tuple that can be used to describe a short video clip.

In recent work [95] we use deep recurrent neural networks based on Long Short Term Memory (LSTM, [39]) to learn what is worth describing directly from video and sentence pairs. Additionally, we overcome the limitation of reduced video training data, by transferring knowledge from the data rich auxiliary task of image captioning to further improve results on the video description task. In [96], we extend this deep video captioning framework further, by proposing a more robust model that can capture the joint dependencies of a sequence of frames and a corresponding sequence of words.

In this proposal, I first develop extensions to incorporate prior linguistic knowledge into deep video captioning models. I propose multiple techniques to integrate knowledge from plain text corpora to improve video description. Second, current video captioning models do not explicitly track and capture interactions between objects. To address this, I propose to use attention models [5, 68, 103] that can learn where to look, to attend to objects and activities in videos to generate a more accurate description of the event in the video. Next, to move beyond single sentence descriptions of short video clips, I outline models that can process multi-activity videos learning to simultaneously segment and describe coherent event sequences. Additionally, I propose to investigate schemes that use movie scripts and subtitles to generate more accurate descriptions of scenes in movies.

Organization

The remainder of this work is organized in three chapters: Chapter 2 presents background, related work and our initial approach on integrating language and vision using factor graphs to generate descriptions; Chapter 3 discusses our recent recurrent neural network approaches to video captioning; Chapter 4 details my proposed work presenting short term extensions to incorporate statistical language models and models of attention, and then proceeds to discuss long-term efforts to address multi-activity videos and further extensions to improve DVS descriptions.

Chapter 2

Background and Related Work

In this chapter, I first review early research on integrating language and vision to generate image and video description. Next, I present some initial models for video description. Then I will briefly describe joint prior work on integrating language statistics with visual detection confidences to generate descriptions of videos.

2.1 Background: Language and Vision

Both natural language processing (NLP) and computer vision (CV) have made great strides in recent years [43, 31], leveraging transformative advances in machine learning and the availability of very large datasets. Now, the two fields are rapidly encroaching upon each other: language is increasingly focused on "grounding" meaning in perception, and vision is exploiting linguistic ontologies and trying to "tell a story" from imagery, relating objects, activities, people, and scenes. Until last year, there was a small but growing body of work at the intersection of NLP and CV on topics like connecting words to pictures [8, 9, 22], describing images in natural language (NL) [30, 53, 58], and comprehending NL instructions in terms of robot perception and action [66, 90, 52, 36].

This past year, saw a dramatic increase in image captioning and retrieval works [25, 98, 44, 47, 29, 64] owing to the release of large image captioning datasets MSCOCO [59] and Flickr30k [40]. More recently there have also been a number of works in image question answering [62, 1, 77, 109]. In comparison, video description has received far less attention.

2.2 Video Description

Large-vocabulary video activity description presents unique challenges, including modeling dynamics and actor-action-object relationships from limited training data, as well as dealing with polysemy and ambiguity. Results on activity description in video have been restricted to a small set of actions and objects[45, 56, 49, 24, 50, 20, 19]. Work on large-vocabulary description has focused mostly on nouns/adjectives, specifically, early work on videos considered tagging videos with metadata [3] and clustering captions and videos [41, 71, 101] for retrieval tasks.

Work on video description used hand-crafted templates, grammars, and/or rules, work in fairly constrained domains. For example, [7, 108] produce sentential descriptions for short videos but only recognizes a limited set of (5-10) objects and activities and uses a manually engineered grammar to generate a fairly restricted range of descriptive sentences.

Several previous methods for generating sentence descriptions divided the task into two parts. The first is the *content generation* where they identify the most salient objects that need to be described. The second is *surface realization* where they generate a sentence based on the identified content. For example, [35, 50] use a two stage pipeline that first identifies the semantic content (subject, verb, object) and then generates a sentence based on a template. In [50] they first train individual classifiers to identify candidate objects, actions and scenes. They then use an *n-gram* language model to determine the best subject-verb-object for describing a video. This is then used to generate a sentence. [50] used a limited set of videos containing a small set of 20 entities. [35] was the first to describe "in-the-wild" videos with large vocabularies. showed an advantage of using linguistic knowledge only for the case of "zero shot activity recognition," in which the appropriate verb for describing the activity was never seen during training.

2.3 Integrating Language and Vision using Factor Graphs

In our prior work [91], we address the task of video description by first recognizing objects, activities and scenes in the video; and then generate a sentence description based on the most likely subject-verb-object-place (SVOP) tuple. We follow the method in [35] to first build object and action classifiers. For detecting objects, we use ObjectBank [57] and the LLC-10k classifiers of [23] trained on ImageNet 2011 with 10k object categories. Our action classifiers used Dense Trajectories [100], and the features for scene recognition were based on [102]. We trained non-linear SVMs [12] to obtain confidences over 45 subjects, 218 verbs, 241 objects and 12 scenes, thus covering a large vocabulary.

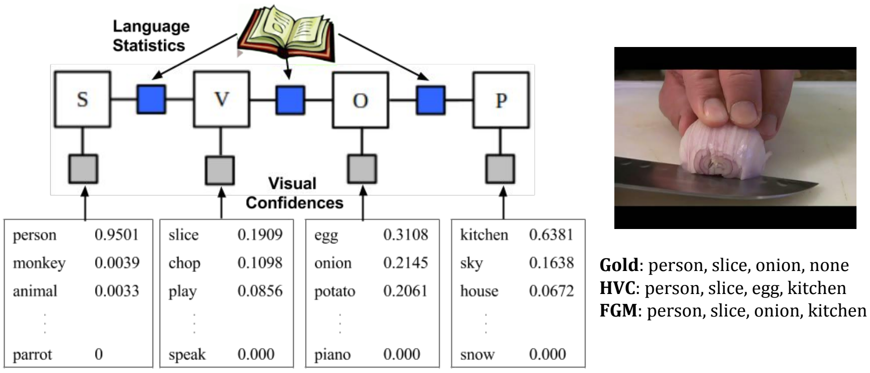
To improve recognition accuracy, we used text-mined knowledge to bias the collective labeling of each test video with a coherent subject (S), verb (V), object (O), and scene/place (P). We used the Stanford dependency parser [21] to syntactically analyze over 35GB of raw text and extracted bigram co-occurrence statistics for SV, VO, and OP word pairs. These determine the language potentials. We then use a factor graph to systematically integrate visual detection confidences with probabilistic knowledge mined from text corpora. During testing, efficient exact MAP inference for this simple linear-chain model is used to predict the most probable (SVOP) description as illustrated in Figure 2.1.

2.4 Background: Long Short-Term Memory Networks

The framework of our proposed models is based on deep recurrent neural networks in particular Long Short-Term Memory (LSTM) units. LSTM based recurrent neural networks have recently shown superior performance on tasks such as speech recognition [34], machine translation [89, 16] and the more related task of generating sentence descriptions of images [25, 98]. This section aims to provide an overview of recurrent neural network, in particular, Long Short-Term Memory (LSTMs) networks with focus on sequence modeling.

2.4.1 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 $(z_1, \ldots, z_t)$



Gold: person, slice, onion, none
HVC: person, slice, egg, kitchen
FGM: person, slice, onion, kitchen

Figure 2.1: The factor graph model for estimating the most likely subject-verb-object-place (SVOP) tuple by combining confidences from visual detectors and statistics from language. (Right) The factor graph model correctly predicts “person, slice, onion, kitchen” whereas the vision system places a higher confidence on “person, slice, egg, kitchen”. HVC refers to the Highest Vision Confidence system, based on just the visual classifiers. FGM refers to the factor graph model’s prediction.

based on the following recurrences:

$$h_t = f(W_{xh}x_t + W_{hh}h_{t-1}) \tag{2.1}$$

$$z_t = g(W_{zh}h_t) \tag{2.2}$$

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.

RNNs can learn to map sequences for which the alignment between the inputs and outputs is known ahead of time [89] 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. This is popularly referred to as the “encoder-decoder” framework. Another known problem with RNNs is that, it can be difficult to train them to learn long-range dependencies [38]. However, LSTMs [39], 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 in Figure 2.2, described in [110], and [25].

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 its current input (x_t), the forget gate (f) allowing the LSTM to forget its 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:

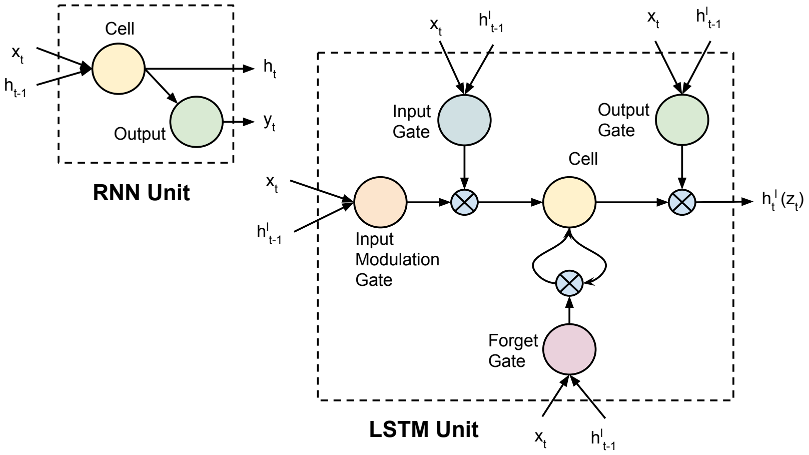


Figure 2.2: The RNN and LSTM units (replicated from [25]). The memory cell is at the core of the LSTM unit and it is modulated by the input, output and forget gates controlling how much knowledge is transferred at each time step.

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1}) \tag{2.3}$$

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1}) \tag{2.4}$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1}) \tag{2.5}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \phi(W_{xc}x_t + W_{hc}h_{t-1}) \tag{2.6}$$

$$h_t = o_t \odot \phi(c_t) \tag{2.7}$$

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

In the next chapter we propose two models that employ the LSTM to “decode" a visual feature vector representing the video to generate textual output.

Chapter 3

Deep Recurrent Neural Networks for Video Description

In this chapter we present two models that use deep recurrent neural networks based on Long Short Term Memory (LSTM, [39]) to generate video descriptions by learning directly from video and sentence pairs. Our models take inspiration from recent breakthroughs in machine translation [89] and image-captioning [25], and treats the input video as another "language" and translates the visual input to a sequence of words. First, I describe LSTMs to model sequence data. Next, I present an LSTM based model that's adapted from an image-captioning network [25] and transfer knowledge from the data rich auxiliary task of image captioning to generate descriptions for short video clips. In the final section, I present an end-to-end deep network that can jointly model a sequence of video frames and a sequence of words.

3.1 Sequence modeling using LSTMs

Our framework is based on deep image description models in [25];[98] 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,

$$\theta^* = \underset{\theta}{\operatorname{argmax}} \sum_{(V,S)} \log\, p(S|V;\theta) \tag{3.1}$$

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 would be apt to model $p(S_{w_t}|V, S_{w_1}, \ldots, S_{w_{t-1}})$, and we choose an RNN. 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:

$$h_t = f_\theta(x_t, h_{t-1}) \tag{3.2}$$

where $(h_0 = 0)$. In this work we use the highly successful Long Short-Term Memory (LSTM) net as the sequence model (Section 2.4.1), since it has shown superior performance on tasks such as speech recognition [34], machine translation [89, 16] and the more related task of generating sentence descriptions of images [25, 98]. We present details of the network in Section 2.4.1. To convert videos to a fixed length representation (input x_t), we use a Convolutional Neural Network (CNN). Each of our two models presented in this chapter uses a different approach to handle input videos, and details of how we apply it is presented when describing the model.

3.2 Translating Videos to Natural Language using LSTMs

In this section we build a model to translate from video pixels to natural language with a single deep neural network. We use deep recurrent nets (RNNs), which have recently demonstrated strong results for machine translation (MT) tasks using Long Short Term Memory (LSTM) RNNs [89, 16]. In contrast to traditional statistical MT [48], RNNs naturally combine with vector-based representations, such as those for images and video. [25] and [98] simultaneously proposed a multimodal analog of this model, with an architecture which uses a visual CNN/convnet to encode a deep state vector, and an LSTM to decode the vector into a sentence. Our model takes inspiration from both these approaches, and adapts their techniques for video description.

Deep NNs can learn powerful features [26, 111], but require a lot of supervised training data. However, annotated video data with descriptions is scarce. We address the problem by transferring knowledge from auxiliary tasks at different levels in the network. Each frame of the video is modeled by a convolutional (spatially-invariant) network pre-trained on 1.2M+ images with category labels [51]. The meaning state and sequence of words is modeled by a recurrent (temporally invariant) deep network pre-trained on 100K+ Flickr [40] and COCO [59] images with associated sentence captions. We show that such knowledge transfer significantly improves performance on the task of video description.

Our approach has several important advantages over existing video description work. The LSTM model effectively models the sequence generation task without requiring the use of fixed sentence templates as in previous work [50, 35, 91]. 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 [81] image classification competition, provides a strong visual representation of objects, actions and scenes depicted in the video.

The main contributions of this approach are:

- It is the first end-to-end deep model for video-to-text generation.
- It leverages still image classification and caption data, and transfers knowledge learned on such data to the video description task.

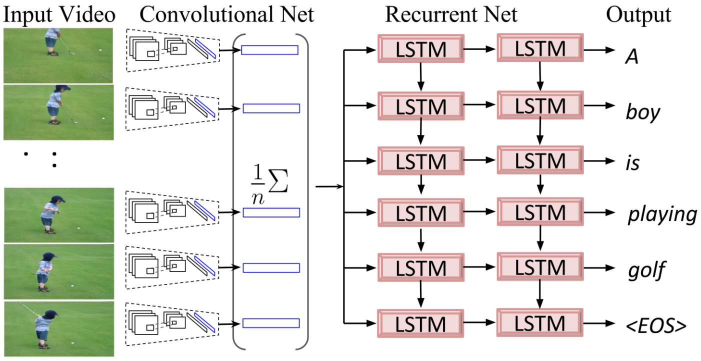


Figure 3.1: The structure of our video description network. We extract fc$_7$ features for each frame, mean pool the features across the entire video and input this at every time step to the LSTM network. The LSTM outputs one word at each time step, based on the video features (and the previous word) until it picks the end-of-sentence tag.

- We provide a detailed evaluation of our model on a collection of YouTube videos [13] and demonstrate that it significantly improves over previous state of the art approaches discussed in Chapter 2.

3.2.1 CNN-LSTMs for video description

Figure 3.1 depicts our model for sentence generation from videos. We choose a two layer LSTM model for the video description task. Our choice on the number and size of layers is based on experiments in [25] comparing different architectures for image captioning. 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 the publicly available *Caffe* [42] reference model, a minor variant of *AlexNet* [51]. The net is pre-trained on the 1.2M image ILSVRC-2012 object classification subset of the ImageNet dataset [81] and hence provides a robust initialization for recognizing objects and thereby expedites training. We sample frames in the video (1 in every 10 frames) and extract the activations from the fully connected layer (fc$_7$) just prior to the classification layer; 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 3.1, 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, where is N is the vocabulary size).

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 (Eq. (3.1)) over the entire training dataset using stochastic gradient descent. At each time step, the input x_t is fed 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). For the first layer of the LSTM, the input x_t is the concatenation of the visual feature vector and the previous encoded word ($S_{w_{t-1}}$, the ground truth word during training and the predicted word during test time). For the second layer of the LSTM, the input x_t is the value of z_t from the first layer. 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 w in the vocabulary D.

$$p(w|z_t) = \frac{\exp(W_w z_t)}{\sum_{w' \in D} \exp(W_{w'} z_t)} \tag{3.3}$$

where W_w is a learnt embedding vector for word w. At test time, we choose the word $\hat{w}$ with the maximum probability for each time step t until we obtain the EOS token.

3.2.2 Transfer Learning from Captioned Images

Since the training data available for video description is quite limited (described in Section 3.2.3), we also leverage much larger datasets available for image captioning to train our LSTM model and then fine tune it on the video dataset. Our LSTM model for images is the same as the one described above for single video frames (in Section 2.4.1, and Section 3.2.1). As with videos, we extract fc$_7$ layer features (4096 dimensional vector) from the network (Section 3.2.1) for the images. This forms the visual feature that is input to the 2-layer LSTM description model. The vocabulary is the combined set of words in the video and image datasets. 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 it to tune to the video dataset. This speeds up training and allows exploiting knowledge previously learned for image description.

3.2.3 Evaluation

Video dataset. We perform all our experiments on the Microsoft Research Video Description Corpus (MSVD) [13]. 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 [69, 50, 35, 91, 104] 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 [35, 91, 104].

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 [89] (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 [40] 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 [59] 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.

Models We compare our models against previous state-of-the-art factor graph model (**FGM**) proposed in [91] (Section 2.3).

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 [40] dataset, and fine tuned on the YouTube dataset as desrcibed in Section 3.2.2. LSTM-YT$_{coco}$ is first trained on the COCO2014 [59] 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. The models trained on image datasets alone, without being tuned on the video corpus, perform rather poorly. The results of these can be found in the paper [95].

Sentence Generation. To evaluate the generated sentences we use automated Machine Translation metrics BLEU [72] and METEOR [6] and compare the predicted sentences against all ground truth sentences. BLEU and METEOR scores are computed based on the alignment between a given hypothesis sentence and a set of candidate reference sentences. BLEU only checks for exact matches of $n - grams$ in the predicted and groundtruth reference. Whereas METEOR computes the alignment by comparing exact token matches, stemmed tokens, paraphrase matches, as well as semantically similar matches using WordNet synonyms. Image description literature often use BLEU for evaluation, but a more recent study [27] has shown METEOR to be a better evaluation metric. However, 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 3.1. We also present qualitative examples in Figure 3.2 Samples of videos clips with the model's predictions can be found at `https://www.youtube.com/watch?v=IGaAoW8bA4c`. The code for this model is available in the caffe framework and can be viewed on github[1].

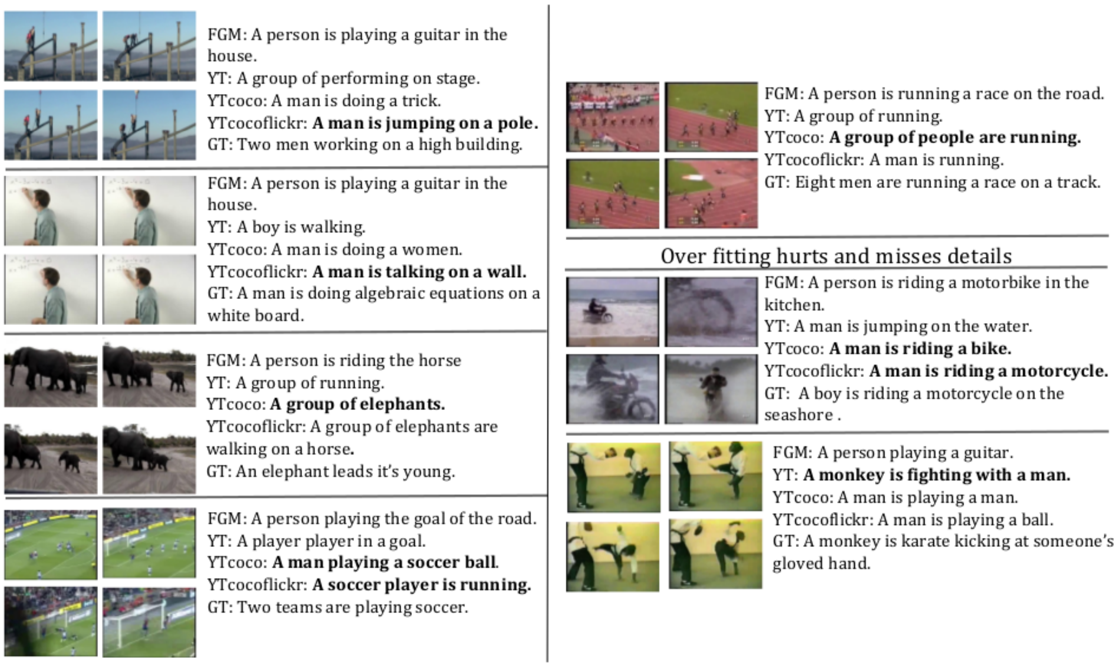
3.2.4 Discussion

From the results in Table 3.1, it is evident that our LSTM based approach significantly outperforms the previous state-of-art (FGM). We also observe that learning from the image description data improves the performance of the model in terms of both METEOR and BLEU. 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.

[1] `https://github.com/vsubhashini/caffe/tree/recurrent/examples/youtube`

Model	BLEU	METEOR
FGM [91]	13.68	23.90
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 3.1: Scores for BLEU at 4 (combined n-gram 1-4), and METEOR scores from automated evaluation metrics comparing the quality of the generation. All values are reported as percentage (%).


FGM: A person is playing a guitar in the house.
YT: A group of performing on stage.
YTcoco: A man is doing a trick.
YTcocoflickr: **A man is jumping on a pole.**
GT: Two men working on a high building.

FGM: A person is playing a guitar in the house.
YT: A boy is walking.
YTcoco: A man is doing a women.
YTcocoflickr: **A man is talking on a wall.**
GT: A man is doing algebraic equations on a white board.

FGM: A person is riding the horse
YT: A group of running.
YTcoco: **A group of elephants.**
YTcocoflickr: A group of elephants are walking on a horse.
GT: An elephant leads it's young.

FGM: A person playing the goal of the road.
YT: A player player in a goal.
YTcoco: **A man playing a soccer ball.**
YTcocoflickr: **A soccer player is running.**
GT: Two teams are playing soccer.

FGM: A person is running a race on the road.
YT: A group of running.
YTcoco: **A group of people are running.**
YTcocoflickr: A man is running.
GT: Eight men are running a race on a track.

Over fitting hurts and misses details

FGM: A person is riding a motorbike in the kitchen.
YT: A man is jumping on the water.
YTcoco: **A man is riding a bike.**
YTcocoflickr: **A man is riding a motorcycle.**
GT: A boy is riding a motorcycle on the seashore .

FGM: A person playing a guitar.
YT: **A monkey is fighting with a man.**
YTcoco: A man is playing a man.
YTcocoflickr: A man is playing a ball.
GT: A monkey is karate kicking at someone's gloved hand.

Figure 3.2: Examples to demonstrate effectiveness of transferring from the image description domain. YT refer to the LSTM-YT, YTcoco to the LSTM-YT$_{coco}$, and YTcocoflickr to the LSTM-YT$_{coco+flickr}$ models. GT is a random human description in the ground truth. Sentences in **bold** highlight the most accurate description for the video amongst the models. Bottom two examples on the right show how transfer can overfit. Thus, while base LSTM-YT model detects water and monkey, the LSTM-YT$_{coco}$ and LSTM-YT$_{cocoflickr}$ models fail to describe the event completely.

3.3 Sequence to Sequence – Video to Text

In this section, we propose a novel end-to-end sequence-to-sequence model to generate captions for videos. A major limitation of our model [95] in the previous section is that it fails to exploit any of the temporal information in the video, treating the video as a "bag of image frames" and simply mean-pooling the results from individual frames to generate a deep-network encoding of the video. To address this shortcoming we develop, S2VT [96],

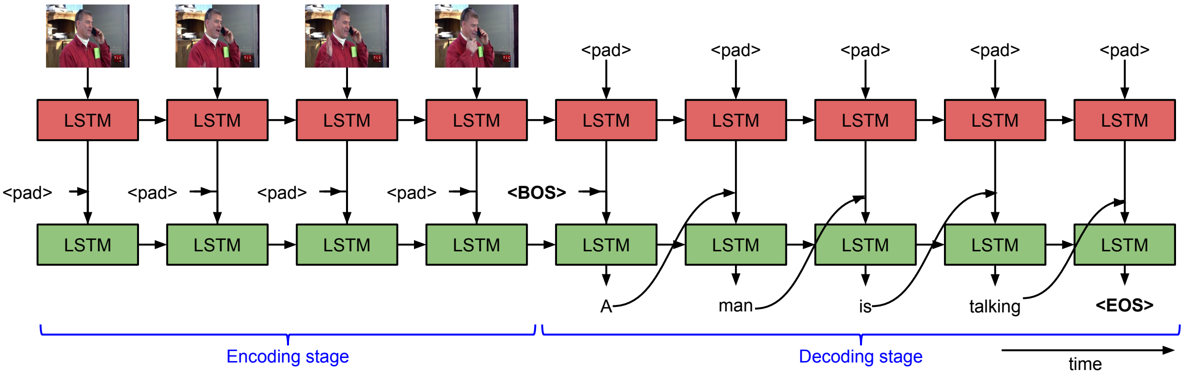


Figure 3.3: S2VT consists of a stack of two LSTMs that learn a representation of a sequence of frames in order to decode it into a sentence that describes the event in the video. The top LSTM layer (colored red) models visual feature inputs (from RGB or optical flow frames). The second LSTM layer (colored green) models language given the text input and the hidden representation of the video sequence. We use <BOS> to indicate begin-of-sentence and <EOS> for the end-of-sentence tag. Since we use the same LSTM layers for both encoding and decoding, zeros are used as a <pad> when there is no input at the time step.

a variant of our LSTM model that is sensitive to temporal structure and allows both input (sequence of frames) and output (sequence of words) of variable length. Figure 3.3 depicts our model. A stacked LSTM first encodes the frames one by one, taking as input the output of a Convolutional Neural Network (CNN) applied to each input frame's intensity values. It sequentially processes video frames, incrementally building up a hidden-layer semantic representation in the LSTM that effectively encodes the underlying activity. Once all frames are read, the model generates a sentence word by word. The encoding and decoding of the frame and word representations are learned jointly from a parallel corpus. To model the temporal aspects of activities typically shown in videos, we also compute the optical flow [10] between pairs of consecutive frames. The flow images are also passed through a CNN and provided as input to the LSTM. Flow CNN models have been shown to be beneficial for activity recognition [85, 25].

To our knowledge, this is the first approach to video description that uses a general sequence to sequence model. This allows our model to (a) handle a variable number of input frames, (b) learn and use the temporal structure of the video and (c) learn a language model to generate natural, grammatical sentences. Our model is learned jointly and end-to-end, incorporating both intensity and optical flow inputs, and does not require an explicit attention model. We demonstrate that S2VT achieves state-of-the-art performance on three diverse datasets, a standard YouTube corpus (MSVD) [13] and the M-VAD [93] and MPII Movie Description [79] datasets. We also make our implementation (based on the *Caffe* [42] deep learning framework) available on github[2].

3.3.1 LSTMs for Sequence-to-Sequence Video-to-Text

Our model uses a stack of two LSTMs with 1000 hidden units each. Figure 3.3 shows the LSTM stack unrolled over time. When two LSTMs are stacked together, as in our case,

[2] https://github.com/vsubhashini/caffe/tree/recurrent/examples/s2vt

the hidden representation (h_t) from the first LSTM layer (colored red) is provided as the input (x_t) to the second LSTM (colored green). The top LSTM layer in our architecture is used to model the visual frame sequence, and the next layer is used to model the output word sequence.

Training and Inference In the first several time steps, the top LSTM layer (colored red in Figure Figure 3.3) receives a sequence of frames and encodes them while the second LSTM layer receives the hidden representation (h_t) and concatenates it with the input padding words (zeros), which it then encodes. There is no loss during this stage when the LSTMs are encoding. After all the frames in the video clip are exhausted, the second LSTM layer is fed the beginning-of-sentence (<BOS>) tag, which prompts it to start decoding its current hidden representation to a sequence of words. While training in the decoding stage, the model maximizes for the log-likelihood of the predicted output sentence given the hidden representation of the visual frame sequence, and the previous words it has seen. For a model with parameters θ and output sequence $Y = (y_1, \ldots, y_m)$, this is formulated as:

$$\theta^* = \underset{\theta}{\operatorname{argmax}} \sum_{t=1}^{m} \log p(y_t | h_{n+t-1}, y_{t-1}; \theta) \tag{3.4}$$

This log-likelihood is optimized over the entire training dataset using stochastic gradient descent. The loss is computed only when the LSTM is learning to decode. Since this loss is propagated back in time, the LSTM learns to generate an appropriate hidden state representation (h_n) of the input sequence. The output (z_t) of the second LSTM layer is used to obtain the emitted word (y). We apply a softmax function to get the probability distribution over the words y' in the vocabulary V:

$$p(y|z_t) = \frac{\exp(W_y z_t)}{\sum_{y' \in V} \exp(W_{y'} z_t)} \tag{3.5}$$

We note that, during the decoding phase, the visual frame representation for the first LSTM layer is simply a vector of zeros that acts as padding input. We require an explicit end-of-sentence tag (<EOS>) to terminate each sentence since this enables the model to define a distribution over sequences of varying lengths. At test time, during each decoding step we choose the word y_t with the maximum probability after the softmax (from Equation Eq. (3.5)) until we obtain the <EOS> token.

3.3.2 Video and text representation

RGB frames. Similar to previous LSTM-based image captioning efforts [25, 99] and video-to-text approaches [97, 107], we apply a convolutional neural network (CNN) to input images and provide the output of the top layer as input to the LSTM unit. In this work, we report results using the output of the fc7 layer (after applying the ReLU non-linearity) on the Caffe Reference Net (a variant of AlexNet) and also the 16-layer VGG model [86]. We use CNNs that are pre-trained on the 1.2M image ILSVRC-2012 object classification subset of the ImageNet dataset [81] and made available publicly via the Caffe ModelZoo.[3] Each input video frame is scaled to 256x256, and is cropped down to a random 227x227 region. It is then processed by the CNN. We remove the original last fully-connected classification

[3] https://github.com/BVLC/caffe/wiki/Model-Zoo

layer and learn a new linear embedding of the features to a 500 dimensional space. The lower dimension features form the input (x_t) to the first LSTM layer. The weights of the embedding are learned in combination with the LSTM layers during training.

Optical Flow. In addition to CNN outputs from raw image (RGB) frames, we also incorporate optical flow measures as input sequences to our architecture. Others [70, 25] have shown that incorporating optical flow information to LSTMs improves activity classification. As many of our descriptions are activity centered, we explore this option for video description as well. We follow the approach in [25, 32] and first extract classical variational optical flow features [10]. We then create flow images [32], by centering x and y flow values around 128 and multiplying by a scalar such that flow values fall between 0 and 255. We also calculate the flow magnitude and add it as a third channel to the flow image. We then use a CNN [32] initialized with weights trained on the UCF101 video dataset to classify optical flow images into 101 activity classes. The fc6 layer activations of the CNN are embedded in a lower 500 dimensional space which is then given as input to the LSTM. The rest of the LSTM architecture remains unchanged for flow inputs.

Text input. The target output sequence of words are represented using one-hot vector encoding (1-of-N coding, where N is the size of the vocabulary). Similar to the treatment of frame features, we embed words to a lower 500 dimensional space by applying a linear transformation to the input data and learning it's parameters via back propagation. The embedded word vector concatenated with the output (h_t) of the first LSTM layer forms the input to the second LSTM layer (marked green in Figure 3.3). When considering the output of the LSTM we apply a softmax over the complete vocabulary as in Equation Eq. (3.5).

3.3.3 Evaluation

In the following we describe how we evaluate our approach. We first describe the datasets we use, then the evaluation protocol, and then the details of our models.

Datasets In addition to the Microsoft Video Description corpus (MSVD) [13] (Section 3.2.3), we also evaluate our approach on two large movie corpora, namely, the MPII Movie Description Corpus (MPII-MD) [79], and the Montreal Video Annotation Dataset (M-VAD) [93]. Statistics of each corpus is presented in Table 3.2.

MPII Movie Description Dataset (MPII-MD) MPII-MD [79] contains around 68,000 video clips extracted from 94 Hollywood movies. Each clip is accompanied with a single sentence description which is sourced from movie scripts and audio description (AD) data. The AD or Descriptive Video Service (DVS) track is an additional audio track that is added to the movies to describe explicit visual elements in a movie for the visually impaired. Although the movie snippets are manually aligned to the descriptions, the data is very challenging due to the high diversity of visual and textual content. Typically most snippets only have single reference sentence. We use the training/validation/test split provided by the authors and extract every fifth frame (videos are shorter than MSVD, averaging 94 frames).

Montreal Video Annotation Dataset (M-VAD) The M-VAD movie description corpus [93] is another recent collection of about 49,000 short video clips from 92 movies. It is similar to MPII-MD, but only contains AD data and only provides automatic alignment. We use the same setup as for MPII-MD.

	MSVD	MPII-MD	MVAD
#-sentences	80,827	68,375	56,634
#-tokens	567,874	679,157	568,408
vocab	12,594	21,700	18,092
#-videos	1,970	68,337	46,009
avg. length	10.2s	3.9s	6.2s
#-sents per video	≈41	1	1-2

Table 3.2: Corpus Statistics. While the number of tokens (words+punctuation) in all datasets are comparable, but MSVD has fewer videos with more sentences per video and both the movie corpora (MPII-MD and MVAD) have a large number of clips with a single description per clip. Thus, the number of video, sentence pairs in all three datasets are comparable.

3.3.4 Evaluation Metrics

Quantitative evaluation of the models are performed using the METEOR [6] metric which was originally proposed to evaluate machine translation results. METEOR is the most appropriate metric for our data since the movie description corpora have just 1 ground truth reference each. [94] showed that METEOR is always better than other MT metrics such as BLEU when the number of references is small. We use the code[4] released with the Microsoft COCO Evaluation Server [14] to obtain the scores for all our models reported in this section.

3.3.5 Related approaches

We compare our sequence to sequence models against the factor graph model (FGM) in [91] (Section 2.3), the mean-pooled models (Mean-Pool) in [97] and the Soft-Attention models of [107].

The **Soft-Attention** model in [107] is a contemporaneous LSTM based approach. It is a combination of weighted attention over a fixed set of video frames with input features from GoogleNet and a 3D-convnet trained on Histogram of Gradients (HoG), Histogram of Flow (HoF) and Motion Boundary Histograms (MBH) features from an activity classification model.

3.3.6 Discussion: MSVD dataset

Table 3.3 shows the results on the MSVD dataset. The upper part shows results of related approaches and the lower part different variants of our S2VT approach.

Our basic S2VT AlexNet model on RGB video frames (line 8 in Table 3.3) achieves 27.9% METEOR and improves over the basic mean pooled model proposed by [97] (line 2, 26.9%) as well as VGG mean pooled model (line 3, 27.7%). This suggests that our sequence to sequence model even with the less powerful AlexNet features is able to encode video

[4] https://github.com/tylin/coco-caption

Model	METEOR	
FGM [91]	23.9	(1)
Mean pool		
- AlexNet [97]	26.9	(2)
- VGG	27.7	(3)
- AlexNet COCO pre-trained [97]	29.1	(4)
- GNet [107]	28.7	(5)
Soft-attention		
- GoogleNet [107]	29.0	(6)
- GoogleNet + 3D-CNN [107]	29.6	(7)
S2VT (ours)		
- Flow (AlexNet)	24.3	(8)
- RGB (AlexNet)	27.9	(9)
- RGB (VGG) random frame order	27.9	(10)
- RGB (VGG)	29.2	(11)
- RGB (VGG) + Flow (AlexNet)	29.8	(12)

Table 3.3: MSVD dataset (METEOR in %, higher is better).

frames well. When the model is trained with the input frame sequence randomly ordered (line 10 in Table 3.3) the score is considerably lower and comparable to the mean pooled approach (line 3) indicating that the model does exploit temporal structure when available.

Our S2VT model which uses flow images (line 9) achieves only 24.3% METEOR but improves the performance of our VGG model from 29.2%(line 10) to 29.8% (line 12), when combined. Our ensemble using both RGB and Flow achieves a score comparable and slightly better than the best model proposed in [107], Soft-attention with GoogleNet + 3D-CNN (line 7). The edge that our model has is only modest, this is likely due to the much stronger 3D-CNN features (as the difference to GoogleNet alone, line 6, suggest). Thus, the closest comparison between the Soft Attention Model [107] and our S2VT is arguably ours with VGG (line 10) vs. their GoogleNet only model (line 6).

Figure 3.4 shows descriptions generated by our model on some of the videos in the MSVD YouTube video dataset. To compare the originality in generation, we compute the Levenshtein distance of the predicted sentences with those in the training set. From Table 3.4, for the MSVD corpus, only 42.9% of the predictions are identical to some training sentence, and another 38.3% can be obtained by inserting, deleting or substituting one word from some sentence in the training corpus.

3.3.7 Discussion: Movie Corpora

For the more challenging MPII-MD and M-VAD datasets we use our single best model, namely S2VT trained on RGB frames and VGG. To avoid over-fitting on the movie corpora we employ drop-out which has proved to be beneficial on these datasets [78]. We found it was best to use dropout at the inputs and outputs of both LSTM layers. Further, we used ADAM [46] for optimization with a first momentum coefficient of 0.9 and a second momentum coefficient of 0.999. For MPII-MD, reported in Table 3.5, we improve over the

Edit-Distance	$k = 0$	$k <= 1$	$k <= 2$	$k <= 3$
MSVD	42.9	81.2	93.6	96.6
MPII-MD	17.7	43.1	51.4	60.1
MVAD	03.0	38.9	43.9	60.1

Table 3.4: Percentage of generated sentences which match a sentence of the training set with an edit (Levenshtein) distance of less than 4. All values reported in percentage (%).

Approach (MPII-MD)	METEOR
SMT (best variant) [79]	5.6
Visual-Labels [78]	7.0
Mean pool (VGG)	6.7
S2VT: RGB (VGG), ours	7.1

Table 3.5: MPII-MD dataset (METEOR in %, higher is better).

Approach	METEOR
Visual-Labels [78]	6.3
Temporal attention [107]	5.7
Mean pool (VGG)	6.1
S2VT: RGB (VGG), ours	6.7

Table 3.6: M-VAD dataset (METEOR in %, higher is better).

SMT approach from [79] from 5.6% to 7.1% METEOR and over Mean pooling [97] by 0.4%. Our performance is similar to Visual-Labels [78], a contemporaneous LSTM-based approach which uses no temporal encoding, but more diverse visual features, namely object detectors, as well as activity and scene classifiers.

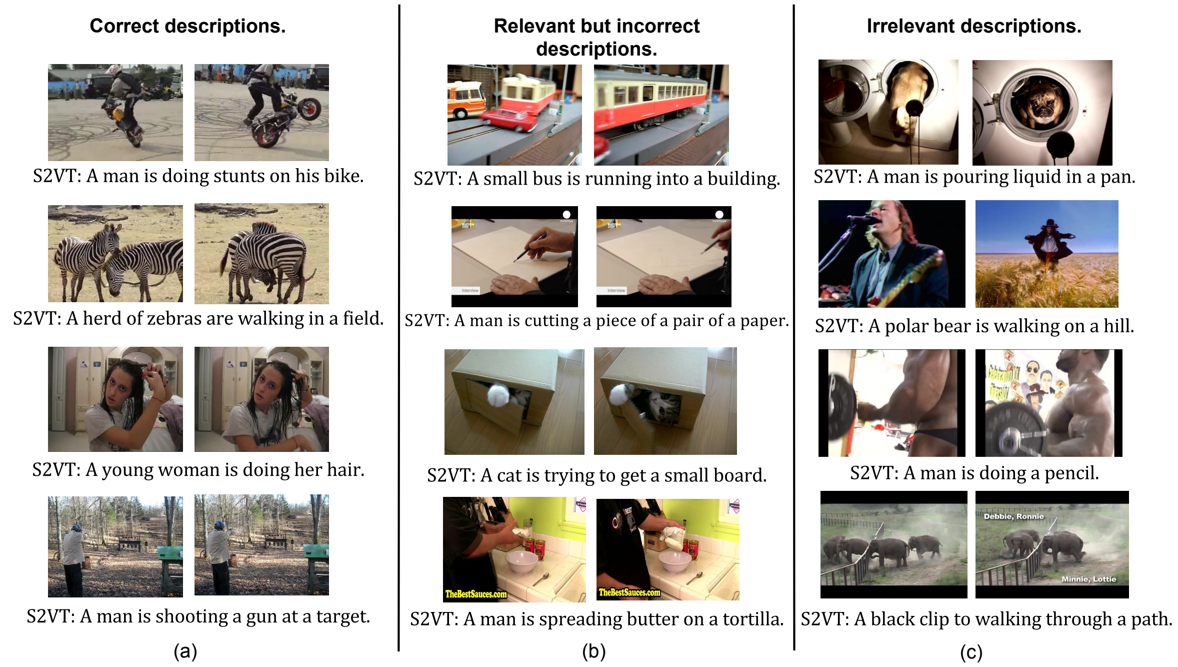
On M-VAD we achieve 6.7% METEOR which significantly outperforms the temporal attention model [107] (5.7%) and Mean pooling (6.1%). On this dataset we also outperform Visual-Labels [78] (6.3%).For the more challenging MPII-MD and M-VAD datasets we use our single best model, namely S2VT trained on RGB frames and VGG. To avoid over-fitting on the movie corpora we employ drop-out which has proved to be beneficial on these datasets [78]. We found it was best to use dropout at the inputs and outputs of both LSTM layers. Further, we used ADAM [46] for optimization with a first momentum coefficient of 0.9 and a second momentum coefficient of 0.999.

In Figure 3.5 we present descriptions generated by our model on some sample clips from the M-VAD dataset. More example video clips, generated sentences, and data are available on the authors' webpages[5].

3.4 Summary of Completed Work

This section proposed two deep models for video description that used convolutional and recurrent networks to translate from pixels to sentences. In our first model we presented techniques to take advantage of large image description datasets, and transfer knowledge from the image captioning task to the video captioning task. We then developed a sequence to sequence video description model, where frames are first read sequentially and then words are generated sequentially. This allows us to handle variable-length input and output while simultaneously modeling the temporal structure. Our model out-performs all

[5] http://vsubhashini.github.io/s2vt.html



Correct descriptions.

S2VT: A man is doing stunts on his bike.

S2VT: A herd of zebras are walking in a field.

S2VT: A young woman is doing her hair.

S2VT: A man is shooting a gun at a target.

(a)

Relevant but incorrect descriptions.

S2VT: A small bus is running into a building.

S2VT: A man is cutting a piece of a pair of a paper.

S2VT: A cat is trying to get a small board.

S2VT: A man is spreading butter on a tortilla.

(b)

Irrelevant descriptions.

S2VT: A man is pouring liquid in a pan.

S2VT: A polar bear is walking on a hill.

S2VT: A man is doing a pencil.

S2VT: A black clip to walking through a path.

(c)

Figure 3.4: Qualitative results on MSVD YouTube dataset from our S2VT model (RGB on VGG net). (a) Correct descriptions involving different objects and actions for several videos. (b) Relevant but incorrect descriptions. (c) Descriptions that are irrelevant to the event in the video.

Temporal Attention (GNet+3D-conv$_{att}$): (1) At night , SOMEONE and SOMEONE step into the parking lot.
(2) Now the van drives away.
(3) They drive away.
(4) They drive off.
(5) They drive off.
(6) At the end of the street , SOMEONE sits with his eyes closed.

S2VT (Ours): (1) Now, the van pulls out a window and a tall brick facade of tall trees . a figure stands at a curb.
(2) Someone drives off the passenger car and drives off.
(3) They drive off the street.
(4) They drive off a suburban road and parks in a dirt neighborhood.
(5) They drive off a suburban road and parks on a street.
(6) Someone sits in the doorway and stares at her with a furrowed brow.

DVS: (1) Now , at night , our view glides over a highway its lanes glittering from the lights of traffic below.
(2) Someone's suv cruises down a quiet road.
(3) Then turn into a parking lot .
(4) A neon palm tree glows on a sign that reads oasis motel.
(5) Someone parks his suv in front of some rooms.
(6) He climbs out with his briefcase , sweeping his cautious gaze around the area.

Figure 3.5: M-VAD Movie corpus: Representative frame from 6 contiguous clips from the movie "Big Mommas: Like Father, Like Son". From left: Temporal Attention (GoogleNet+3D-CNN) [107], S2VT (in blue) trained on the M-VAD dataset, and DVS: ground truth.

previous works on Youtube clips from the MSVD dataset, and the DVS movie description datasets.

Chapter 4

Proposed Work

The goal of this proposal is to develop models that achieve a deeper integration of linguistic and visual semantics to automatically describe a wide range of ordinary videos in natural language. Automatic video description techniques should be capable of identifying salient events worth describing and should be able to appropriately describe a wide variety of video content with a large number of diverse actions, objects, scenes and other properties. Deep neural network video description models take a significant step in this direction by learning to describe salient objects directly from video and caption pairs. Although, recent deep recurrent approaches to video description show promising results, they are still limited in many ways. Current deep neural video-captioning models, (i) rely largely on linguistic knowledge in paired image/video-sentence corpora, (ii) fail to generalize for multiple event sequences in longer videos, (iii) fail to track and capture interactions between a variety of objects, and (iv) lack the ability to generate detailed and accurate descriptions for natural everyday scenes, particularly in movies. In the next few sections, I propose approaches to address these shortcomings.

First, as immediate extensions, I propose a variety of methods to integrate prior linguistic knowledge into existing video captioning networks. Specifically we take advantage of large monolingual text corpora, and propose methods to incorporate knowledge from different kinds of neural net language models to aid video description. These models can be further extended to attend to specific objects and actions, using attention methods [68, 103] to track and capture interactions between objects and generate more accurate descriptions. Moving beyond single sentence descriptions of short video clips, as a long-term goal, I propose models to address description of longer multi-activity videos. These models will learn to simultaneously segment a long video into coherent scenes and generate a description for each event at it's completion. In the final section, as bonus work, I propose to investigate schemes to enhance DVS descriptions for movies. In particular, the focus of the model will be on leveraging movie script and subtitle information to include names of characters during the generation process. This additional information can also be used to enhance the overall quality of the generated DVS descriptions.

4.1 Using Statistical Linguistic Knowledge to Aid Video Description

Real-world videos depict interaction of actors with a range of objects, scenes and actions, and recent neural network-based architectures have shown promising results on recognizing and describing these activities "in-the-wild". A significant factor contributing to the success of neural network architectures for image description is the availability of large amounts of paired image-sentence corpora. In the case of videos however, there is a lack of high-quality paired video-sentence corpora. In contrast, monolingual text corpora are widely available. Despite the lack of visual grounding, plain text corpora exhibit rich linguistic structure that can aid video to text translation. Most work in statistical machine translation utilizes both a language model trained on a large corpus of monolingual data for the target language as well as a translation model trained on more limited parallel bilingual data. In this work, we explore ways to incorporate knowledge from language corpora to improve natural language descriptions for videos.

We investigate three approaches to integrate linguistic information into an LSTM-based sequence to sequence video to text system [96]. Our first approach is to incorporate distributional vector representation of words in addition to (or as a substitute) to word embeddings learned internally by the video description network. Our second approach is to pre-train the video description model on large corpora of raw NL text to capture general linguistic regularities. Our third approach is to integrate a trained RNN language model employing early and mid-level fusion techniques to improve video description. [37] developed an LSTM model for machine translation that incorporates a monolingual language model (LM) for the target language showing improved results. We utilize a similar approach to integrate an LM so as to include the representation of words learned by the LM as an input feature to the video to text decoder. We can additionally, rescore the output words generated by the caption model based on the trained language model.

4.1.1 Distributional Vector Representations

A drawback of our previous deep video captioning network is that, they represent words using a 1-of-N (one hot) encoding. This naive representation reduces the vocabulary of the captioning model significantly (by an order of magnitude) compared to a regular language model. In order to encode a wider variety of words, it would be advantageous to use embeddings learned from a distributional vector space. Approaches such as Word2Vec [67] and Glove [73] use large text corpora to learn vector-space representations of words that capture fine-grained semantic and syntactic regularities. One approach to learn a similar representation in the caption model is to embed (one-hot) words into a lower dimensional space by applying a linear transformation, and learning its parameters via backpropagation. This can be represented as $f : \mathcal{V}_{1hot} \rightarrow \mathcal{V}_{lstm}$, where $\mathcal{V}$ denotes the vocabulary[1]. However, this embedding is learned only from the text in the parallel training data which is very limited. We propose ways to enhance this embedding further using external distributional vectors:

Initialization A simple technique to incorporate a trained distribution vector representations is to use the weights from the external embedding, Word2Vec or Glove, to initialize the caption model's word embedding (that maps from one-hot representation to a lower dimensional vector).

Additional Embedding Another technique that has been used in the process of learning sentence vectors [113], is to learn an additional embedding mapping from the external

[1] subscripts $1hot$, $lstm$, and $w2v$ refer to the representations of the 1-hot vectors, the LSTM word embedding, and the distributional vector space respectively

distributional vectors to the lower dimensional vector, i.e $f : \mathcal{V}_{w2v} \rightarrow \mathcal{V}_{lstm}$. This has the added advantage that any word in $\mathcal{V}_{w2v}$ can now be mapped to a vector when encoding the input.

Concatenate as a feature Yet another technique, is to simply concatenate words from the $\mathcal{V}_{w2v}$ external embedding, Word2Vec or Glove, as an additional input feature along with the one-hot vectors.

4.1.2 LSTM Language Pre-Training

In our second approach, we propose pre-training the language layers of the LSTM network to learn an LM using web-scale text corpora. [88] showed LSTMs to be very effective language models. Additionally, [37] have also used LSTM based language models for machine translation. Since the LSTM model learns to estimate the probability of an output sequence given an input sequence (Section), we can learn a language model, by training the LSTM layer to predict the next word given the previous words. We propose to use both web-scale text corpora and text from parallel corpora to train the LM. As in the original captioning network, the embedding and the LSTM parameters for just the language layers can be learned via backpropagation using stochastic gradient descent. The weights from this language model can then be used to initialize the embedding and weights of the LSTM layers of the complete captioning model. The network can finally be tuned on the video description datasets. The LM pre-training should help the caption model create a better representation of the text and enable it to generate more diverse descriptions by capturing regularities inherent in natural language.

4.1.3 Language Fusion

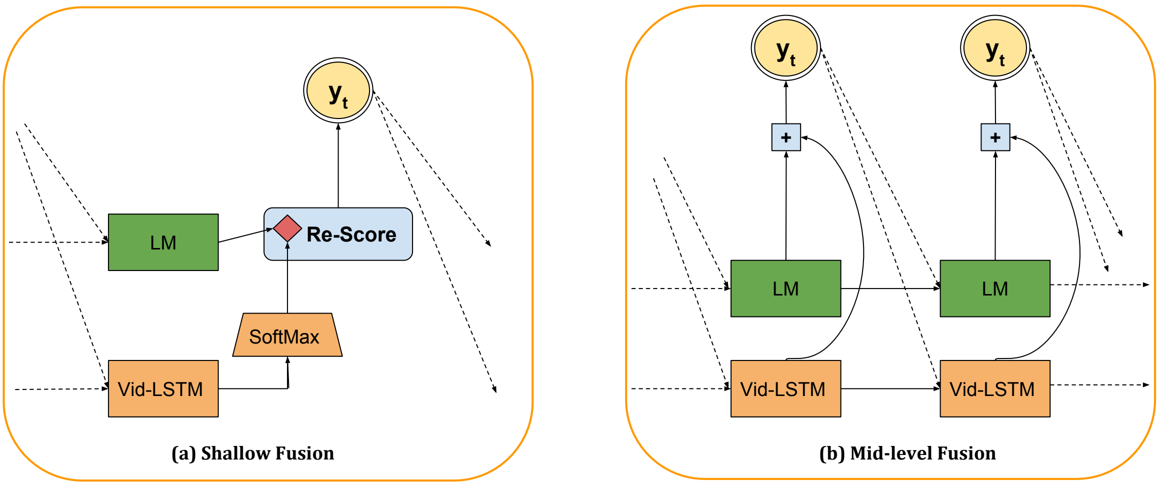
In addition to the language model learned during captioning, we will investigate whether an external LM can further enhance fluency during final caption generation stage. While paired image-text corpora can be scarce, particularly for new concepts, monolingual text corpora is widely available. Despite the lack of visual grounding, plain text corpora exhibit rich linguistic structure, and we can take advantage of this for enhancing caption generation. We propose two ways (i) a shallow fusion and (ii) a mid-level fusion technique to integrate a pre-trained LM to aid sentence generation. These are illustrated in Figure 4.1

Shallow LM Fusion

Our fusion approach is similar to how machine translation models incorporate a trained language model during decoding [37]. At each step of sentence generation, the video caption model proposes a distribution over the vocabulary word. We can use the language model to re-score the final output by considering the weighted average of the sum of scores proposed by the LM as well as the video-description model (VM). More specifically, if y_t denotes the output at time step t, and if p_{VM} and p_{LM} denote the proposal distributions of the video captioning model, and the language models respectively, then for all words $y' \in V$ in the vocabulary we can recompute the score of each new word, $p(y_t = y')$ as:

$$\alpha \cdot p_{VM}(y_t = y') + (1 - \alpha) \cdot p_{LM}(y_t = y') \tag{4.1}$$

The hyper-parameter α can be tuned on the validation set.



(a) Shallow Fusion

(b) Mid-level Fusion

Figure 4.1: Illustration of our shallow and mid-level fusion approaches to integrate an independently trained language model to aid captioning.

Mid-level LM Fusion

In the mid-level fusion approach we integrate the LM a step deeper in the generation process by concatenating the hidden state of the language model LSTM (h_t^{LM}) with the hidden state of the video description model (h_t^{VM}) and use the combined vector to predict the output word. In this process, the video captioning model is fine-tuned to use hidden states from both models to predict the next word. Thus, in this case, probability of the predicted word at time step t is

$$p(y_t|y_{<t}, \mathrm{x}) \propto \exp(y_t^\intercal(\mathrm{W}\mathrm{f}(h_t^{VM}, h_t^{LM}) + b)) \tag{4.2}$$

where W is the weight matrix and b the biases. In this method we need to only tune those weights used to parameterize the output prediction. We should avoid tuning the LM or the video captioning network to prevent overwriting already learned weights.

4.2 Using Attention to Generate Descriptions

Recent deep visual captioning models learn to describe salient objects in images and videos directly from captions and input visual features [98, 25, 29]. However, steering a model to the more important information and learning to attend to different aspects of the input has been shown to further improve caption generation [103]. Inspired by the recent success of attention models in sequence to sequence frameworks for machine translation [5], object recognition [4, 68], image captioning [103] and speech recognition [11] we propose to investigate models that can attend to different objects/actors at different times while generating captions of events as they unfold in the video. Our approach would focus on incorporating attention mechanisms in combination with our LSTM based segmentation and description model. Specifically, we plan to extract features from a lower convolution layer for each frame over different spatial regions (pertaining to objects/actors). We can then define an attention mechanism to generate weights for different spatial locations and frames. This allows the decoder to selectively focus on different locations/objects at different times weighting a subset of all the feature vectors. The network will learn to look at different locations depending on the sequence of words that it has already generated. Such models should be able to attend to salient aspects of the video while generating a description.

4.3 Multi-sentential Descriptions of Longer Videos

Generating multi-sentential descriptions of longer videos will be longer-term focus of my proposed research. A major limitation of existing works is that they only generate a single-sentence description of short (6-20 sec) video clips. We propose to extend our LSTM approach to produce multi-sentential descriptions of longer multi-activity videos, initially training and testing on DVS-annotated videos. We plan to develop an LSTM-based system to simultaneously segment a video-stream into single-activity clips and produce a sentential description of each individual clip. Using the segmented data in our DVS corpus as supervision, we will train the LSTM to also detect activity boundaries in video and signal when to end a segment, translate the resulting hidden-state representation of the current clip into an NL sentence, and then reset the state to begin encoding the next activity.

4.3.1 Using Text-Mined Scripts to Improve the Recognition of Event Sequences

Current video description models tend to generate repetitive descriptions for consecutive shots in a video.We propose to use text-mined "scripts" to improve the recognition of activity sequences in videos. *Scripts* encode knowledge of stereotypical events including information about their ordered sequence of sub-events and their arguments [82]. The classic example is the "restaurant script," which encodes knowledge about what normally happens when dining out: A patron enters a restaurant, a hostess shows them to a table, the waiter brings them a menu, the patron orders food, and so forth. Scripts can be used to improve text understanding by supporting inference of implied information as well as resolution of anaphora and lexical and syntactic ambiguities [83].

Recently, [74, 75] have developed improved approaches to script learning that employs a richer model of events. Further [75] also uses an LSTM network to model scripts. We propose to develop methods for using such learned scripts to aid the interpretation of activity sequences in videos. Our approach will use the learned script model to estimate the prior probability of a sequence of activities and use it to bias the visual recognition and segmentation of action sequences. For example, in a cooking video, the sequence of actions "open, pour, mix" is a priori more likely than "mix, pour, open" and therefore is a preferred interpretation. We plan to incorporate such a script model into our proposed method for joint event segmentation and description.

Adapting Script Models to DVS data on event sequence One approach would be to use existing LSTM based script models, but include visual features as an additional input to predict the next event in the sequence. In this case, our models would be trained on consecutive scenes from pre-segmented clips, along with the input representations for each of the events. Given the next clip, the model should learn to predict the next event in the sequence conditioned on the previous event sequences and the current clip. Additionally, we can incorporate another layer that generates a full sentence description of the event from the short sequence of actions representing the script event.

4.3.2 Temporal Video Segmentation

Temporal video segmentation typically refers to sub-dividing a video into spatio-temporal volumes. Oftentimes, it is at either shot boundaries, where the camera view changes or at scene boundaries where the background scene itself changes. In movies, video editing and interleaving can also create a natural temporal boundary. In our work we are primarily interested in temporal video segmentation based on the content and event in the video. Our approach needs to be capable of not only identifying high level scene boundaries, but also *event boundaries* (or *changepoints*), which could be subtle boundaries between the completion of a coherent event sequence and the beginning of another. E.g. in a cooking video this could be the point where the cook finishes pouring some sauce into the dish and starts mixing it.

There have been some works that have investigated temporal video segmentation to identify events [61], however they are applied in the case of ego-centric videos to identify when the wearer is static, in transit, or moving just their head. Most video segmentation techniques focus mainly on difference between consecutive frame pixel values to identify boundaries. They rely on image descriptors such as color histograms and local motion features and bag-of-features descriptors. Our approach would focus more on the event and hence we will exploit deep features which can aid in object, scene, and action recognition to identify coherent event boundaries. Additionally, we also want to simultaneously describe each event as it completes.

4.3.3 Bi-directional LSTM for video segmentation

Conventional LSTMs have a deficiency in the sense that they are only able to make use of previous context. For the task of video segmentation however, it would be optimal to make use of the future context as well. Bidirectional RNNs (BRNNs) [84] achieve this by processing the data in both directions using two separate hidden layers, one each for the previous context and the future context, which are both then fed forwards to the same output layer. In the case of bi-directional LSTMs, it consists of two LSTMs, where one of the LSTMs first processes the reverse sequence. Then, the other (forward) LSTM processes the sequence in-order, and the hidden state of the forward ($\overrightarrow{h}$) and reverse LSTMs ($\overleftarrow{h}$) are concatenated to generate the output at each time-step. If x_t and y_t represent the input and output at each time step t, then this can be represented as

$$\overrightarrow{h} = \mathcal{H}\left(W_{x\overrightarrow{h}} x_t + W_{\overrightarrow{h}\overrightarrow{h}} \overrightarrow{h}_{t-1} + b_{\overrightarrow{h}}\right) \tag{4.3}$$

$$\overleftarrow{h} = \mathcal{H}\left(W_{x\overleftarrow{h}} x_t + W_{\overleftarrow{h}\overleftarrow{h}} \overleftarrow{h}_{t-1} + b_{\overleftarrow{h}}\right) \tag{4.4}$$

$$y_t = W_{\overrightarrow{h}y} \overrightarrow{h}_t + W_{\overleftarrow{h}y} \overleftarrow{h}_t + b_y \tag{4.5}$$

where W are the weight matrices, b the bias terms and $\mathcal{H}$ is the composite function of the LSTM unit (Section 3.1 Equations (2.3–2.7)). Deep bi-directional LSTMs have been used successfully for hand-writing recognition [60] to predict characters and in speech recognition [33] to identify phonemes with very low error rates. This proposal will explore both conventional LSTMs and bi-directional LSTMs for video segmentation and captioning.

We propose to use a hierarchy of LSTM layers to address the problem of jointly segmenting and captioning longer videos. Our key idea, is to train one LSTM layer (conventional, or bi-directional, or even pyramidal [11]) to focus exclusively on segment/shot boundaries i.e. it will determine whether the current frame belongs to the previous shot or if it is a new shot. Then, the next layer will predict a value to explicitly reset the LSTM units (in the layer below) to indicate the beginning of a new sub-sequence. We will use an additional (output) LSTM layer, that will take as input the last state (before resetting) from the previous LSTM layer, in order to generate the output sentence (or script event). This architecture can be thought of as skip-lists, where each higher layer receives inputs from the layers below and decides whether or not it should output something at the current time step.

4.3.4 Combining features from unsupervised segmentation approaches

A strong baseline to compare against, would be to have different models for temporal segmentation and sentence generation. There have been a few unsupervised approaches to temporal video segmentation [87, 76, 112]. The underlying principle in these approaches is to cluster similar shots into coherent scene segments using color based features. [76, 112] are of particular interest since they apply their technique to segment TV shows and movies. Shot boundaries are easily detected using color histogram differences between consecutive frames, hence both works assume the video is already sub-divided into shots (based on camera view). They then consider key frames from these shots to extract object descriptors and color histogram features to group shots containing similar, objects, actors, and background scenes into a single scene segment. Generating descriptions for videos segmented using such unsupervised techniques can provide a strong baseline. Additionally, we can concatenate color histogram features and other intensity based features used by the unsupervised methods along with our deep features to assist our LSTM based video segmentation models.

4.3.5 Dataset for evaluation

We will evaluate the ability of the system to properly segment and produce NL descriptions for each segment by using BLEU and METEOR metrics to score the descriptions for novel DVS-annotated videos (Section 3.3.3). Additionally, we will also compare our temporal segmentation approach against the small set of movies and TV shows in [76]. We also plan to test the resulting system on full YouTube videos, using crowd-sourcing to manually annotate a small corpus of YouTube videos for training and testing, and also using crowd-sourcing to collect human evaluations of the automatically-generated multi-sentential descriptions.

4.4 Exploiting Movie Scripts and Sub-titles to Enhance DVS Descriptions for Movies

As bonus work, we propose to go beyond generating simplistic sentence descriptions for movies by developing effective methods for identifying characters in order to generate more precise descriptions including character names and associating actions to characters. Current movie description models [96, 107] are trained on sentences where character names are replaced with the generic noun “Someone”. While this is based on the premise that movies in the test set are never seen before and hence characters in it are unknown; in practice however, we have access to additional sources of information such as movie scripts and subtitles that can help in learning and recognizing characters.

It's important to note that by itself, neither the script nor the subtitles contain the required information to label the identity of the people in the video. The subtitles record what is said, but not by whom, whereas the script records who says what, but lacks timing information. Movie scripts typically include names of all characters and most movies loosely follow the sequence of events in the original script. Both the scripts and the subtitles together can be used to estimate the presence of a character on the video screen [28, 2]. Moreover, movie scripts are readily available and subtitles can also be easily obtained using automatic speech recognition. In addition, we can also obtain a few annotations of the characters by clustering similar faces and actively requesting for annotations on some examples. Then we can use techniques from [18, 17] to learn characters from ambiguously or partially labeled images.

4.4.1 Identifying Character Screen Presence from Movie Scripts

There is a body of prior work on identifying characters in video streams, e.g.,[28, 2] that uses subtitles and scripts to automatically assign character names to faces in the video frames. However, these works only recognize the presence of a character in the frame and do not identify the sequence of actions/events or generate their descriptions. Another closely related set of works look at aligning text from the web or books to appropriate positions in videos [63, 113]. Our proposed work differs from these since we are not looking to directly align existing text, but instead we wish to compose information in these texts (character names, actions) to generate a description of the event on the screen. Our proposal is to combine the subtitle and movie script information, to first identify the time intervals at which a character is present on screen. Then, given the clips from the movie, using the time stamp, each shot can be tagged with the characters that are likely to be present. Then we can use multiple instance learning (MIL) and other methods to learn character identities from ambiguously and partially labeled images [18, 17]. This can be used to generate more accurate DVS descriptions by including names of characters even on new test videos. Additionally, scripts and subtitle dialogues can be used to improve text understanding by supporting inference of implied information as well as co-reference resolution.

Datasets We plan to utilize some of the videos from the MPII-MD movie dataset described in Section 4.3.5. The DVS dataset contains aligned traditional Hollywood scripts for 40 of the 90 movies. They contain scene captions, dialogs and scene descriptions. They are temporally pre-aligned as described in [79] and a detailed alignment is performed manually. We plan to use data from the 40 movies to train our model, and evaluate it's ability to incorporate character information in the remaining videos.

4.4.2 Detection models for characters based on examples

Another technique would be to employ semi-supervised approaches to identify characters [17] in movies by taking advantage of face detection algorithms. An initial approach would be to run a frontal face detector on frames from the clips. We could then use a simple clustering algorithm to cluster similar faces, or employ a face tracker such as the Kanade-Lucas-Tomasi tracker [92]. Clustering and face tracking can establish correspondence between pairs of faces within the same shot. Additionally, face-tracking is more robust as it can also establish matches between faces where the frontal face detector may have missed detection due to pose variation or expression change. Then, based on the example image for each character, we can learn a classifier to classify images to any of the characters (or identify none-of-the-above). This can then be integrated with our existing LSTM based description models. The final network, will include features from a regular object classifier as well as the character classifier, and needs to be tuned on a few sentences containing character names to generate appropriate sentences.

Alternately, we can employ simple language transfer techniques to completely avoid annotating some sentences with character names on the test movies. The primary reason for fine-tuning the caption model on sentences with character names is to update the language model within the LSTM network enabling it to generate coherent sentences incorporating the names. However, if we can transfer knowledge from words used in similar contexts, such as man, woman, person, to the new words (names of characters), then the language model will be able to construct sentences describing the event including the character names. Adopting the model in [64], it is possible to make a slight modification to the LSTM decoder to include an additional layer that combines the visual features and language features just before prediction. This will enable the LSTM model to easily integrate a character classifier and also allow the transfer of weights from known words to new words facilitating the generation of coherent sentences incorporating names of characters.

Chapter 5

Conclusion

Generating natural language descriptions for events in videos enables several applications. The last year has seen a dramatic interest description of static images and growing interest in video description. This proposal focuses on generating natural language descriptions that capture sequences of activities depicted in diverse video corpora, where limited prior work exists. The major obstacles to scale video description are limited training data, wide diversity of visual and language content, and lack of rich and robust representations. As a step in addressing these challenges, this proposal presents the first fully deep model to generate descriptions of events depicted in videos. Our model is capable of learning salient entities worth describing directly from video and sentence pairs. It treats the video domain as another "language" and takes a machine translation approach to translate videos to text. This proposal also highlights several directions to significantly extend work in this area. Specifically, we propose strategies to generate more diverse and accurate descriptions by integrating prior linguistic knowledge and attention methods to focus on object interactions. We also propose schemes to process longer multi-activity videos by learning to jointly segment and describe coherent event sequences.

Acknowledgements

I thank Marcus Rohrbach for help with some of the experiments. I also thank Jeff Donahue and contributors of Caffe for building the framework and tools used to conduct my experiments. This work is funded by the ONR ATL Grant N00014-11-1-010. My experiments were run on clusters funded by the Berkeley Vision and Learning Center and on the UT Mastodon Cluster, provided by NSF Grant EIA-0303609.

Bibliography

[1] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. *arXiv preprint arXiv:1505.00468*, 2015. 4

[2] N. E. Apostoloff and A. Zisserman. Who are you? – real-time person identification. In *BMVC*, 2007. 28

[3] H. Aradhye, G. Toderici, and J. Yagnik. Video2text: Learning to annotate video content. In *IEEE International Conference on Data Mining Workshops (ICDMW)*, 2009. 4

[4] Jimmy Ba, Volodymyr Mnih, and Koray Kavukcuoglu. Multiple object recognition with visual attention. *arXiv preprint arXiv:1412.7755*, 2014. 24

[5] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014. 3, 24

[6] Satanjeev Banerjee and Alon Lavie. 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*, 2005. 12, 17

[7] A. Barbu, A. Bridge, Z. Burchill, D. Coroian, S. Dickinson, S. Fidler, A. Michaux, S. Mussman, S. Narayanaswamy, D. Salvi, L. Schmidt, J. Shangguan, J.M. Siskind, J. Waggoner, S. Wang, J. Wei, Y. Yin, and Z. Zhan. Video in sentences out. In *UAI*, 2012. 2, 4

[8] Kobus Barnard, Pinar Duygulu, David Forsyth, Nando de Freitas, David M. Blei, and Michael I. Jordan. Matching words and pictures. *JMLR*, 3:1107–1135, 2003. 4

[9] Tamara L. Berg, Alexander C. Berg, Jaety Edwards, Michael Maire, Ryan White, Yee Whye Teh, Erik G. Learned-Miller, and David A. Forsyth. Names and faces in the news. In *CVPR*, pages 848–854, 2004. 4

[10] Thomas Brox, Andrés Bruhn, Nils Papenberg, and Joachim Weickert. High accuracy optical flow estimation based on a theory for warping. In *ECCV*, pages 25–36, 2004. 14, 16

[11] William Chan, Navdeep Jaitly, Quoc V Le, and Oriol Vinyals. Listen, attend and spell. *arXiv preprint arXiv:1508.01211*, 2015. 24, 27

[12] C.C. Chang and C.J. Lin. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27, 2011. 5

[13] David L. Chen and William B. Dolan. Collecting highly parallel data for paraphrase evaluation. In *ACL*, Portland, Oregon, USA, June 2011. 10, 11, 14, 16

[14] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollar, and C Lawrence Zitnick. Microsoft COCO captions: Data collection and evaluation server. *arXiv:1504.00325*, 2015. 17

[15] Xinlei Chen and C. Lawrence Zitnick. Learning a recurrent visual representation for image caption generation. *arXiv:1411.5654*, 2014. 2

[16] Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv:1409.1259*, 2014. 5, 9

[17] Timothee Cour, Ben Sapp, and Ben Taskar. Learning from partial labels. *The Journal of Machine Learning Research*, 12:1501–1536, 2011. 28

[18] Timothee Cour, Benjamin Sapp, Chris Jordan, and Ben Taskar. Learning from ambiguously labeled images. In *CVPR*, 2009. 28

[19] P. Das, R. K. Srihari, and J. J. Corso. Translating related words to videos and back through latent topics. In *WSDM*, 2013. 2, 4

[20] P. Das, C. Xu, R. F. Doell, and J. J. Corso. A thousand frames in just a few words: Lingual description of videos through latent topics and sparse object stitching. In *CVPR*, 2013. 2, 4

[21] M.C. De Marneffe, B. MacCartney, and C.D. Manning. Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC*, volume 6, pages 449–454, 2006. 5

[22] J. Deng, K. Li, M. Do, H. Su, and L. Fei-Fei. Construction and Analysis of a Large Scale Image Ontology. In *Vision Sciences Society*, 2009. 4

[23] Jia Deng, Jonathan Krause, Alexander C Berg, and Li Fei-Fei. Hedging your bets: Optimizing accuracy-specificity trade-offs in large scale visual recognition. In *CVPR*, pages 3450–3457, 2012. 5

[24] D. Ding, F. Metze, S. Rawat, P.F. Schulam, S. Burger, E. Younessian, L. Bao, M.G. Christel, and A. Hauptmann. Beyond audio and video retrieval: towards multimedia summarization. In *ICMR*. ACM, 2012. 2, 4

[25] Jeff Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*, 2015. 2, 4, 5, 6, 7, 8, 9, 10, 14, 15, 16, 24

[26] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. *arXiv preprint arXiv:1310.1531*, 2013. 9

[27] Desmond Elliott and Frank Keller. Comparing automatic evaluation measures for image description. In *ACL*, 2014. 12

[28] M. Everingham, J. Sivic, and A. Zisserman. "Hello! My name is... Buffy" – automatic naming of characters in TV video. In *BMVC*, 2006. 28

[29] 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. From captions to visual concepts and back. *CVPR*, 2015. 4, 24

[30] A. Farhadi, M. Hejrati, M. Sadeghi, P. Young, C. Rashtchian, J. Hockenmaier, and D. Forsyth. Every picture tells a story: Generating sentences from images. *ECCV*, 2010. 2, 4

[31] D. A. Forsyth and J. Ponce. *Computer Vision: A Modern Approach*. Prentice Hall, 2011. 4

[32] G. Gkioxari and J. Malik. Finding action tubes. 2014. 16

[33] Alan Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 6645–6649. IEEE, 2013. 26

[34] Alex Graves and Navdeep Jaitly. Towards end-to-end speech recognition with recurrent neural networks. In *ICML*, 2014. 5, 9

[35] S. Guadarrama, N. Krishnamoorthy, G. Malkarnenkar, S. Venugopalan, R.J. Mooney, T. Darrell, and K. Saenko. YouTube2Text: Recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition. In *ICCV*, 2013. 5, 9, 11

[36] Sergio Guadarrama, Lorenzo Riano, Dave Golland, Daniel Göhring, Yangqing Jia, Dan Klein, Pieter Abbeel, and Trevor Darrell. Grounding spatial relations for human-robot interaction. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS),*, 2013. 4

[37] C. Gulcehre, O. Firat, K. Xu, K. Cho, L. Barrault, H.C. Lin, F. Bougares, H. Schwenk, and Y. Bengio. On using monolingual corpora in neural machine translation. *arXiv preprint arXiv:1503.03535*, 2015. 22, 23

[38] Sepp Hochreiter, Yoshua Bengio, Paolo Frasconi, and Jürgen Schmidhuber. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies, 2001. 6

[39] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8), 1997. 3, 6, 8

[40] Peter Hodosh, Alice Young, Micah Lai, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. In *TACL*, 2014. 4, 9, 11, 12

[41] Haiqi Huang, Yueming Lu, Fangwei Zhang, and Songlin Sun. A multi-modal clustering method for web videos. In *Trustworthy Computing and Services*. Springer, 2013. 4

[42] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014. 10, 14

[43] D. Jurafsky and J. H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall, Upper Saddle River, NJ, 2009. 4

[44] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. *arXiv:1412.2306*, 2014. 2, 4

[45] Muhammad Usman Ghani Khan and Yoshihiko Gotoh. Describing video contents in natural language. *Proceedings of the Workshop on Innovative Hybrid Approaches to the Processing of Textual Data*, 2012. 2, 4

[46] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 18, 19

[47] Ryan Kiros, Ruslan Salakhuditnov, and Richard. S Zemel. Unifying visual-semantic embeddings with multimodal neural language models. *TACL*, 2015. 2, 4

[48] Philipp Koehn. *Statistical Machine Translation*. Cambridge University Press, 2010. 9

[49] A. Kojima, T. Tamura, and K. Fukunaga. Natural language description of human activities from video images based on concept hierarchy of actions. *IJCV*, 50(2), 2002. 2, 4

[50] Niveda Krishnamoorthy, Girish Malkarnenkar, Raymond J. Mooney, Kate Saenko, and Sergio Guadarrama. Generating natural-language video descriptions using text-mined knowledge. In *AAAI*, July 2013. 4, 5, 9, 11

[51] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet classification with deep convolutional neural networks. In *NIPS*, 2012. 9, 10

[52] Geert-Jan M. Kruijff, Hendrik Zender, Patric Jensfelt, and Henrik I. Christensen. Situated dialogue and spatial organization: What, where... and why? *International Journal of Advanced Robotic Systems*, 4(2), 2007. 4

[53] G. Kulkarni, V. Premraj, S. Dhar, S. Li, Y. Choi, A.C. Berg, and T.L. Berg. Baby talk: Understanding and generating simple image descriptions. In *CVPR*, 2011. 2, 4

[54] Polina Kuznetsova, Vicente Ordonez, Alexander C Berg, Tamara L Berg, and Yejin Choi. Collective generation of natural image descriptions. *ACL*, 2012. 2

[55] Polina Kuznetsova, Vicente Ordonez, Tamara L Berg, UNC Chapel Hill, and Yejin Choi. Treetalk: Composition and compression of trees for image descriptions. In *TACL*, 2014. 2

[56] M.W. Lee, A. Hakeem, N. Haering, and S.C. Zhu. Save: A framework for semantic annotation of visual events. In *CVPR*, 2008. 2, 4

[57] Li-Jia Li, Hao Su, Li Fei-Fei, and Eric P Xing. Object bank: A high-level image representation for scene classification & semantic feature sparsification. In *NIPS*, 2010. 5

[58] S. Li, G. Kulkarni, T.L. Berg, A.C. Berg, and Y. Choi. Composing simple image descriptions using web-scale N-grams. In *CoNLL*, 2011. 2, 4

[59] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 4, 9, 12

[60] Marcus Liwicki, Alex Graves, Horst Bunke, and Jürgen Schmidhuber. A novel approach to on-line handwriting recognition based on bidirectional long short-term memory networks. In *Proc. 9th Int. Conf. on Document Analysis and Recognition*, volume 1, pages 367–371, 2007. 26

[61] Zheng Lu and Kristen Grauman. Story-driven summarization for egocentric video. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 2714–2721. IEEE, 2013. 26

[62] Mateusz Malinowski and Mario Fritz. A multi-world approach to question answering about real-world scenes based on uncertain input. In *Advances in Neural Information Processing Systems*, pages 1682–1690, 2014. 4

[63] Jonathan Malmaud, Jonathan Huang, Vivek Rathod, Nick Johnston, Andrew Rabinovich, and Kevin Murphy. What's cookin'? interpreting cooking videos using text, speech and vision. *NAACL*, 2015. 28

[64] Junhua Mao, Wei Xu, Yi Yang, Jiang Wang, Zhiheng Huang, and Alan L. Yuille. Learning like a child: Fast novel visual concept learning from sentence descriptions of images. In *ICCV*, 2015. 4, 29

[65] Junhua Mao, Wei Xu, Yi Yang, Jiang Wang, and Alan L. Yuille. Deep captioning with multimodal recurrent neural networks (m-rnn). *arXiv:1412.6632*, 2014. 2

[66] Cynthia Matuszek, Dieter Fox, and Karl Koscher. Following directions using statistical machine translation. In *Proceedings of the Human Robot Interaction Conference (HRI)*, 2010. 4

[67] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013. 22

[68] Volodymyr Mnih, Nicolas Heess, Alex Graves, et al. Recurrent models of visual attention. In *Advances in Neural Information Processing Systems*, pages 2204–2212, 2014. 3, 21, 24

[69] Tanvi S. Motwani and Raymond J. Mooney. Improving video activity recognition using object recognition and text mining. In *Proceedings of the European Conference on Artificial Intelligence (ECAI)*, 2012. 11

[70] Joe Yue-Hei Ng, Matthew J. Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici. Beyond short snippets: Deep networks for video classification. *CoRR*, abs/1503.08909, 2015. 16

[71] Paul Over, George Awad, Martial Michel, Jonathan Fiscus, Greg Sanders, B Shaw, Alan F. Smeaton, and Georges Quéenot. TRECVID 2012 – an overview of the goals, tasks, data, evaluation mechanisms and metrics. In *Proceedings of TRECVID 2012*, 2012. 4

[72] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a method for automatic evaluation of machine translation. In *ACL*, 2002. 12

[73] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. *Proceedings of the Empiricial Methods in Natural Language Processing (EMNLP 2014)*, 12:1532–1543, 2014. 22

[74] Karl Pichotta and Raymond J. Mooney. Statistical script learning with multi-argument events. In *EACL*, 2014. 25

[75] Karl Pichotta and Raymond J. Mooney. Statistical script learning with multi-argument events. In *AAAI*, 2015. 25

[76] Zeeshan Rasheed and Mubarak Shah. Scene detection in hollywood movies and tv shows. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, pages II–343. IEEE, 2003. 27

[77] Mengye Ren, Ryan Kiros, and R Zemel. Exploring models and data for image question answering. In *NIPS*, volume 1, page 3, 2015. 4

[78] Anna Rohrbach, Marcus Rohrbach, and Bernt Schiele. The long-short story of movie description. *GCPR*, 2015. 18, 19

[79] Anna Rohrbach, Marcus Rohrbach, Niket Tandon, and Bernt Schiele. A dataset for movie description. In *CVPR*, 2015. 14, 16, 19, 28

[80] Marcus Rohrbach, Wei Qiu, Ivan Titov, Stefan Thater, Manfred Pinkal, and Bernt Schiele. Translating video content to natural language descriptions. In *ICCV*, 2013. 2

[81] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge, 2014. 9, 10, 15

[82] Roger C. Schank and Robert P. Abelson. *Scripts, Plans, Goals and Understanding: An Inquiry into Human Knowledge Structures*. Lawrence Erlbaum and Associates, Hillsdale, NJ, 1977. 25

[83] Roger C. Schank and Christopher K. Riesbeck. *Inside Computer Understanding: Five Programs plus Miniatures*. Lawrence Erlbaum, Hillsdale, NJ, 1981. 25

[84] Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neural networks. *Signal Processing, IEEE Transactions on*, 45(11):2673–2681, 1997. 26

[85] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, 2014. 14

[86] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. 15

[87] Hari Sundaram and Shih-Fu Chang. Video scene segmentation using video and audio features. In *Multimedia and Expo, 2000. ICME 2000. 2000 IEEE International Conference on*, volume 2, pages 1145–1148. IEEE, 2000. 27

[88] M. Sundermeyer, R. Schluter, and H. Ney. Lstm neural networks for language modeling. In *INTERSPEECH*, 2010. 23

[89] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In *NIPS*, 2014. 5, 6, 8, 9, 11

[90] Stefanie Tellex, Thomas Kollar, Steven Dickerson, Matthew R. Walter, Ashis Gopal Banerjee, Seth Teller, and Nicholas Roy. Understanding natural language commands for robotic navigation and mobile manipulation. In *Proceedings of the 25th AAAI Conference on Artificial Intelligence (AAAI)*, August 2011. 4

[91] J. Thomason, S. Venugopalan, S. Guadarrama, K. Saenko, and R.J. Mooney. Integrating language and vision to generate natural language descriptions of videos in the wild. In *COLING*, 2014. 2, 5, 9, 11, 12, 13, 17, 18

[92] Carlo Tomasi and Takeo Kanade. *Detection and tracking of point features*. School of Computer Science, Carnegie Mellon Univ. Pittsburgh, 1991. 28

[93] Atousa Torabi, Christopher Pal, Hugo Larochelle, and Aaron Courville. Using descriptive video services to create a large data source for video annotation research. *arXiv:1503.01070v1*, 2015. 14, 16

[94] Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. *CoRR*, abs/1411.5726, 2014. 17

[95] S. Venugopalan, H. Xu, J. Donahue, M. Rohrbach, R.J. Mooney, and K. Saenko. Translating videos to natural language using deep recurrent neural networks. In *NAACL-HLT*, 2015. 3, 12, 13

[96] Subhashini Venugopalan, Marcus Rohrbach, Jeff Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko. Sequence to sequence – video to text. In *ICCV*, 2015. 3, 13, 22, 27

[97] Subhashini Venugopalan, Huijuan Xu, Jeff Donahue, Marcus Rohrbach, Raymond Mooney, and Kate Saenko. Translating videos to natural language using deep recurrent neural networks. In *NAACL*, 2015. 15, 17, 18, 19

[98] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. *CVPR*, 2015. 2, 4, 5, 8, 9, 24

[99] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. *arXiv:1411.4555*, 2014. 15

[100] Heng Wang, Alexander Klaser, Cordelia Schmid, and Cheng-Lin Liu. Action recognition by dense trajectories. In *CVPR*, 2011. 5

[101] Shikui Wei, Yao Zhao, Zhenfeng Zhu, and Nan Liu. Multimodal fusion for video search reranking. *IEEE Transactions on Knowledge and Data Engineering,*, 22(8), 2010. 4

[102] Jianxiong Xiao, James Hays, Krista A. Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. *CVPR*, 2010. 5

[103] Kelvin Xu, Jimmy Ba, Ryan Kiros, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. *arXiv preprint arXiv:1502.03044*, 2015. 3, 21, 24

[104] R. Xu, C. Xiong, W. Chen, and J. J. Corso. Jointly modeling deep video and compositional text to bridge vision and language in a unified framework. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2015. 11

[105] Yezhou Yang, Ching Lik Teo, Hal Daumé, III, and Yiannis Aloimonos. Corpus-guided sentence generation of natural images. In *EMNLP*, 2011. 2

[106] 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. 2

[107] Li Yao, Atousa Torabi, Kyunghyun Cho, Nicolas Ballas, Christopher Pal, Hugo Larochelle, and Aaron Courville. Describing videos by exploiting temporal structure. *arXiv:1502.08029v4*, 2015. 15, 17, 18, 19, 20, 27

[108] Haonan Yu and Jeffrey Mark Siskind. Grounded language learning from videos described with sentences. In *ACL*, 2013. 2, 4

[109] Licheng Yu, Eunbyung Park, Alexander C Berg, and Tamara L Berg. Visual madlibs: Fill in the blank image generation and question answering. *arXiv preprint arXiv:1506.00278*, 2015. 4

[110] Wojciech Zaremba and Ilya Sutskever. Learning to execute. *arXiv:1410.4615*, 2014. 6

[111] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *ECCV*. Springer, 2014. 9

[112] Yun Zhai and Mubarak Shah. Video scene segmentation using markov chain monte carlo. *Multimedia, IEEE Transactions on*, 8(4):686–697, 2006. 27

[113] Yukun Zhu, Ryan Kiros, Richard S. Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. *ICCV*, 2015. 22, 28